Bayes' Rule for classification

Recall Bayes' rule:

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

You might remember that each of these terms has a name:

- $P(A)$: the prior probability
- $P(A \mid B)$: the posterior probability
- $P(B \mid A)$: the likelihood
- $P(B)$: the marginal (total/overall) probability

In classification, "A" is a class label and "B" is a set of features.

Bayes' Rule for classification

Bayes's rule:

$$P(y = k \mid x) = \frac{P(y = k) \cdot P(x \mid y = k)}{P(x)}$$

$P(y = k)$ is the prior probability for class k. We usually get this from the raw class frequencies in the training data. For example:

```
table(fgl_train$type) %>% prop.table %>% round(3)
```

```
##
##  WinF WinNF   Veh   Con  Tabl  Head
## 0.297 0.366 0.087 0.058 0.047 0.145
```

Bayes' Rule for classification

Bayes's rule:

$$P(y = k \mid x) = \frac{P(y = k) \cdot P(x \mid y = k)}{P(x)}$$

$P(x)$ is the marginal probability of observing feature vector x. Notice it doesn't depend on k! It's the same number for all classes.

Thus we usually write the posterior probabilities up to this constant of proportionality, without bothering to compute it:

$$P(y = k \mid x) \propto P(y = k) \cdot P(x \mid y = k)$$

(Note: often we do the actual computations on a log scale instead.)

Bayes' Rule for classification

Bayes's rule:

$$P(y = k \mid x) = \frac{P(y = k) \cdot P(x \mid y = k)}{P(x)}$$

The hard part is estimating the likelihood $P(x \mid y = k)$. In words: how likely is it that we would have observed feature vector x if the true class label were k?

This is like regression in reverse!

Naive Bayes

Recall that $x = (x_1, x_2, \ldots, x_p)$ is a vector of p features. The simplest strategy for estimating $P(x \mid y = k)$ is called "Naive Bayes."

It's "naive" because we make the simplifying assumption that *every feature x_j is independent* of all other features, conditional on the class labels:

$$\begin{aligned} P(x \mid y = k) &= P(x_1, x_2, \ldots, x_p \mid y = k) \\ &= \prod_{j=1}^{p} P(x_j \mid y = k) \quad \text{(independence)} \end{aligned}$$

This simplifies the requirements of the problem: *just calculate the marginal distribution of the features,* i.e. $P(x_j \mid y = k)$ for all features j and classes k.

Naive Bayes: a small example

In `congress109.csv` we have data on all speeches given on the floor of the U.S. Congress during the 109th Congressional Session (January 3, 2005 to January 3, 2007).

Every row is a set of *phrase counts* associated with a single representative's speeches across the whole session. X_{ij} = number of times that rep i utter phrase j during a speech.

The target variable $y \in$ R, D is the party affiliation of the representative.

Naive Bayes: a small example

We'll focus on just a few phrases and famous politicians:

```
# read in data
congress109 = read.csv("../data/congress109.csv", header=TRUE, row.names=1)
congress109members = read.csv("../data/congress109members.csv", header=TRUE, row.names=1)
```

Focus on a few key phrases and a few famous pols:

```
X_small = dplyr::select(congress109, minimum.wage, war.terror, tax.relief, hurricane.katrina)
X_small[c('John McCain', 'Mike Pence', 'John Kerry', 'Edward Kennedy'),]
```

```
##                minimum.wage war.terror tax.relief hurricane.katrina
## John McCain               0         27          0                14
## Mike Pence                0         12          1                11
## John Kerry               12         16         13                23
## Edward Kennedy          260          8          1                53
```

Naive Bayes: a small example

Let's look at these counts summed across all members in each party:

```
y = congress109members$party

# Sum phrase counts by party
R_rows = which(y == 'R')
D_rows = which(y == 'D')
colSums(X_small[R_rows,])
```

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##               294               604               497               717
```

```
colSums(X_small[D_rows,])
```

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##               767               237               176              1295
```

So we get the sense that some phrases are "more Republican" and some "more Democrat."

Naive Bayes: a small example

To make this precise, let's build our Naive Bayes model for a Congressional speech:

- Imagine that every phrase uttered in a speech is a random sample from a "bag of phrases," where each phrase has its own probability. (*This is the Naive Bayes assumption of independence.*)
- Here the bag consists of just four phrases: "minimum wage", "war on terror", "tax relief," and "hurricane katrina".
- Each class (R or D) has its own probability vector associated with the phrases in the bag.

Naive Bayes: a small example

We can estimate these probability vectors for each class from the phrase counts in the training data.

For Republicans:

```
probhat_R = colSums(X_small[R_rows,])
probhat_R = probhat_R/sum(probhat_R)
probhat_R %>% round(3)
```

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##             0.139             0.286             0.235             0.339
```

And for Democrats:

```
probhat_D = colSums(X_small[D_rows,])
probhat_D = probhat_D/sum(probhat_D)
probhat_D %>% round(3)
```

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##             0.310             0.096             0.071             0.523
```

Naive Bayes: a small example

Let's now look at some particular member of Congress and try to build the likelihood, $P(x \mid y)$, for his or her phrase counts

```
X_small['Sheila Jackson-Lee',]
```

```
##                     minimum.wage war.terror tax.relief hurricane.katrina
## Sheila Jackson-Lee           11         15          3                66
```

Are Sheila Jackon-Lee's phrase counts $x = (11, 15, 3, 66)$ more likely under the Republican or Democrat probability vector?

Naive Bayes: a small example

Recall $P(x \mid y = R)$:

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##            0.1392            0.2860            0.2353            0.3395
```

Under this probability vector:

$$
\begin{aligned}
P(x \mid y = \mathrm{R}) &= P(x_1 = 11 \mid y = \mathrm{R}) \\
&\times P(x_2 = 15 \mid y = \mathrm{R}) \\
&\times P(x_3 = 3 \mid y = \mathrm{R}) \\
&\times P(x_4 = 66 \mid y = \mathrm{R}) \\
&= (0.1392)^{11} \cdot (0.2860)^{15} \cdot (0.2353)^{3} \cdot (0.3395)^{66} \\
&= 3.765 \times 10^{-51}
\end{aligned}
$$

Naive Bayes: a small example

Now recall $P(x \mid y = D)$:

```
##      minimum.wage        war.terror        tax.relief hurricane.katrina
##            0.1392            0.2860            0.2353            0.3395
```

Under this probability vector:

$$
\begin{aligned}
P(x \mid y = \mathrm{D}) &= P(x_1 = 11 \mid y = \mathrm{D}) \\
&\times P(x_2 = 15 \mid y = \mathrm{D}) \\
&\times P(x_3 = 3 \mid y = \mathrm{D}) \\
&\times P(x_4 = 66 \mid y = \mathrm{D}) \\
&= (0.3099)^{11} \cdot (0.0958)^{15} \cdot (0.0711)^{3} \cdot (0.5232)^{66} \\
&= 1.293 \times 10^{-43}
\end{aligned}
$$

Naive Bayes: a small example

These numbers are tiny, so it's much safer to work on a log scale:

$$\log P(x \mid y = k) = \sum_{j=1}^{p} x_j \log p_j^{(k)}$$

where $p_j^{(k)}$ is the jth entry in the probability vector for class k.

```
x_try = X_small['Sheila Jackson-Lee',]
sum(x_try * log(probhat_R))
```

```
## [1] -116.1083
```

```
sum(x_try * log(probhat_D))
```

```
## [1] -98.75633
```

Naive Bayes: a small example

Let's use Bayes' rule (posterior $\propto$ prior times likelihood) to put this together with our prior, estimated using the empirical class frequencies:

```
table(y) %>% prop.table %>% round(3)
```

```
## y
##     D     I     R
## 0.457 0.004 0.539
```

So:

$$P(R \mid x) \propto 0.539 \cdot (3.765 \times 10^{-51})$$

and

$$P(D \mid x) \propto 0.457 \cdot (1.293 \times 10^{-43})$$

Naive Bayes: a small example

To actually calculate a posterior, we must turn this into a set of probabilities by normalizing, i.e. dividing by the sum across all classes:

$$
\begin{aligned}
P(D \mid x) &= \frac{0.457 \cdot (1.293 \times 10^{-43})}{0.457 \cdot (1.293 \times 10^{-43} + 0.539 \cdot (3.765 \times 10^{-51})} \\
&\approx 1
\end{aligned}
$$

So:

1. Our model thinks Sheila Jackson-Lee is a Democrat.
2. The data completely overwhelm the prior! This is often the case in Naive Bayes models.

Naive Bayes: a bigger example

Let's turn to `congress109_bayes.R` to see a larger example of Naive Bayes classification, where we fit our model with all 1000 phrase counts.

Naive Bayes: summary

- Works by directly modeling $P(x \mid y)$, versus $P(y \mid x)$.
- This **regression in reverse** only works because we assume that each feature in x is independent, given the class labels.
- Simple and easy to compute, and therefore scalable to very large data sets and classification problems.
- Unlike a logit model, it works even more with features P than examples N.
- Often too simple: the "naive" assumption of independence really is a drastic simplification.
- The resulting probabilities are useful for classification purposes, but often not believeable as probabilities.
- Most useful when the features x are categorical variables (like phrase counts!) Very common in text analysis.