

Hw 7

Musie Gebreegziabher

4/19/2024

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 .

$$\hat{P} = \frac{\pi + \theta - \pi \cdot \theta}{2\pi - \pi \cdot \theta}$$

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

$$\hat{P} = \frac{1}{3} + \frac{K}{3}$$

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.

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.

Student Answer

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.

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

```

#student input
#chebychev function
chebychev <- function(x, y) {
  max(abs(x - y))
}

#nearest_neighbors function
nearest_neighbors <- function(data, query_point, k, distance_function) {
  distances <- apply(data, 1, function(x) distance_function(x, query_point))

  nearest_indices <- order(distances)[1:k]

  data[nearest_indices, ]
}

x<- c(3,4,5)
y<-c(7,10,1)
data <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), ncol = 3)
chebychev(x,y)

## [1] 6

nearest_neighbors(data, y, 2, chebychev)

##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8

```

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(neighbors, class_column) {
  neighbor_labels <- neighbors[, class_column]

  mode_label <- names(sort(table(neighbor_labels), decreasing = TRUE))[1]

  mode_label
}

#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)[[1]]
as.matrix(x[ind,1:4])

```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## 6           5.4           3.9           1.7           0.4
## 5           5.0           3.6           1.4           0.2
## 6.1         5.4           3.9           1.7           0.4
## 5.1         5.0           3.6           1.4           0.2
## 6.2         5.4           3.9           1.7           0.4
```

```
obs[,1:4]
```

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

```
knn_classifier(x[ind,], 'Species')
```

```
## [1] "setosa"
```

```
obs[, 'Species']
```

```
## [1] virginica
```

```
## Levels: setosa versicolor virginica
```

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?

In this output we did get the correct classification.

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.

Any access or acquisition of any data of any individual should be at the consent of the individual and prioritize their right to privacy. Full autonomy should be given to patients as to where or to whome their data is being shared with full precautions and awareness of what's being shared. Any use outside of patient consent and awareness may hard trust and make necessities like health care something harder than it already is.