Homework 7

Serena Zhang

4/14/2024

Recall that in class we showed that for randomized response differential privacy based on a fair coin (that is a coin that lands heads up with probability 0.5), the estimated proportion of incriminating observations $\hat{P}$ [1] was given by $\hat{P} = 2\pi - \frac{1}{2}$ where π is the proportion of people answering affirmative to the incriminating question.

I want you to generalize this result for a potentially biased coin. That is, for a differentially private mechanism that uses a coin landing heads up with probability $0 \leq \theta \leq 1$, find an estimate $\hat{P}$ for the proportion of incriminating observations. This expression should be in terms of θ and π.

Student Answer
p(heads) $= \theta$
$\pi = \theta\hat{p} + (1-\theta)\theta$
$\hat{p} = \frac{\pi-(1-\theta)\theta}{\theta}$

Next, show that this expression reduces to our result from class in the special case where $\theta = \frac{1}{2}$.

Student Answer
$\hat{p} = \frac{\pi-(1-\theta)\theta}{\theta}$
$\hat{p} = \frac{\pi-(1-1/2)1/2}{1/2}$
$\hat{p} = \frac{\pi-1/4}{1/2} * \frac{2}{2}$
$\hat{p} = 2\pi - \frac{1}{2}$

Part of having an explainable model is being able to implement the algorithm from scratch. Let's try and do this with `KNN`. Write a function entitled `chebychev` that takes in two vectors and outputs the Chebychev or L^∞ distance between said vectors. I will test your function on two vectors below. Then, write a `nearest_neighbors` function that finds the user specified k nearest neighbors according to a user specified distance function (in this case L^∞) to a user specified data point observation.

```
#student input

#chebychev function
chebychev <- function(vector1, vector2) {
  dist <- max(abs(vector1 - vector2))
  return(dist)
```

[1] in class this was the estimated proportion of students having actually cheated

```
}

# testing chebychev func
x<- c(3,4,5)
y<-c(7,10,1)
chebychev(x,y)

#nearest_neighbors function
nearest_neighbors <- function(data, datapoint, k, distfunc) { # assuming distfunc is chebychev
    distances <- apply(data, MARGIN = 1, function (x) chebychev(x, datapoint)) # 1: cc row-wise
    nearest_indices <- order(distances)[1:k+1] # 1:k+1 to prevent datapoint being included as nn
    nearest_data <- as.data.frame(data[nearest_indices, ])
    nearest_data$Indices <- nearest_indices # creating indices column
    return(nearest_data)
}

# testing nearest_neighbors function

# creating data
X <-c(1,2,3,4,5)
Y <-c(2,10,200,500,1000)
data <- data.frame(unlist(X),unlist(Y))
names(data) = c("X", "Y")
head(data)

datapoint <- c(1,2)

nn = nearest_neighbors(data, datapoint, 3, chebychev) # finding 3 nn to (1,2)
nn
```

Finally create a `knn_classifier` function that takes the nearest neighbors specified from the above functions and assigns a class label based on the mode class label within these nearest neighbors. I will then test your functions by finding the five nearest neighbors to the very last observation in the `iris` dataset according to the `chebychev` distance and classifying this function accordingly.

```
library(class)
df <- data(iris)
#student input

knn_classifier <- function(nearest_neighbors, class_label) {
  neighbor_labels <- nearest_neighbors[[class_label]]
  mode <- which.max(table(neighbor_labels))
  return(mode)
}


#data less last observation
x = iris[1:(nrow(iris)-1),]
#observation to be classified
obs = iris[nrow(iris),]
```

```
#find nearest neighbors
ind = nearest_neighbors(x[,1:4], obs[,1:4],5, chebychev)[[5]]
as.matrix(x[ind,1:4])
knn_classifier(x[ind,], 'Species')
obs[,'Species']
```

Interpret this output. Did you get the correct classification? Also, if you specified $K = 5$, why do you have 7 observations included in the output dataframe?

Student Answer

The ind matrix returns the 5 nearest neighbors' indices, sepal length, sepal width, petal length, and petal width. The knn_classifier function then returns the mode of the species labels among the 5 nearest neighbors. Finally, obs[,'Species'] reveals the actual species of the observation of interest. The KNN classifier returned 'Virginica' as the most common (3/5) species label of the 5 nearest neighbors, which is the correct classification for our observation.

Earlier in this unit we learned about Google's DeepMind assisting in the management of acute kidney injury. Assistance in the health care sector is always welcome, particularly if it benefits the well-being of the patient. Even so, algorithmic assistance necessitates the acquisition and retention of sensitive health care data. With this in mind, who should be privy to this sensitive information? In particular, is data transfer allowed if the company managing the software is subsumed? Should the data be made available to insurance companies who could use this to better calibrate their actuarial risk but also deny care? Stake a position and defend it using principles discussed from the class.

Student Answer

According to the harm principle, the acquisition of sensitive healthcare data should be prevented at the point that it could cause harm to an agent. Therefore, access to patient health data should only be given by explicit informed consent from the patient, or a surrogate if the patient is incapacitated. Data transfer in the case that the company managing the software is subsumed, or to an insurance company, should only be allowed if explicit informed consent is given again. While this process may be more inefficient than immediately allowing data transfer, it makes patients aware of what is happening to their sensitive data and prevents data misuse. Without informed consent given at each step of the process, the potential for harm from data misuse is too high, even if the company using the data has good intentions.

Some may argue from a paternalistic perspective that use of patient data is in order to ultimately help improve patients' well-being, and that no serious harm is caused by using this data. In the case of Google's DeepMind, although the intention to create an algorithm to predict kidney injury may have been good, a company being allowed access to millions of patients' sensitive data without their knowledge is a slippery slope. In addition, the patients used in the training data may not even necessarily benefit from this algorithm in the future. Ultimately, it is important to avoid setting a precedent that could allow other companies to subvert principles of informed consent, especially when their intentions may be less pure.