SL_EX1

roi hezkiyahu

28 10 2022

```
library(dplyr)
library(ggplot2)
library(purrr)
library(caret)
library(class)
```

Q1

1. **Population Optimizer of absolute loss**
   Prove that for absolute loss: $L_{\text{abs}}(Y, f(X)) = |Y - f(X)|$, EPE is minimized by setting $f^*(x) = \text{Median}(Y|X = x)$
   Hint: you may find the following identity useful:

$$\int_{y>c} (y-c)dP(y) = \int_{y>c} Pr(Y > y)dy$$

   (a) **Generalization to quantile loss** The τth quantile loss for $0 < \tau < 1$ is defined as:

$$L_\tau(Y, f(X)) = \begin{cases} \tau \times (Y - f(X)) & \text{if } Y - f(X) > 0 \\ -(1-\tau) \times (Y - f(X)) & \text{otherwise} \end{cases}$$

   Prove that the EPE is minimized by setting $f^*(x)$ to be the τth quantile of $P(Y|X = x)$, i.e., $P(Y \leq f^*(x)|X = x) = \tau$

notice that the median is a specific case for the quantile loss with $\tau = 0.5$ so proving for the general case will cover both questions

1a

$$\frac{\partial E_{Y|X}[(L_\tau(Y, f(X)))|X = x]}{\partial f(x)} = \frac{\partial}{\partial f(x)}[\int_{min_y}^{f(x)} (1-\tau) f_Y(y)dy + \int_{f(x)}^{max_y} -\tau f_Y(y)dy] = F_Y(f(x)) - \tau F_Y(f(x)) - \tau + \tau F_Y(f(x)) =$$
$$= F_Y(f(x)) - \tau = 0 \iff f(x) = F_y^{-1}(\tau)$$
$$\text{thus the minimizer is the } \tau th\ qunatile\ of P(Y|X = x)$$

Q2

2. **ESL 2.3:** Derive equation (2.24) (expected median distance to origin's nearest neighbor in an ℓ_p ball):

$$d(p, n) = (1 - \frac{1}{2}^{1/n})^{1/p}$$

   Suggested approach:

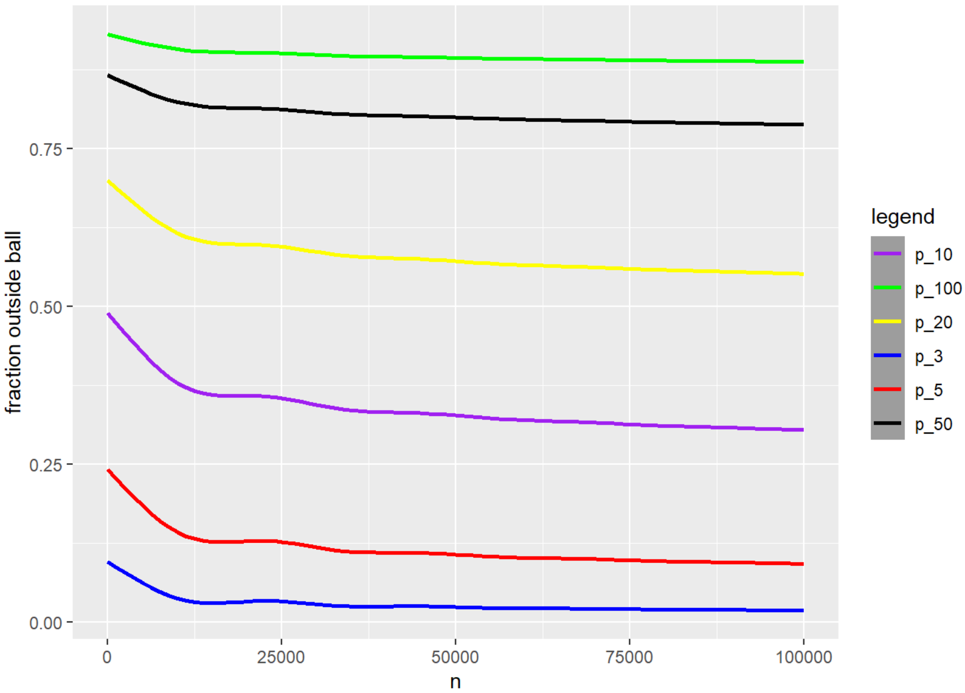
   (a) Find the probability that all observations are outside a ball of radius $r < 1$, as a function of r.
   (b) You are looking for r such that this probability is $1/2$.

   Plot $d(p, n)$ against p for $n \in \{100, 5000, 100000\}$ and $p \in \{3, 5, 10, 20, 50, 100\}$ (make one curve for every value of n — use the R functions plot() and lines()) and interpret the graph.

$$denote\ D_c = min(||X_1||, \dots, ||X_n||)$$
$$P(\text{all observations are outside a ball of radius r}) = P(D_c > r)$$
$$P(D_c > r) = \Pi_{i=1}^n P(||X_i|| > r) = P(||X_1|| > r)^n = (1 - P(||X_1|| < r))^n = (1 - r^p)^n$$
$$\text{we are looking for such r such that } P(D_c > r) = \frac{1}{2} \Rightarrow (1 - r^p)^n = \frac{1}{2} \Rightarrow (1 - \frac{1}{2^{1/n}})^{1/p} = r$$

```
radius_function <- function(p){
  n <- 1:100000
  return( (1 - 1/2^(1/n))^(1/p))
}
p_s <- c(3,5,10,20,50,100)
df <- map(p_s,radius_function)
df <- as.data.frame(df)
colnames(df) <- paste0("p_",p_s)
df["n"] = 1:100000
colors = c("p_3"="blue", "p_5" ="red","p_10"="purple","p_20"="yellow", "p_50"="black","p_100"="green")
ggplot(data = df) +
  geom_smooth(aes(x = n, y = p_3,color = "p_3")) +
  geom_smooth(aes(x = n, y = p_5,color = "p_5")) +
  geom_smooth(aes(x = n, y = p_10,color = "p_10")) +
  geom_smooth(aes(x = n, y = p_20,color = "p_20")) +
  geom_smooth(aes(x = n, y = p_50,color = "p_50")) +
  geom_smooth(aes(x = n, y = p_100,color = "p_100")) +
  labs(x="n", y = "fraction outside ball", color = "legend") +
  scale_color_manual(values = colors)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



we can see that an p grows the fraction of observations outsde the ball increaces

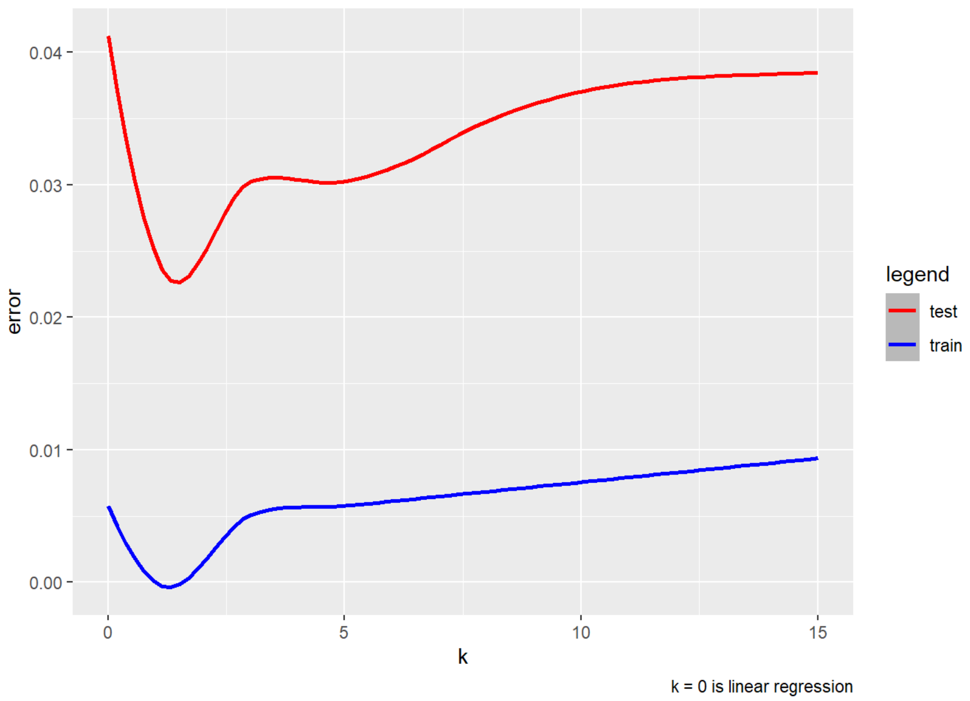
Q3

3. **ESL 2.7:** Compare classification performance of k-NN and linear regression on the zipcode[1] data, on the task of separating the digits 2 and 3. Use $k \in \{1, 3, 5, 7, 15\}$. Plot training and test error for k-NN choices and linear regression. Comment on the shape of the graph.

```
acc_err <- function(y_true,y_pred){
  return (1-mean(y_true == y_pred))
}

df_train <- read.table("zip.train") %>% rename("y" = "V1") %>%
  filter(y %in% c(2,3)) %>% mutate(y = y-2)
df_test <- read.table("zip.test") %>% rename("y" = "V1") %>%
  filter(y %in% c(2,3)) %>% mutate(y = y-2)

get_knn_res <- function(k,df_train,df_test){
  X_tr <- as.matrix(df_train %>% select(-y))
  X_te <- as.matrix(df_test %>% select(-y))
  y_tr <- df_train %>% select(y)
  train_preds <- knn(train = X_tr,test = X_tr,cl = as.matrix(y_tr),k = k)
  test_preds <- knn(train = X_tr,test = X_te,cl = as.matrix(y_tr),k = k)
  train_err <- acc_err(df_train$y,train_preds)
  test_err <- acc_err(df_test$y,test_preds)
  return( c(train_err, test_err))
}

zip_lm <- lm(y~.,data = df_train)
lr_train_err <- acc_err(df_train$y,(predict(zip_lm,df_train) > 0.5)*1)
lr_test_err <- acc_err(df_test$y,(predict(zip_lm,df_test) > 0.5)*1)
k_s = c(1,3,5,7,15)
knn_train_res <- c()
knn_test_res <- c()
for (i in 1:length(k_s)){
  knn_res <- get_knn_res(k=k_s[i],df_train,df_test)
  knn_train_res <- c(knn_train_res,knn_res[1])
  knn_test_res <- c(knn_test_res,knn_res[2])
}
result_df <- tibble(k = c(0,k_s),train_res = c(lr_train_err,knn_train_res) ,test_res = c(lr_test_err,knn_test_res))
colors = c("train"="blue", "test" ="red")
ggplot(data = result_df) +
  geom_smooth(aes(x = k, y = train_res,color = "train")) +
  geom_smooth(aes(x = k, y = test_res,color = "test")) +
  labs(x="k", y = "error", color = "legend", caption = "k = 0 is linear regression") +
  scale_color_manual(values = colors)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```



we can see that as k increaces the knn model error rate increace as well in the train and test set.
we can also see that linear regression has the worst test results

Q4

4. **ESL 2.9 (second edition only)** Consider a linear regression model, fit by least squares to a set of training examples $T = \{(X_1, Y_1), ..., (X_N, Y_N)\}$, drawn i.i.d from some population. Let $\hat{\beta}$ be the least

squares estimate. Suppose we also have some other ("test") data drawn independently from the same distribution $\{(\tilde{X}_1, \tilde{Y}_1), ..., (\tilde{X}_M, \tilde{Y}_M)\}$. Prove that:

$$\frac{1}{N}\mathbb{E}(\sum_{i=1}^{N}(Y_i - X_i^T\hat{\beta})^2) \leq \frac{1}{M}\mathbb{E}(\sum_{i=1}^{M}(\tilde{Y}_i - \tilde{X}_i^T\hat{\beta})^2),$$

that is, the expected squared error in-sample is always bigger than out of sample in least squares fitting. Note that the values X are also random variables here, and the expectation is over everything that is random, including X, Y and $\hat{\beta}$.
Hint: There are several ways to prove this. One starts from considering the best possible linear model we derived in class:

$$\beta^* = (E(XX^T))^{-1}E(XY),$$

and comparing both sides to it.
Note: Students who find more than one valid way to prove the result will get a bonus grade.

* **Extra credit problem: Optimality of k-NN in fixed dimension**
Assume $X \sim \text{Unif}([0,1]^p)$, and $Y = f(X) + \epsilon$ with $\epsilon \sim (0, \sigma^2)$ (that is, $f(x) = E(Y|X = x)$).
Assume f is Lipschitz: $\|x_1 - x_2\| < \delta \Rightarrow |f(x_1) - f(x_2)| < c\delta$, $\forall x_1, x_2 \in [0,1]^p$. Choose any sequence $k(n)$ such that:

$$k(n) \overset{n\to\infty}{\longrightarrow} \infty$$
$$k(n)/n \overset{n\to\infty}{\longrightarrow} 0$$

Then:

$$\text{EPE(k-NN using } k(n)) \overset{n\to\infty}{\longrightarrow} EPE(f) = \sigma^2$$

(The proof does not have to be completely formal, for example you can replace a binomial with its normal approximation without proof of the relevant asymptotics).

the expected error is the same for all obesravtions so we can assume M=N=1
$denote\ E_{tr}, E_{te}$ the expected train and test error
$E((\tilde{Y} - \tilde{X}\hat{\beta})^2) \geq E((\tilde{Y} - \tilde{X}\tilde{\beta})^2)$ ($\tilde{\beta}$ being LSE for the test set) $\Rightarrow$
$E_{te} \geq E((\tilde{Y} - \tilde{X}\tilde{\beta})^2)$
$E((\tilde{Y} - \tilde{X}\tilde{\beta})^2) = E((Y - X\hat{\beta})^2)$ (its an expected value thus the point of estimate doesn't matter)
to conclude we get:
$E_{te} = E((\tilde{Y} - \tilde{X}\hat{\beta})^2) \geq E((\tilde{Y} - \tilde{X}\tilde{\beta})^2) = E((Y - X\hat{\beta})^2) = E_{tr}$

try #2, i am not completly sure about the varinace claim here
the expected error is the same for all obesravtions so we can assume M=N=1
$denote\ E_{tr}, E_{te}$ the expected train and test error
$E_{tr} = \sigma^2 + bias(f(x)) + var(f(x)) = \sigma^2 + var(f(x))$ ($\hat{\beta}$ is unbaised)
$E_{te} = \sigma^2 + bias(f(\tilde{x})) + var(f(\tilde{x}))$
$var(f(\tilde{x})) = var(f(x)),\ and\ bias(f(\tilde{x})) \geq 0$
thus we get:$E_{te} \geq E_{tr}$

Extra

$let\ x_0 \in X$ be a fixed point

$\forall\varepsilon\ \forall k\ \exists m_k\ ; \forall n > m_k, \quad ||x_0 - x_j|| < \varepsilon_k\ \forall x_j \in N_k(x_0)$
thus form f being lipschitz :$|f(x_0) - f(x_1)| < \varepsilon_k\delta$

$$\text{thus } |\hat{f}\,(x_0) - f(x_0)| = |\sum_{x_j \in N_k(x_0)} \frac{f(x_j)}{k} - f(x_0)| = |\sum_{x_j \in N_k(x_0)} \frac{f(x_j) - f(x_0) + f(x_0)}{k} - f(x_0)| \leq$$

$$\leq \sum_{x_j \in N_k(x_0)} \frac{|f(x_j) - f(x_0)|}{k} + |\sum_{x_j \in N_k(x_0)} \frac{f(x_0)}{k} - f(x_0)| = \sum_{x_j \in N_k(x_0)} \frac{|f(x_j) - f(x_0)|}{k} \leq \varepsilon_k \delta$$

$$\text{thus for } \varepsilon_k \xrightarrow{n \to \infty} 0 : bias(\hat{f}\,(x_0)) \xrightarrow{n \to \infty} 0$$

$$V(E(\hat{f}\,(x_0))) = V(\sum_{x_j \in N_k(x_0)} \frac{E(f(x_j))}{k}) \xrightarrow{n \to \infty} V(\sum_{x_j \in N_k(x_0)} \frac{f(x_0)}{k}) = \sum_{x_j \in N_k(x_0)} \frac{V(f(x_0))}{k} = 0$$

we get that the bais and variance terms both converge to zero as n goes to ∞ so from the prediction error decomposition we get tha

$$EPE(\text{k-nn using k(n)}) \xrightarrow{n \to \infty} \sigma^2$$