MNIST Image Processing with GLM Binomial

Accuracy with Rpart, randomForest and Rborist

Sang Kim

```
library(dslabs)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
---------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------------
----------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(Rborist)
```

```
## Warning: package 'Rborist' was built under R version 3.5.3
```

```
## Rborist 0.1-17
```

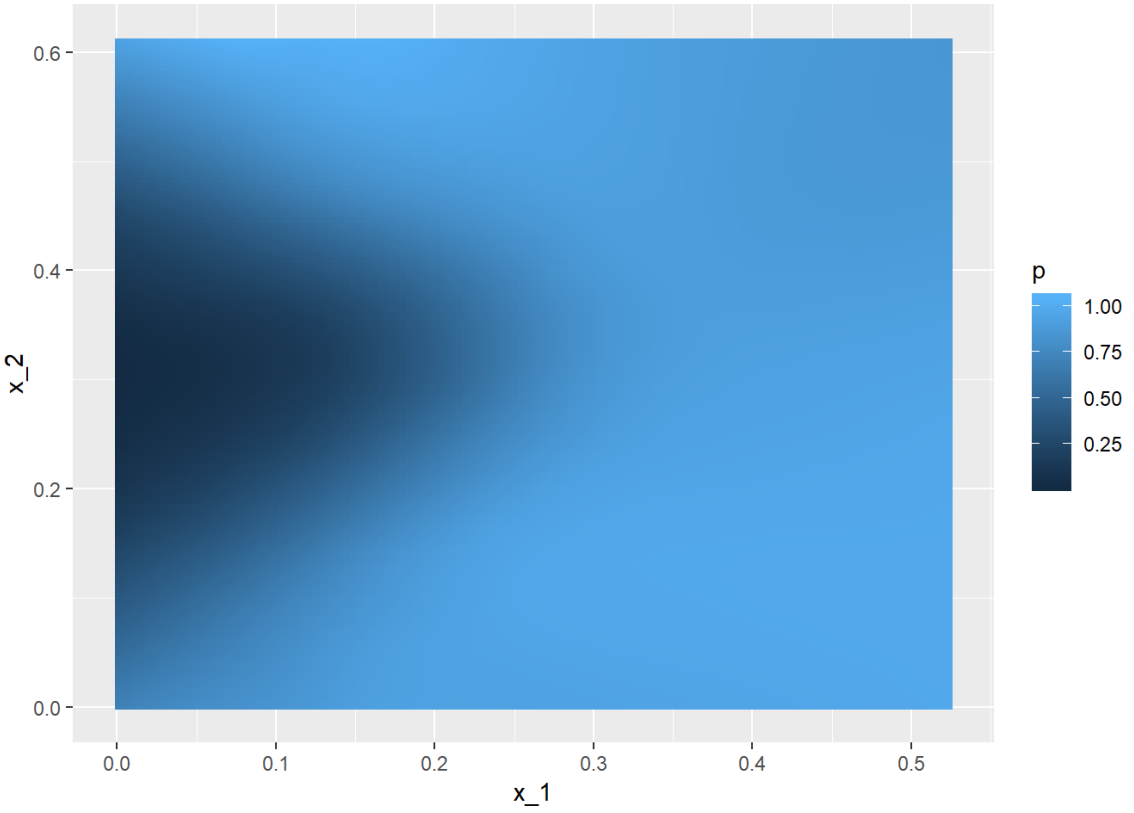
```
## Type RboristNews() to see new features/changes/bug fixes.
```

```
data("mnist_27")

mnist_27$train %>% ggplot(aes(x_1, x_2, color=y)) +
  geom_point()
```

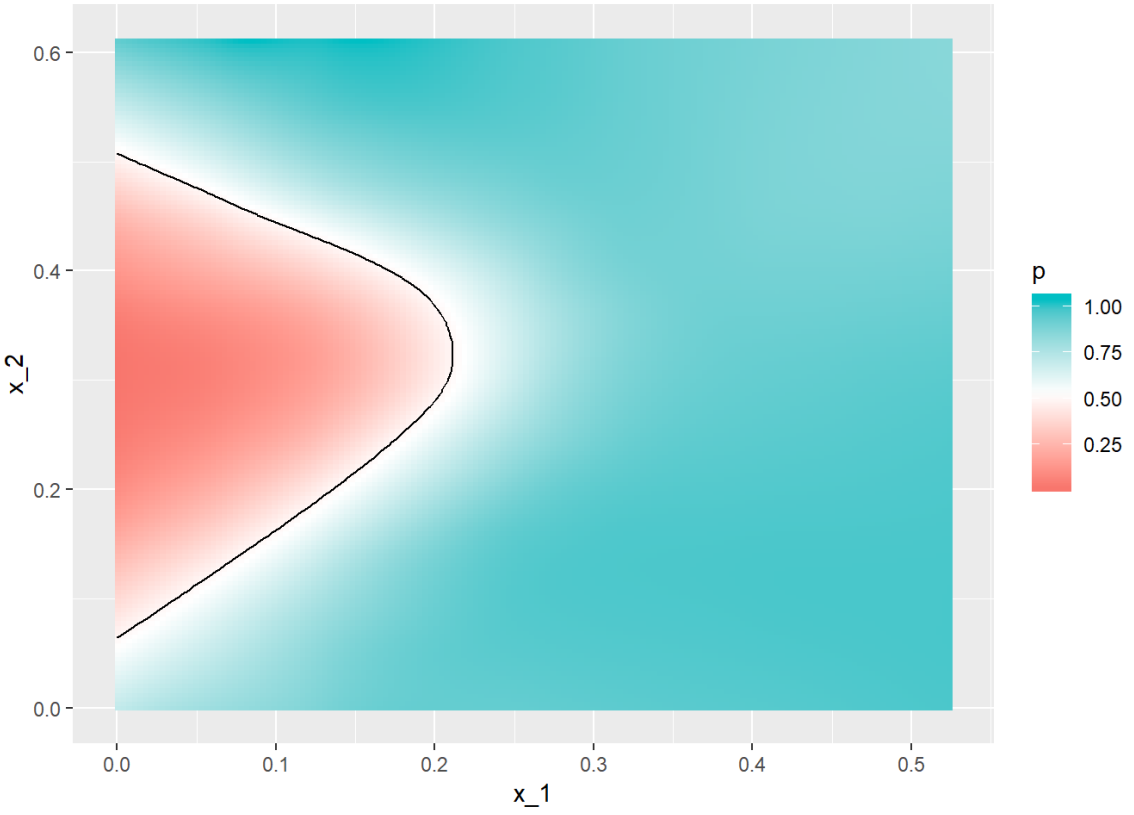

```
fit <- glm(y ~ x_1 + x_2, data=mnist_27$train, family="binomial")

p_hat <- predict(fit, newdata = mnist_27$test)

y_hat <- factor(ifelse(p_hat > 0.5, 7, 2))

confusionMatrix(data=y_hat, reference = mnist_27$test$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  2  7
##          2 92 34
##          7 14 60
##
##                Accuracy : 0.76
##                  95% CI : (0.6947, 0.8174)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 1.668e-11
##
##                   Kappa : 0.5124
##  Mcnemar's Test P-Value : 0.006099
##
##             Sensitivity : 0.8679
##             Specificity : 0.6383
##          Pos Pred Value : 0.7302
##          Neg Pred Value : 0.8108
##              Prevalence : 0.5300
##          Detection Rate : 0.4600
##    Detection Prevalence : 0.6300
##       Balanced Accuracy : 0.7531
##
##        'Positive' Class : 2
##
```

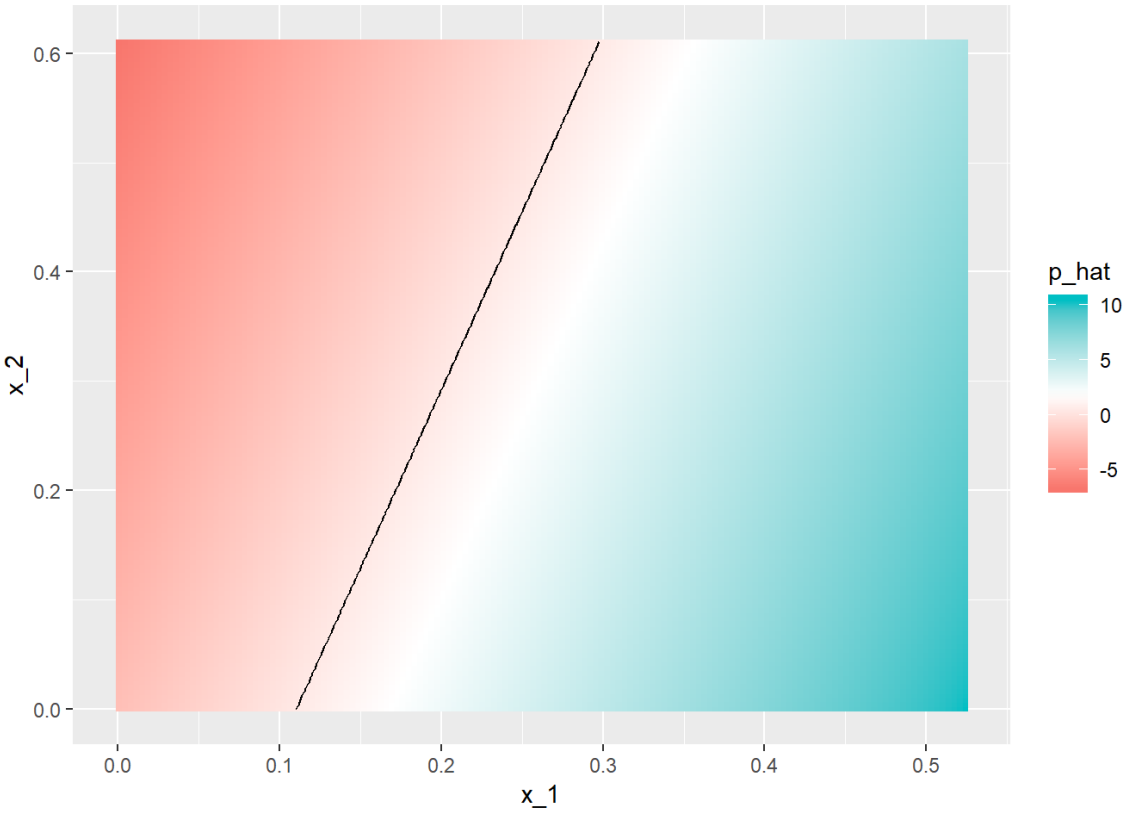
```
mnist_27$true_p %>% ggplot(aes(x_1, x_2, fill=p)) +
  geom_raster()
```



```
mnist_27$true_p %>% ggplot(aes(x_1, x_2, z = p, fill=p)) +
  geom_raster() +
  scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4")) +
  stat_contour(breaks=c(0.5), color="black")
```




```
p_hat <- predict(fit, newdata = mnist_27$true_p)
mnist_27$true_p %>% mutate(p_hat = p_hat) %>%
  ggplot(aes(x_1, x_2, z=p_hat, fill=p_hat)) +
  geom_raster() +
  scale_fill_gradientn(colors=c("#F8766D","white","#00BFC4")) +
  stat_contour(breaks=c(0.5),color="black")
```




```
p_hat <- predict(fit, newdata = mnist_27$true_p)
mnist_27$true_p %>% mutate(p_hat = p_hat) %>%
  ggplot() +
  stat_contour(aes(x_1, x_2, z=p_hat),breaks=c(0.5), color="black") +
  geom_point(mapping = aes(x_1, x_2, color=y), data=mnist_27$test)
```

```
# Classification(decision Tree)
# minimize training error within the partitions. - Gini index and Entropy
train_rpart <- train(y ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0, 0.05, len = 25)),
                     data = mnist_27$train)

conf <- confusionMatrix(predict(train_rpart, mnist_27$test), mnist_27$test$y)$overall["Accuracy"]

conf
```

```
## Accuracy
##     0.82
```

```
#library(randomForest)
train_rf <- randomForest(y ~ ., data=mnist_27$train)

confusionMatrix(predict(train_rf, mnist_27$test),
                mnist_27$test$y)$overall["Accuracy"]
```

```
## Accuracy
##    0.795
```

```
#plot_cond_prob(predict(train_rf, mnist_27$true_p, type = "prob")[,2])

fit <- train(y ~ .,
             method = "Rborist",
             tuneGrid = data.frame(predFixed = 2, minNode = c(3,50)),
             data = mnist_27$train)
confusionMatrix(predict(fit, mnist_27$test), mnist_27$test$y)$overall["Accuracy"]
```

```
## Accuracy
##      0.8
```

```
minNode <- seq(25, 100,25)

fit <- train(y ~ .,
             method = "Rborist",
             tuneGrid = data.frame(predFixed = 2, minNode = minNode),
             data = mnist_27$train)

fit$bestTune
```

```
##   predFixed minNode
## 1         2      25
```

```
fit$results$Accuracy
```

```
## [1] 0.8331187 0.8297824 0.8227853 0.8193121
```

```
confusionMatrix(predict(fit, mnist_27$test), mnist_27$test$y)$overall["Accuracy"]
```

```
## Accuracy
##     0.82
```

```
# a random selection of features to split when deciding on partitions. mtry
#  function varImp that extracts variable importnance from any model in which the calculation is implemented.
```