Data Mining
W4240 Section 001

Giovanni Motta

Columbia University, Department of Statistics

October 21, 2015

Outline

Outline

Classification: Why and When

Naive Bayes Classification

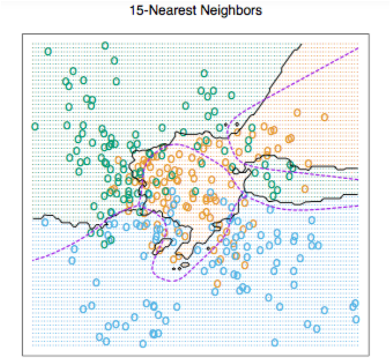
The ‘Naive’ assumption: what and why

The Dangers of Naiveté

Naive Bayes in Practice

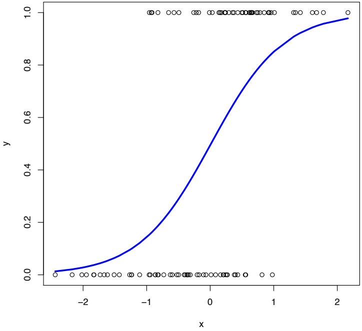
Building a Classification Toolbox

- k-NN



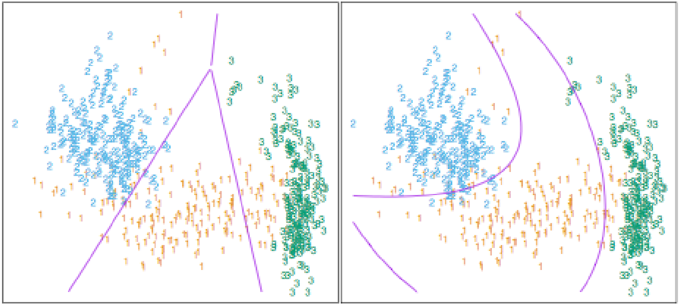
Building a Classification Toolbox

- k-NN
- Logistic Regression



Building a Classification Toolbox

- k-NN
- Logistic Regression
- Discriminant Analysis



- When is one the right choice, and when not?
- What other data scenarios exist?

Outline

A Different Classification Problem

- Suppose that I have two coins, C_1 and C_2
- Now suppose I pull a coin out of my pocket, flip it a bunch of times, record the coin and outcomes, and repeat many times:

```
C1: 0 1 1 1 1
C1: 0 1 0
C2: 1 0 0 0 0 0 0 1
C1: 0 1
C1: 1 1 0 1 1 1
C2: 0 0 1 1 0 1
C2: 1 0 0 0
```

- Now suppose I am given a new sequence, `0 0 1 0 0 1`; which coin is it from?

A Classification Problem

This problem has particular challenges:

- different numbers of covariates for each observation
- number of covariates can be large

However, there is some structure:

- Easy to estimate $P(C_1)$, $P(C_2)$
- Also easy to get $P(X_i = 1 \,|\, C_1)$ and $P(X_i = 1 \,|\, C_2)$
- By conditional independence,

$$P(X = 0\,1\,0 \mid C_1) = P(X_1 = 0 \,|\, C_1)P(X_2 = 1 \,|\, C_1)P(X_3 = 0 \,|\, C_1)$$

- Bayes rule yields $P(C_1 \,|\, X = \; 0\,0\,1\,0\,0\,1\,)$

Reminder

Suppose we want to classify an observation into one of K classes, where $K \geq 2$. Let C_k denote the k-th class, $k = 1, \ldots, K$.

Define

$$\begin{aligned} \pi_k =& \mathbb{P}\left(Y = k\right) && \textit{prior} \text{ probability that an observation } Y \text{ belongs to } C_k \\ p_k(x) =& \mathbb{P}\left(Y = k \,|\, X = x\right) && \textit{posterior} \text{ probability that an observation } X = x \text{ belongs to } C_k \\ f_k(x) =& \mathbb{P}\left(X = x \,|\, Y = k\right) && \textit{density} \text{ function of } X \text{ for an observation that belongs to } C_k \end{aligned}$$

- $f_k(x)$ is the density for class k
- π_k is the probability of class k, with $\displaystyle\sum_{k=1}^{K} \pi_k = 1$

$$\mathbb{P}\left(Y = k \,|\, X = x\right) = \frac{\mathbb{P}(X = x \,|\, Y = k)\mathbb{P}(Y = k)}{\sum_{\ell=1}^{K} \mathbb{P}(X = x \,|\, Y = \ell)\mathbb{P}(Y = \ell)} = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

A Classification Problem

$$\frac{\mathbb{P}(C_1 \,|\, X = 0\,0\,1\,0\,0\,1)}{\mathbb{P}(C_2 \,|\, X = 0\,0\,1\,0\,0\,1)} = \frac{\mathbb{P}(C_1)\,\mathbb{P}(X = 0\,0\,1\,0\,0\,1 \,|\, C_1)}{\mathbb{P}(C_2)\,\mathbb{P}(X = 0\,0\,1\,0\,0\,1 \,|\, C_2)}$$

- How to estimate $\mathbb{P}(C_1)$ and $P(C_2)$?
- How to estimate $\mathbb{P}(X = 0\,0\,1\,0\,0\,1 \,|\, C_1)$ and $\mathbb{P}(X = 0\,0\,1\,0\,0\,1 \,|\, C_2)$?

A Classification Problem

Training data:

```
C1: 0 1 1 1 1
C1: 0 1 0
C2: 1 0 0 0 0 0 0 1
C1: 0 1
C1: 1 1 0 1 1 1
C2: 0 0 1 1 0 1
C2: 1 0 0 0
```

Testing data: `0 0 1 0 0 1`

Calculate empirical estimates of $P(C_1|X = 001001)$, $P(C_2|X = 001001)$
Naive Bayes:

$$\hat{C} = \begin{cases} C_1 & P(C_1|X = 001001) > P(C_2|X = 001001) \\ C_2 & otherwise \end{cases}$$

Midterm-esque example: process control

Naive Bayes:

$$\hat{C} = \begin{cases} C_1 & P(C_1|X) > P(C_2|X) \\ C_2 & otherwise \end{cases}$$

Two factories produce my product, and QA is independently performed on each day's batch. Training data:

```
F1: 0 0 0 1 1
F2: 0 1 1
F2: 1 1 1 1 0 0 0 1
F2: 1 1 1 1 1 1
F1: 0 1 1 1 0 1
```

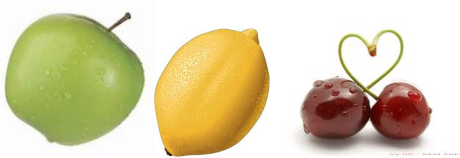
Question: I get unmarked QA results `1 1 1 0 0 1`. Do I believe this came from `F1` or `F2`?

Outline

Naive Bayes Classifier

Note that the coin flips are **conditionally independent** given the coin parameter. What about this case:

- want to identify the type of fruit given a set of features: color, shape and size
- color: red, green, yellow or orange (categorical)
- shape: round or oval (categorical)
- size: diameter in inches (continuous)



Naive Bayes Classifier

Conditioned on type of fruit, these features are not necessarily independent:

Given category "apple," the color "green" has a different probability given "size < 2":

$$P(green \,|\, size < 2,\, apple) \neq P(green \,|\, apple)$$

Naive Bayes Classifier

$$
\begin{aligned}
P(apple \,|\, green, round, size = 2) & \\
&= \frac{P(green, round, size = 2 \,|\, apple) P(apple)}{\sum_{fruits} P(green, round, size = 2 \,|\, fruit\, j) P(fruit\, j)} \\
&\propto P(green \,|\, round, size = 2, apple) P(round \,|\, size = 2, apple) \\
&\qquad \times P(size = 2 \,|\, apple) P(apple)
\end{aligned}
$$

We used Bayes and chain rule

$$P(green, round, size = 2 \,|\, apple) P(apple) = P(green, round, size = 2, apple)$$

$$
\begin{aligned}
P(green, round, size = 2, apple) = & P(green \,|\, round, size = 2, apple) \\
& \times P(round \,|\, size = 2, apple) \\
& \times P(size = 2 \,|\, apple) P(apple)
\end{aligned}
$$

Computing conditional probabilities has challenges: there are many combinations of $(color, shape, size)$ for each fruit.

Naive Bayes Classifier

'Naive' idea: assume conditional independence for all features given class,

$$P(green \,|\, round, size = 2, apple) = P(green \,|\, apple)$$
$$P(round \,|\, green, size = 2, apple) = P(round \,|\, apple)$$
$$P(size = 2 \,|\, green, round, apple) = P(size = 2 \,|\, apple)$$

More generally for features (covariates) $X_1, \ldots, X_m$ and class Y,

$$P(X_i \,|\, X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_m, Y) = P(X_i \,|\, Y),$$
$$P(X_1, \ldots, X_m \,|\, Y) = \prod_{i=1}^{m} P(X_i \,|\, Y)$$

Assuming conditional independence is an approximation

Naive Bayes Classifier

What is naive?

Wikipedia definition:

1. lacking experience, wisdom, or judgement
2. (*of art*) produced in a simple, childlike style, deliberately rejecting sophisticated techniques

Merriam-Webster definition:

1. marked by unaffected simplicity
2. deficient in worldly wisdom or informed judgement
3. self-taught, primitive

Naive assumption: conditional independence

Naive Bayes Classifier

Why conditional independence?

- estimating multivariate functions (like $P(X_1, \ldots, X_m \,|\, Y)$) is mathematically more difficult than estimating univariate functions (like $P(X_i \,|\, Y)$)
- need less data to fit univariate functions well
- univariate estimators differ much less than multivariate estimator (low variance)
- ... but they may end up finding the wrong values (more bias)
- (Remember the bias-variance decomposition of error)

Naive Bayes Classifier

Naive Bayes model:

$$\begin{aligned} P(Y = y \quad | \quad & X_1 = x_1, \ldots, X_m = x_m) \\ \propto \quad & P(Y = y) P(X_1 = x_1, \ldots, X_m = x_m \,|\, Y = y) \\ \approx \quad & P(Y = y) \prod_{i=1}^{m} P(X_i = x_i \,|\, Y = y) \end{aligned}$$

Naive Bayes classifier:

$$\begin{aligned} \hat{y}^{NB} \quad &= \quad \arg\max_{\tilde{y}} \frac{P(Y = \tilde{y}) P(X = x_{test} \,|\, Y = \tilde{y})}{P(X = x_{test})} \\ &= \quad \arg\max_{\tilde{y}} P(Y = \tilde{y}) \prod_{i=1}^{m} P(X_i = x_{test,i} \,|\, Y = \tilde{y}) \end{aligned}$$

Naive Bayes Classifier

The conditional independence assumption makes Naive Bayes good for high dimensional data:

- often not enough data for high dimensional problems without strong assumptions
- may not estimate class probabilities correctly, but often makes decisions correctly
  - Ex: may not estimate $P(apple \,|\, yellow, round, size = 1.8)$ correctly, but will say that

$$P(apple \,|\, yellow, round, s = 1.8) > P(lemon \,|\, yellow, round, s = 1.8)$$

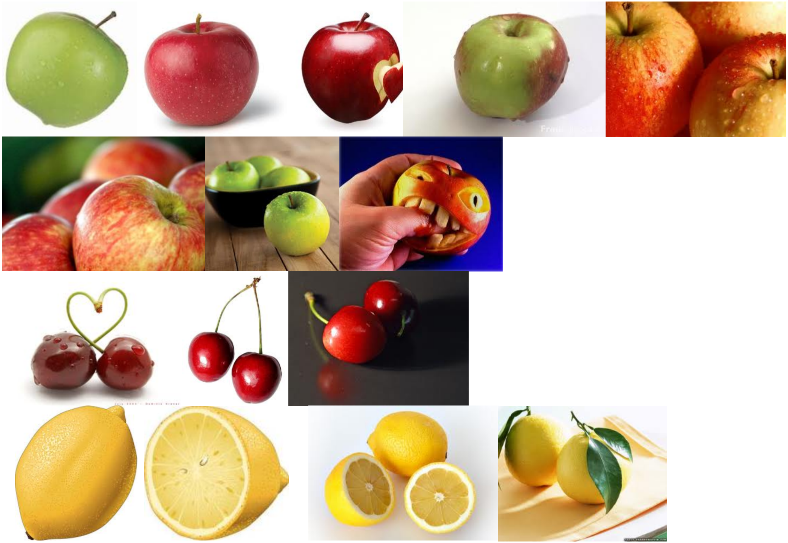
Outline

Naive Bayes Classifier

Naive Bayes does not do well or as well as competitors when:

- there are repeated covariates
- there is a lot of data and few covariates (can be beaten by other methods)
- the covariates are not all equally important
- the testing data is not from the same distribution as the training data

Example



Example

Need to compute:

- Class probabilities: $P(apple)$, $P(cherry)$, $P(lemon)$
- Feature probabilities given class: $P(green \,|\, apple)$, $P(red \,|\, apple)$, ...

Test on:

For example, what quantities do we need to classify fruit type from color?

Example

Color	Shape	Size	Fruit
Green	Round	2.1	Apple
Red	Round	1.9	Apple
Red	Round	2	Apple
Green	Round	1.8	Apple
Red	Round	1.9	Apple
Red	Round	2.1	Apple
Green	Round	1.6	Apple
Red	Round	1.7	Apple
Red	Round	1.1	Cherry
Red	Round	1	Cherry
Red	Round	1.2	Cherry
Yellow	Oval	2.8	Lemon
Yellow	Oval	2.6	Lemon
Yellow	Oval	2.5	Lemon
Yellow	Round	2.7	Lemon

Example

Class probabilities:

- $P(apple) =$
- $P(cherry) =$
- $P(lemon) =$

Example

Conditional color probabilities:

- $P(red \,|\, apple) =$
- $P(green \,|\, apple) =$
- $P(yellow \,|\, apple) =$

If we use proportions of data seen, $P(yellow \,|\, apple) = 0$

Can we work around this issue?

Laplace Smoothing

- With the Nave Bayes Assumption, we can still end up with zero probabilities
- E.g., if we receive an email that contains a word that has never appeared in the training emails
  - $\mathbb{P}(X|Y)$ will be 0 for all Y values
  - We can only make prediction based on $\mathbb{P}(Y)$
- This is bad because we ignored all the other words in the email because of this single rare word
- Laplace smoothing can help

$$\mathbb{P}(X_1 = 1|Y = 0) = \frac{1 + \text{ \# of examples with } Y = 0, X_1 = 1}{m + \text{ \# of examples with } Y = 0}$$

  where $m =$ the total number of possible values of x
- For a binary feature like above, $p(X|Y)$ will not be 0.

Laplace (or Lidstone) Smoothing Multinomial Data

Idea: add μ 'phantom' observations to each category

$$P(yellow \,|\, apple) = \frac{\#\ yellow\ apples\ seen + \mu}{\#\ apples\ seen\ +\ (\#\ colors)\mu}$$

Set (for example) $\mu = \frac{1}{\#\ colors}$. Now compute:

- $P(red \,|\, apple)$, $P(green \,|\, apple)$, $P(yellow \,|\, apple)$
- $P(red \,|\, cherry)$, $P(green \,|\, cherry)$, $P(yellow \,|\, cherry)$
- $P(red \,|\, lemon)$, $P(green \,|\, lemon)$, $P(yellow \,|\, lemon)$

A very good idea in practice, for exactly this reason.

Example

Conditional color probabilities: set $\mu = \frac{1}{\#\ shapes}$

- $P(round \,|\, apple)$, $P(oval \,|\, apple)$
- $P(round \,|\, cherry)$, $P(oval \,|\, cherry)$
- $P(round \,|\, lemon)$, $P(oval \,|\, lemon)$

Example

Conditional size probabilities:

- bin sizes to make discrete data:
  $\{size < 2\}$, $\{2 \leq size < 2.5\}$, $\{size \geq 2.5\}$
- other option: places positive density on *all* sizes, so no need to add unseen examples

Calculate:

- apple: $P(size < 2 \,|\, apple)$, $P(2 \leq size < 2.5 \,|\, apple)$, $P(size \geq 2.5 \,|\, apple)$
- cherry: $P(size < 2 \,|\, cherry)$, $P(2 \leq size < 2.5 \,|\, cherry)$, $P(size \geq 2.5 \,|\, cherry)$
- lemon: $P(size < 2 \,|\, lemon)$, $P(2 \leq size < 2.5 \,|\, lemon)$, $P(size \geq 2.5 \,|\, lemon)$

Example

So which class is this?

Color $=$ yellow, shape $=$ round, size $=$ 1.8

Example

Calculate probabilities:

$$
\begin{aligned}
&P(apple \,|\, yellow, round, size < 2) \\
&\quad = \frac{P(yellow \,|\, apple)P(round \,|\, apple)P(size < 2 \,|\, apple)P(apple)}{\sum_c P(yellow \,|\, c)P(round \,|\, c)P(size < 2 \,|\, c)P(c)} \\
&\quad \propto P(apple)P(yellow \,|\, apple)P(round \,|\, apple)P(size < 2 \,|\, apple)
\end{aligned}
$$

Remove constants that belong to all classes and compare (proportional) probabilities

Example

Compute:

- $P(apple \,|\, yellow, round, size < 2) \propto$
- $P(cherry \,|\, yellow, round, size < 2) \propto$
- $P(lemon \,|\, yellow, round, size < 2) \propto$

Maximum value is the selected class

Outline

Real World Examples

Naive Bayes performs surprisingly well on many real world applications:

- Spam filtering (document classification)
- Medical diagnoses
- Sentiment analysis (attitude of writer positive or negative?)

Spam Filtering

Under conditional independence, a document can be reduced to a *bag of words*

Original Wikipedia article:

```
Thomas Bayes was an English mathematician and Presbyterian minister, known
for having formulated a specific case of the theorem that bears his name:
Bayes' theorem. Bayes never published what would eventually become his most
famous accomplishment; his notes were edited and published after his death
by Richard Price.
```

Bag of words representation:

$$\begin{array}{cc} 2 & \texttt{a} \\ 0 & \texttt{aardvark} \\ 0 & \texttt{aardwolt} \\ \vdots & \vdots \\ 3 & \texttt{Bayes} \\ \vdots & \vdots \\ 0 & \texttt{zygmurgy} \end{array}$$

Spam Filtering

In a bag of words model, a document d has a distribution over the frequency of these words,

$$\begin{array}{cc}
p_a & \texttt{a} \\
p_{aardvark} & \texttt{aardvark} \\
p_{aardwolt} & \texttt{aardwolt} \\
\vdots & \vdots \\
p_{Bayes} & \texttt{Bayes} \\
\vdots & \vdots \\
p_{zygmurgy} & \texttt{zygmurgy}
\end{array}$$

Each word, x_j, in the document is drawn from a multinomial distribution defined by this distribution,

$$x_j \sim Multi(p)$$

Spam Filtering

- In spam filtering, we would like to separate junk email (spam), from legitimate email (ham)
- We assume that there are two different word frequencies, p_{spam} and p_{ham}, for spam and ham
- Naive Bayes classification uses documents to estimate $P(spam)$, $P(ham)$, $P(word\ i \,|\, ham)$ and $P(word\ i \,|\, spam)$

Spam Filtering

There is a lot of junk in documents:

- punctuation (!.,":;}¡)
- stopwords (a, the, and, to, from, an,...)
- verb conjugations (type vs. typed vs. types)
- noun declensions (horse vs. horses, goose vs. geese)

So, let's get rid of it! This step is done through stopword removal and stemming.

Spam Filtering

Subject: negative concord

i am interested in the grammar of negative concord in various dialects of american and british english . if anyone out there speaks natively a dialect that uses negative concord and is willing to answer grammaticality questions about their dialect , would they please send me an email note to that effect and i ' ll get back to them with my queries . my address is : kroch @ change . ling . upenn . edu thanks .

The cleaned version is

negative concord interest grammar negative concord various dialect american british english anyone speak natively dialect negative concord answer grammaticality question dialect please send email note effect ll back query address kroch change ling upenn edu thank

Spam Filtering

Subject: the best , just got better

the 2 newest and hottest interactive adult web sites will soon be the largest ! ! ! check out both of the following adult web sites free samples and trial memberships ! ! ! ! live triple x adult entertainment and pictures . new content added weekly ! ! ! check them both out : http : / / www2 . dirtyonline . com http : / / www . chixchat . com

The cleaned version is

best better newest hottest interactive adult web site soon largest check both follow adult web site free sample trial membership live triple x adult entertainment picture content add weekly check both http www dirtyonline com http www chixchat com

Spam Filtering

Label all spam $y_i = 1$ and all ham $y_i = 0$. We have a new document $\mathbf{x}^{test}$ with n^{test} words:

$$\begin{aligned}
p(Y = 1 \mid \mathbf{X} = \mathbf{x}^{test}) &= \frac{p(\mathbf{x}_{test} \mid Y = 1)p(Y = 1)}{p(\mathbf{x}^{test})}, \\
&= \frac{1}{p(\mathbf{x}^{test})} p(Y = 1) \prod_{j=1}^{n^{test}} p(X_j = x_j^{test} \mid Y = 1), \\
p(Y = 0 \mid \mathbf{X} = \mathbf{x}^{test}) &= \frac{p(\mathbf{x}^{test} \mid Y = 0)p(Y = 0)}{p(\mathbf{x}^{test})}, \\
&= \frac{1}{p(\mathbf{x}^{test})} p(Y = 0) \prod_{j=1}^{n^{test}} p(X_j = x_j^{test} \mid Y = 0).
\end{aligned}$$

Spam Filtering

Just like the previous example, we need to approximate:

- class probabilities, $p(Y = y)$
- conditional word probabilities, $p(X_j = x_j \mid Y = \tilde{y})$

Suppose we have m documents. To approximate $p(Y = \tilde{y})$, set

$$\hat{p}(Y = \tilde{y}) = \sum_{i=1}^{m} \frac{1}{m} \mathbf{1}_{\{Y_i = \tilde{y}\}}.$$

Spam Filtering

To compute the conditional word probabilities, smooth with μ phantom instances:

$$\hat{p}(X_{ij} = k \,|\, Y_i = \tilde{y}) = \frac{\mu + \sum_{i=1}^{m} \sum_{j=1}^{n_i} \mathbf{1}_{\{X_{ij}=k, Y_j=\tilde{y}\}}}{\mu|D| + \sum_{i=1}^{m} n_i \mathbf{1}_{\{Y_i=\tilde{y}\}}}.$$

Here $|D|$ is the size of your dictionary.

(note Laplace smoothing here)

A Procedure for Spam Filtering

To calculate these probabilities:

1. read in the training data
2. make a dictionary
3. store word counts for each document in a *document term matrix*
4. use document term matrix to compute probabilities
5. input a new test document and use probabilities to classify

Spam Filtering

To read in documents:

```
#==================================================================

# To read in data from the directories:
# Partially based on code from C. Shalizi
read.directory <- function(dirname) {
    # Store the emails in a list
    emails = list();
    # Get a list of filenames in the directory
    filenames = dir(dirname,full.names=TRUE);
    for (i in 1:length(filenames)){
        emails[[i]] = scan(filenames[i],what="",quiet=TRUE);
         }
    return(emails)
}
# Example: ham.test <- read.directory("Homework3Data/nonspam-test/")

#==================================================================
```

Spam Filtering

To make a dictionary:

```
#=================================================================

# Make dictionary sorted by number of times a word appears in corpus
# (useful for using commonly appearing words as factors)
# NOTE: Use the *entire* corpus: training, testing, spam and ham
make.sorted.dictionary.df <- function(emails){
    # This returns a dataframe that is sorted by the number of times
    # a word appears

    # List of vectors to one big vector
    dictionary.full <- unlist(emails)
    # Tabulates the full dictionary
    tabulate.dic <- tabulate(factor(dictionary.full))
    # Find unique values
    dictionary <- unique(dictionary.full)
    # Sort them alphabetically
    dictionary <- sort(dictionary)
    dictionary.df <- data.frame(word = dictionary, count = tabulate.dic)
    sort.dictionary.df <- dictionary.df[order(dictionary.df$count,decreasing=TRUE),];
    return(sort.dictionary.df)
}

#=================================================================
```

Spam Filtering

To make a document term matrix:

```
#==============================================================

# Make a document-term matrix, which counts the number of times each
# dictionary element is used in a document
make.document.term.matrix <- function(emails,dictionary){
    # This takes the email and dictionary objects from above and outputs a
    # document term matrix
    num.emails <- length(emails);
    num.words <- length(dictionary);
    # Instantiate a matrix where rows are documents and columns are words
    dtm <- mat.or.vec(num.emails,num.words); # A matrix filled with zeros
    for (i in 1:num.emails){
        num.words.email <- length(emails[[i]]);
        email.temp <- emails[[i]];
        for (j in 1:num.words.email){
            ind <- which(dictionary == email.temp[j]);
            dtm[i,ind] <- dtm[i,ind] + 1;
        }
    }
return(dtm);
}
# Example: dtm <- make.document.term.matrix(emails.train,dictionary)

#==============================================================
```

Spam Filtering

To make log probabilities:

```
#==================================================================

make.log.pvec <- function(dtm,mu){
    # Sum up the number of instances per word
    pvec.no.mu <- colSums(dtm)
    # Sum up number of words
    n.words <- sum(pvec.no.mu)
    # Get dictionary size
    dic.len <- length(pvec.no.mu)
    # Incorporate mu and normalize
    log.pvec <- log(pvec.no.mu + mu) - log(mu*dic.len + n.words)
    return(log.pvec)
}

#==================================================================
```

Spam Filtering

To make classifier:

on next homework

Spam Filtering

- Easy to train Bayesian filter for individual users
- More sensitive and easily tuned than rules-based classification
- Very good at avoiding false positives (rules might say "Nigeria" = spam, but NB might discount "Nigeria" if there is a lot of other legitimate text)

Bayesian spam filtering works really well. It is implemented by many modern mail clients and server-side filters, like DSPAM, SpamAssassin, SpamBayes, Bogofilter and ASSP.