Hw 7

Hailey Flo

04/19/2024

1

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.

If a differentially private mechanism flips a coin with the probability θ of landing heads up, the result will be truthful if it's heads and a random yes or no if it's tails. That means the probability of the coin landing on tails will be $1 - \theta$. If the result is tails, then the probability of the answer being incriminating will be 1 divided by the number of responses, or $\frac{1}{2} = 0.5$.

Using these proportions we can get the total proportion of affirmative answers and solve for π. This gives us the equation $\pi = \theta\hat{P} + (1 - \theta) * 0.5$, which is the probability of getting heads and answering affirmative plus the probability of getting tails and answering affirmative. Next, we can rewrite this equation to factor out $\theta\hat{P}$. That gives us $\theta\hat{P} = \pi - 0.5 * (1 - \theta)$. And finally separating out $\hat{P}$ results in the equation $\hat{P} = \frac{\pi - 0.5 * (1-\theta)}{\theta}$.

Thus $\hat{P}$ or the proportion of incriminating observations can be expressed as:

$$\hat{P} = \frac{\pi - 0.5 * (1 - \theta)}{\theta}$$

2

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

If $\theta = \frac{1}{2}$, then we can substitute the value of 0.5 into our equation for the variable theta. This gives us:

$$\hat{P} = \frac{\pi - 0.5 * (1 - 0.5)}{0.5}$$

$$\hat{P} = \frac{\pi - 0.5 * 0.5}{0.5}$$

$$\hat{P} = \frac{\pi - 0.25}{0.5}$$

$$\hat{P} = (\pi - 0.25) * 2$$

$$\hat{P} = 2\pi - 0.5$$

The final result when $\theta = \frac{1}{2}$ reduces to: $\hat{P} = 5.783$

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

3

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.

```
library(dplyr)

#chebychev function
chebychev <- function(x, y) {
  distance <- max(abs(x - y))
  return(distance)
}
#nearest_neighbors function

nearest_neighbors <- function(x, obs, k, distance_function) {

  dist = apply(x,1, distance_function,obs)
  distances = sort(dist)[1:k]
  neighbor_ind = which(dist %in% sort(dist)[1:k])

  if(length(neighbor_ind)!= k){
    warning(
      paste('Several variables with equal distance, used k:',length(neighbor_ind))
    )
  }

  neighbors = list(neighbor_ind, distances)
  return(neighbors)
}

x<- c(3,4,5)
y<-c(7,10,1)
chebychev(x,y)
```

4

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, classifier_col) {
    class_labels <- nearest_neighbors[[classifier_col]]
    freq_table <- table(class_labels)
    predicted_class_label <- names(freq_table)[which.max(freq_table)]
    return(predicted_class_label)
}
#data less last observation
```

```r
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)[[1]]
as.matrix(x[ind,1:4])
obs[,1:4]
knn_classifier(x[ind,], 'Species')
obs[,'Species']
```

5

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?

Yes, I got the correct classification of virginica. The species of the last observation is virginica, and the knn classifier function found that the mode classifier for the nearest neighbors of this last observation was virginia, so the classification matches. The reason why there are 7 observations included in the dataframe despite specifying k as 5 is because there were some points with equal distances. The nearest neighbors function could not limit the number of observations to 5 because 2 more points had the same distances. The function could not distinguish between them given that their distances were equal to one another from the original data point, so it provided all 7 nearest neighbors instead of 5.

6

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.

Maintaining sensitive healthcare information in private databases to be used in algorithmic models necessitates additional procedural protocols to ensure people's personal data isn't stolen. If healthcare companies want to gather this much data about us, they need to have a plan A, plan B, C, etc. in place of what to do with the data if their systems are compromised. The only people who need to be privy to the healthcare data itself are the ones who understand the gravity of maintaining it and are working to use it in a way that benefits others. This includes healthcare researchers who are studying the data, healthcare company executives who give approval for this research, and the database administrators who maintain it for use. Anyone who might take advantage of the data for their own personal gain should not have access, namely the insurance companies. As stated, insurance firms could argue they need the data to better calibrate their actuarial risk, but will also manipulate the data to use in their claims and even deny care to future patients in need. The risk of denying care is so significant in this scenario that it negates the value of any other data needs. No patient should have their own data used against them, nor would the patients who contributed to the database want to know that providing their data has negatively impacted another individual. The principles of utilitarianism and consequentialism would support not providing insurance companies with sensitive healthcare information. Utilitarianism argues that the right decision is the one that benefits the most people, which in this case the populus is better protected by keeping their data private. Consequentialism believes the best choice is the one which has the most positive outcome, and seeing as insurance companies could use healthcare data to discriminate against individuals and potentially restrict care, this moral philosophy also does not support giving them the information.

The overall aim of creating healthcare algorithms is to help patients, which should also be the goal of maintaining their data. It should be stored in a way that maximizes their privacy and receives explicitly informed consent for the data collected. If a healthcare database is hacked, the company should work to recover the data as quickly as possible through whatever means possible, because the needs of individual patients are more important than those of the private firm. Therefore, data transfer would be allowed if the company managing the software is subsumed. However, it must be transferred to the next ideal overseer who will maintain the same level of attention and dedication to patient protection. Ideally this would not be a privately held company unless they were the only option, as the new company will more than likely use the data for corporate gain. A better temporary holder would be the government, because they are guaranteed to have public interests in mind. Another ideal stipulation would be to offer the patients some choice of who their data goes to if it's hacked. The company could send them a survey or allow them to vote online anonymously toward the final solution, giving them some say in the matter. John Rawl's idea of justice supports that where differences exist, resources need to be allocated to the most vulnerable. In this case, the most vulnerable group is clearly the patients whose data is being stored. The utmost priority needs to be protecting them and their data, therefore resources need to be allocated to make data breaches as unlikely as possible.