Have a look at the paper "ada: An R Package for Stochastic Boosting" by Mark Culp, Kjell Johnson, and George Michailidis that appeared in the *Journal of Statistical Software*. It provides perspective and help with both the R packages *ada* and *randomforest*.

27. Use both an adaBoost algorithm and a random forest to identify appropriate classifiers based on the Ripley dataset. For a fine grid of points in $[-1.5, 1.0] \times [0.2, 1.3]$, indicate on a 2-D plot which points get classified to classes $-1$ and $1$ so that you can make visual comparisons to the SVM classifiers referred to in problem 26.

I use defaults for all classifiers including `svm()` for SVM, `ada()` for adaBoost, and `randomForest()` for random forest. The followings are plots and code. Without any tuning process, it is very difficult to say which one is better than the others.

```
l library(e1071)
l library(ada)
l library(randomForest)

da <- read.table("synth.tr", sep = "", quote = "", header = TRUE)
da$yc <- da$yc * 2 - 1
set.seed(1234)
ret.svm <- svm(as.factor(yc) ~ ., data = da)
ret.ada <- ada(as.factor(yc) ~ ., data = da)
ret.rf <- randomForest(as.factor(yc) ~ ., data = da)

### Establish grids for classification.
N.grid <- 25
x.grid <- seq(-1.5, 1.0, length = N.grid)
y.grid <- seq(-0.2, 1.3, length = N.grid)
```
da.grid <- data.frame(xs = rep(x.grid, N.grid), ys = rep(y.grid, rep(N.grid, N.grid)))

f.hat.svm <- as.numeric(predict(ret.svm, da.grid)) * 2 - 3
f.hat.adaBoost <- as.numeric(predict(ret.ada, da.grid)) * 2 - 3
f.hat.random.forest <- as.numeric(predict(ret.rf, da.grid)) * 2 - 3

par(mfrow = c(1, 3))
for(i.method in c("svm", "adaBoost", "random.forest")){
eval(parse(text = paste("f.hat <- f.hat.", i.method, sep = "")))
plot(da.grid$xs, da.grid$ys, col = f.hat + 3, pch = 19, cex = 0.3,
    xlab = "xs", ylab = "ys", main = i.method)
contour(x.grid, y.grid, matrix(f.hat, nrow = N.grid),
    nlevels = 1, labels = "", add = TRUE)
points(da$xs, da$ys, pch = (da$yc + 1) / 2 + 1)
}

28. The Culp et al. paper discusses using a one-versus-all strategy to move adaBoost to a multiclass problem. Apparently, this is known elsewhere as the "adaBoost.mh" algorithm. Return to the use of the first two canonical coordinates of the vowel training data (as in problem 21). Find both an appropriate random forest classifier and an adaBoost.mh classifier for the 11-class problem with \( p = 2 \), and once more show how the classifiers break the 2-D space up into regions to be compared to the plots in problem 20.

The code for adaBoost.mh and random forest for 11-class problem is given at the end. The following tables show the number of missclassification for each class where the row \( Y \) is the true and the column is the predictor. The following plots show the results which are not quite clear since the error for each class is small and not able to distinguish.

\[\begin{array}{ccccccccccc}
\text{Y} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
1 & 38 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\
2 & 6 & 37 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 4 & 43 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 41 & 0 & 5 & 0 & 0 & 0 & 0 & 2 \\
5 & 0 & 0 & 0 & 39 & 7 & 1 & 0 & 0 & 0 & 1 & 0 \\
6 & 0 & 1 & 1 & 2 & 2 & 36 & 0 & 0 & 0 & 0 & 6 \\
7 & 0 & 0 & 0 & 1 & 2 & 1 & 43 & 0 & 1 & 0 & 0 \\
8 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 42 & 3 & 1 & 0 \\
9 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 3 & 39 & 4 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 2 & 40 & 0 \\
11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 45 \\
\end{array}\]

\[\begin{array}{ccccccccccc}
\text{Y} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\
1 & 37 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\
2 & 4 & 30 & 10 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \\
3 & 0 & 12 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{array}\]
true

coordinate 1 for training data
coordinate 2 for training data

adaBoost.mh

coordinate 1 for training data
coordinate 2 for training data

random forest

coordinate 1 for training data
coordinate 2 for training data

−4 −2 0 2 4
−6 −4 −2 0 2 4
true
coordinate 1 for training data
coordinate 2 for training data

−4 −2 0 2 4
−6 −4 −2 0 2 4
random forest
coordinate 1 for training data
coordinate 2 for training data
vowel <- read.table("vowel.train", header = TRUE, sep = ",", quote = """)
da <- vowel[, -1]
Y <- da[, 1]
X <- da[, -1]
color <- c("#60FFFF", ","B52000", ","FF99FF", ","#20B500", ","FFCA00", ","red", "green", "blue", "grey75", ","#FFB2B2", "#928F52")

### LDA to get first two coordinates
library(MASS)
LDA <- lda(y ~ ., data = da)
X.pred <- predict(LDA, newdata = X)
X.1 <- X.pred$x[, 1]
X.2 <- X.pred$x[, 2]
(error.LDA <- mean(as.numeric(X.pred$class) != Y))

### Make grids
X.new <- X.pred$x
x.lim <- range(X.new[, 1])
y.lim <- range(X.new[, 2])
x.grid <- 100
y.grid <- 100
grid <- data.frame(X.1 = seq(x.lim[1], x.lim[2], length = x.grid),
X.2 = rep(seq(y.lim[1], y.lim[2], length = y.grid), each = x.grid))

### adaBoost.mh
library(ada)
set.seed(1234)
ret.ada <- NULL
for(i in 1:11){
Y.new <- (Y == i) * 2 - 1
ret <- ada(as.factor(Y.new) ~ X.1 + X.2)
pred.ada <- predict(ret, newdata = grid, type = "F")
ret.ada <- cbind(ret.ada, pred.ada)
}
grid.class <- apply(ret.ada, 1, which.max)
plot(X.new[, 1], X.new[, 2], col = color[Y], type = "n", main = "Classified by adaBoost.mh",
     xlab = "Coordinate 1", ylab = "Coordinate 2")
points(grid[, 1], grid[, 2], col = color[grid.class], pch = 19, cex = 0.3)
points(X.new[, 1], X.new[, 2], pch = Y + 1, cex = 0.8)

### random forest
library(randomForest)
set.seed(1234)
ret.rf <- randomForest(as.factor(Y) ~ X.1 + X.2)
grid.class <- as.numeric(predict(ret.rf, grid))
plot(X.new[, 1], X.new[, 2], col = color[Y], type = "n", main = "Classified by randomForest",
     xlab = "Coordinate 1", ylab = "Coordinate 2")
points(grid[, 1], grid[, 2], col = color[grid.class], pch = 19, cex = 0.3)
points(X.new[, 1], X.new[, 2], pch = Y + 1, cex = 0.8)

### compare
par(mfrow = c(2, 2))
plot(X.1, X.2, col = color[Y],
     pch = Y + 1, main = "true",
     xlab = "Coordinate 1 for Training Data", ylab = "Coordinate 2 for Training Data")
plot(X.1, X.2, col = color[f.hat.adaBoost.mh],
     pch = f.hat.adaBoost.mh + 1, main = "adaBoost.mh",
     xlab = "Coordinate 1 for Training Data", ylab = "Coordinate 2 for Training Data")
plot(X.1, X.2, col = color[f.hat.random.forest],
     pch = f.hat.random.forest + 1, main = "random forest",
     xlab = "Coordinate 1 for Training Data", ylab = "Coordinate 2 for Training Data")

### show error
table(Y, f.hat.adaBoost.mh)
table(Y, f.hat.random.forest)