DATA 1010
Problem Set 11
Due 30 November 2018 at 11 PM

Problem 1

Show that for each $\alpha \in [0,1]$, there exists $t \in [0,\infty]$ such that such that the likelihood ratio classifier h_t is the function $h : \mathcal{X} \to \mathcal{Y}$ which minimizes

$$L(h) = \alpha\,\mathbb{P}(h(X) = +1 \text{ and } Y = -1) + (1-\alpha)\mathbb{P}(h(X) = -1 \text{ and } Y = +1).$$

(a) Identify the relationship between α and its corresponding t value. (For simplicity, assume that $\mathcal{X}$ is finite.)

Hint: write $L(h)$ as a sum over the elements $x \in \mathcal{X}$. For each x, consider the resulting contribution to that sum if $h(x) = +1$, and similarly for $h(x) = -1$. Classify each x according to which of the two contributions is smaller.

(b) Determine and explain the motivation for this problem.

Solution

(a) We begin by writing L as

$$L(h) = \sum_{\mathbf{x}\in\mathcal{X}} \left(\alpha p_{-1} f_{-1}(\mathbf{x}) + (1-\alpha) p_{+1} f_{+1}(\mathbf{x})\right).$$

For each $\mathbf{x} \in \mathcal{X}$, classifying it as $+1$ contributes $\alpha p_{-1} f_{-1}(\mathbf{x})$ to this sum, while classifying it as -1 contributes $(1-\alpha)p_{-1}f_{-1}(\mathbf{x})$. Since each of these contributions can be minimized independently of the others, the overall minimum h is the one that minimizes each contribution. So the minimizing h is

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \alpha p_{-1} f_{-1}(\mathbf{x}) \le (1-\alpha) p_{-1} f_{-1}(\mathbf{x}) \\ -1 & \text{otherwise.} \end{cases}$$

This is equivalent to the likelihood ratio classifier h_t with $t = \frac{\alpha p_{-1}}{(1-\alpha)p_{+1}}$.

(b) The loss function is a generalization of the loss function that that Bayes classifier minimizes, since setting $\alpha = \frac{1}{2}$ weights the two types of misclassification equally. Varying α allows us to weight the two misclassification probabilities differently. So this exercise shows that we could have obtained the likelihood ratio classifier beginning with the intuitive idea of a loss function with different weights for the two misclassification probabilities.

Problem 2

(a) Consider the coordinates of n points in $\mathbb{R}^p$, organized into an $n \times p$ matrix A. Suppose that `U, Σ, V = svd(A .- mean(A, dims=1))`, and explain why `V[:,1:k]'` is the matrix which maps each point in $\mathbb{R}^p$ to its coordinates in the subspace of $\mathbb{R}^p$ spanned by the columns of `V[:,1:k]`.

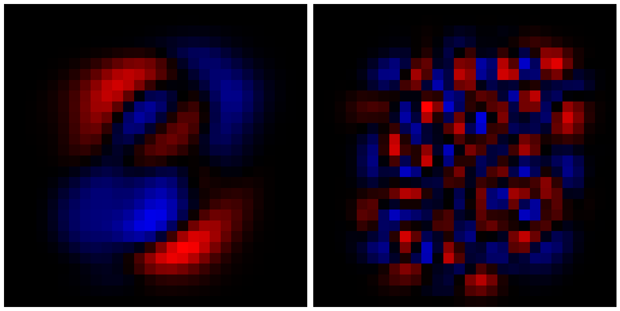
(b) Plot an image of the *third* principal component for the MNIST dataset. Identify a digit which you think should predominantly have a large or small dot product with this image, and make a scatter plot of which shows the dot product with the first principal component on the x-axis and the dot product with the third principal component on the y-axis. Check whether your prediction was accurate.

(c) What do you think the 100th principal component might look like, compared to the first few? Display it and check your prediction.

Solution

(a) The component of a given vector $\mathbf{x}$ in the direction of the first column of V is obtained by dotting the $\mathbf{x}$ with the first column of V. Since $V'\mathbf{x}$ yields the dot product of the vector with each column of V (by the definition of matrix multiplication), the components of $V'\mathbf{x}$ give the coordinates of $\mathbf{x}$ with respect to the columns of V. In particular, the projection of $\mathbf{x}$ onto the first k columns of V is equal to the linear combination of those columns with weights given by the components of $V'\mathbf{x}$.

(b) My prediction is that 2's are going to be generally near the middle, since most of them will catch a roughly equal amount of each color. This turns out to be reasonably accurate. The third and hundredth principal components:



(c) One might guess that it would be noisier than the first few, with lots of small negative and positive splotches. That prediction is accurate (see figure above).

Problem 3

(a) Write an R function called `makelabels` which takes a vector of postive integers and returns a vector of strings with "label" prepended to the string representation of each integer:

R

```r
makelabels(c(4,5,7)) == c('label4','label5','label7')
```

(b) Write an R function called `numzeros` which accepts a vector as an argument and returns the number of zeros in the vector.

R

```r
numzeros(c(-1,0,2,3,0,1)) == 2
```

(c) Write an R function called `numincreasing` which accepts a vector as an argument and the number of components of that vector which are greater than the immediately preceding component.

R

```r
numincreasing(c(-1,0,2,3,0,1)) == 4
```

Solution

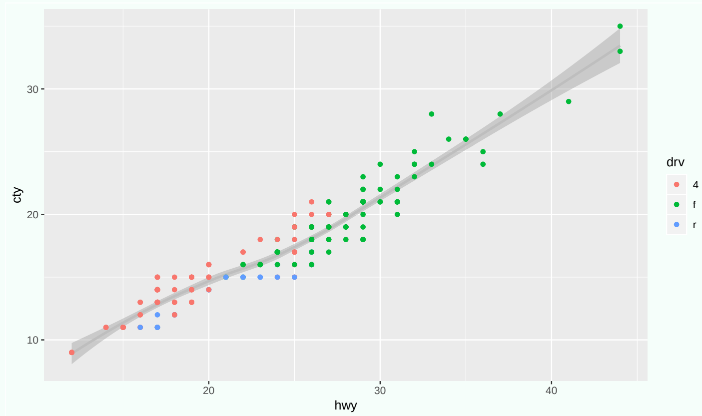
(a) `makelabels <- function(v) {sapply(v,function (x) {paste('label',x,sep='')})}`

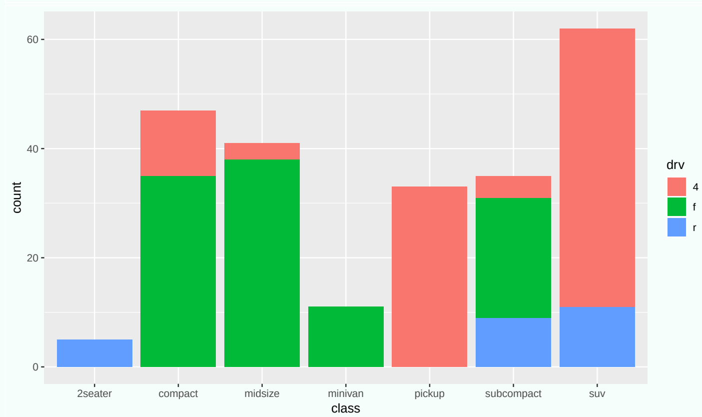
(b) `numzeros <- function(v) {sum(v == 0)}`

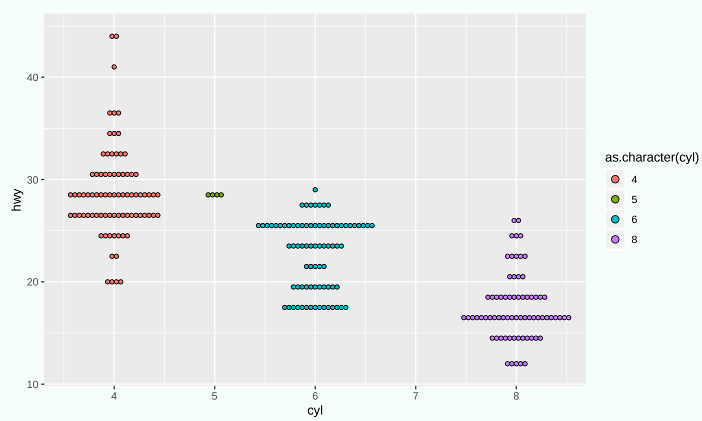
(c) `numincreasing <- function(v) {sum(v[2:length(v)] > v[1:length(v)-1])}`

Problem 4

Use `ggplot2` to reproduce each of the following graphs. The dataset used is `mpg`, which is automatically loaded when you run `library(tidyverse)`.







Solution

R

```r
library(tidyverse)

ggplot(data = mpg) +
  geom_smooth(mapping = aes(x=hwy,y=cty), color = 'gray') +
  geom_point(mapping = aes(x=hwy,y=cty,color=drv))

ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class, fill = drv))

ggplot(data = mpg) +
  geom_dotplot(aes(x = cyl, y = hwy, fill = as.character(cyl)),
               binaxis = 'y', stackdir = 'center' , dotsize=0.4)
```

Problem 5

Write `dplyr` code to perform each of the following operations on the `mpg` dataset. We say "average mpg" to mean the $\frac{1}{2}$ times the sum of the highway and city mpg recorded for each vehicle.

(i) Return a dataframe containing only the Audis with an average mpg of at least 24.

(ii) Return a dataframe with all of the cars sorted in decreasing order of average miles per gallon.

(iii) Return a dataframe with just the `trans` and `hwy` columns for all of the Volkswagens.

(iv) Return a dataframe with a new column containing each vehicle's average miles per gallon.

(v) Return a dataframe showing the average highway miles per gallon and average city miles per gallon for each manufacturer.

Solution

R

```r
# (i)
mpg %>% filter(manufacturer == 'audi', hwy+cty > 48)
# (ii)
mpg %>% arrange(desc(hwy+cty))
# (iii)
mpg %>%
 filter(manufacturer == 'audi') %>%
 select(trans, hwy)
# (iv)
mpg %>% mutate(avg_mpg = (hwy+cty)/2)
# (v)
mpg %>% group_by(manufacturer) %>%
  summarize(mean(hwy,na.rm=TRUE),mean(cty,no.rm=TRUE))
```