Homework 7

IZ Raad

4/17/2024

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.

I am formulating this under the assumption that a first result of heads implies that the second flip will determine if we give an incriminating answer (heads again) or a nonincriminating answer (tails). A frist flip of tails implies that we state the truth regardless.

$$\pi = \theta^2 + (1-\theta)\hat{P}$$

$$\hat{P} = \frac{\pi - \theta^2}{1-\theta}$$

2

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

$$\hat{P} = \frac{\pi - \theta^2}{1-\theta}$$

$$\hat{P} = \frac{\pi - \frac{1}{2}^2}{1-\frac{1}{2}}$$

$$\hat{P} = \frac{\pi - \frac{1}{4}}{\frac{1}{2}}$$

$$\hat{P} = 2\pi - \frac{1}{2}$$

3

Consider the additive feature attribution model: $g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$ where we are aiming to explain prediction f with model g around input x with simplified input x'. Moreover, M is the number of input features.

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

Give an expression for the explanation model g in the case where all attributes are meaningless, and interpret this expression. Secondly, give an expression for the relative contribution of feature i to the explanation model.

If all attributes are meaningless, it is simply the case that $g = \phi_0$. If the input features have no effect on the prediction of the model, then $\phi_i = 0$ for $i \in 1...M$. Therefore, the value of the series simplifies to zero. Therefore, regardless of model input, we have that g will always yield a prediction of ϕ_0 (i.e. the model is constant).

Based on my response to the first part of the question, we see that if the total contribution of all features is given by $\sum_{i=1}^{M} \phi_i x'_i$, then the relative contribution of each feature i is $\phi_i x'_i$ (proportionally, $\frac{\phi_i x'_i}{\sum_{i=1}^{M} \phi_i x'_i}$). That is, the input of each feature is weighted by ϕ_i.

4

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.

```r
chebychev <- function(x, y){
  return(max(abs(x-y)))
}

nearest_neighbors = function(x, obs, k, dist_func){
  dist = apply(x, 1, dist_func, obs)
  distances = sort(dist)[1:k]
  neighbor_list = which(dist %in% sort(dist)[1:k])
  return(list(neighbor_list, distances))
}

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

```
## [1] 6
```

5

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.

```r
library(class)
df <- data(iris)

knn_classifier = function(x,y){
  groups = table(x[,y])
  pred = groups[groups == max(groups)]
  return(pred)
}
```

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

#find indices of 5 nearest neighbors in iris using cols 1:4 ("nearest" using L-inf norm)
ind = nearest_neighbors(x[,1:4], obs[,1:4], 5, chebychev)[[1]]

#show columns 1:4 of 5 nearest neighbors as a matrix
as.matrix(x[ind,1:4])
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 71           5.9         3.2          4.8         1.8
## 84           6.0         2.7          5.1         1.6
## 102          5.8         2.7          5.1         1.9
## 127          6.2         2.8          4.8         1.8
## 128          6.1         3.0          4.9         1.8
## 139          6.0         3.0          4.8         1.8
## 143          5.8         2.7          5.1         1.9
```

```
#show columns 1:4 of our observation
obs[,1:4]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 150          5.9           3          5.1         1.8
```

```
#classify our obs using nearest neighbors... mode species of our nearest neighbors
knn_classifier(x[ind,], 'Species')
```

```
## virginica
##         5
```

```
#what is the actual observed species
obs[,'Species']
```

```
## [1] virginica
## Levels: setosa versicolor virginica
```

6

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?

The first matrix in our output displays the sepal length, sepal width, petal length, and petal width out the nearest neighbors of our observation, found using our k nearest neighbors function. This matrix contains seven neighbors rather than the specified five because our nearest neighbors function operates by returning data points whose distance to our observation is equal to one of the five lowest distances (in our case, distance is defined by the L-infinity norm). It is entirely possible that our iris data set had three Chebychev distances tied for fifth closest. Thus, all of the associated data points were returned, totaling to seven neighbors returned by our function. The second output is a dataframe showing the sepal length, sepal width, petal length, and petal width of the observation we are trying to classify. Next, we see that we classify our observation as virginica, with five neighbors belonging to this category. Indeed, we have correctly classified our observation, as the final output shows that its true species is virginica (one of three species: setosa, versicolor, and virginica).

7

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.

I am not entirely sure of the innerworkings of this algorithm, but if there is a way for the data to remain inside the model and not be stored externally, then this is how we should operate. Perhaps, once data is used to train the model, its raw form is immediately deleted, and the only evidence that remains of it is in the model's performance. This way, if the company is subsumed, only the full working model is transferred (as opposed to raw data). Under no circumstance should insurance companies gain access to sensitive health care data.

I am extremely wary of data acquisition in accordance with the harm principle. Actions of those operating DeepMind involving sensitive data should be limited because of the harm it may cause to individuals whose data is being collected. One form of harm is actions of insurance companies. If an insurance company decides an individual is more at risk based on their data, they may be less likely to fund medical care for this individual. After all, an insurance company is most concerned with its bottom line. Limiting access to care to those who need it most is indeed a form of harm. Beyond just insurance companies, a lack of security around personal health information may affect someone's personal life. If someone with malicious intent was able to access data like allergies or highly stigmatized diagnoses, this puts the individual whose data was collected in jeopardy.