4.4.4 Naive Bayes

Dr. Lauren Perry

Bayes Classifier

Recall: Bayes' theorem gives us the expression

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

and we need to estimate $\pi_1, \ldots, \pi_K$ and $f_1(x), \ldots f_K(x)$.

Bayes Classifier

- In practice, estimating $\pi_1, \ldots, \pi_K$ is fairly simple.
- Estimating $f_1(x), \ldots f_K(x)$ is relatively more involved.
  - We can simplify this with strong assumptions about the distribution, e.g., multivariate normal.
    - In this case, we need only to estimate the distributions *parameters*.
  - Naive Bayes' takes a different approach.

Naive Bayes

Assumption: within the kth class, the p predictors are independent.

Mathematically, this means we can write

$$f_k(x) = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)$$

where f_{kj} is the density function of the jth predictor among observations in the kth class.

- Assuming independence means we only have to estimate the K *marginal distributions*.
- Without independence, we also need to consider the *joint* distributions, or associations between the different predictors.
    - This is OK for multivariate normal (Σ_k), but can be very complex.

Naive Bayes

How strong is this assumption?

- Not very strong compared to assuming all predictors are normal.
- Even when it's violated (and it usually is), it tends to give pretty good results.
  - Especially true when n is small relative to p.
    - We need a lot of data to estimate those joint distributions!
- Naive Bayes introduces a little bias, but reduces variance.

In practice, Naive Bayes works quite well.

Naive Bayes

Under this assumption,

$$P(Y = k|X = x) = \frac{\pi_k \times f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)}{\sum_{l=1}^{K} \pi_l \times f_{l1}(x_1) \times f_{l2}(x_2) \times \cdots \times f_{lp}(x_p)}$$

for $k = 1, \ldots K$.

Estimating f_{kj}: Parametric Approach

- Assume $X_j|Y = k \sim N(\mu_{jk}, \sigma^2_{jk})$.
  - Like QDA, but the class-specific covariance matrices are diagonal.
- Assume some other specific distribution for $X_j|Y = k$.

Estimating f_{kj}: Nonparametric Approach

- Use a non-parametric estimate for f_{jk}.
  - Simple approach: construct histograms for the jth predictor in each class. Estimate $f_{kj}(x_j)$ as the proportion of the training observations in the kth class that belong to the same histogram bin as x_j.
  - Can also use a *kernel density estimator*, which is essentially the smoothed version of the above.

Estimating f_{kj} Qualitative Predictors

- If X_j is qualitative, fond the proportion of training observations for the jth predictor corresponding to each class.
  - That is, if the jth predictor takes on the value 1 in 50 of 100 times it appears in the data, estimate

$$\hat{f}_{kj}(x_j) = 0.5 \quad \text{if } x_j = 1$$

Example: Default Data

```
library(ISLR2)
data(Default)
set.seed(1)

train.ind <- sample(1:nrow(Default), floor(0.8*nrow(Default)), replace=F)
def.train <- Default[train.ind,]
def.test <- Default[-train.ind,]
```

Example: Default

```r
require(e1071)
```

```
## Loading required package: e1071
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
mod1 <- naiveBayes(default ~ ., def.train)
predval <- predict(mod1, def.test)
actual <- def.test$default
mean(predval==actual)
```

```
## [1] 0.972
```

Note: the corresponding LDA classifier had 0.9705

Example: Default Data Confusion Matrix

```r
table(predval, actual)
```

```
##         actual
## predval    No  Yes
##      No  1926   52
##      Yes    4   18
```