Hw 7

Yuewei Wang

1/5/2024

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 $\widehat{P}$ [1] was given by $\widehat{P} = 2\pi - \frac{1}{2}$ 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 $\widehat{P}$ for the proportion of incriminating observations. This expression should be in terms of θ and π.

$P(\text{ "yes" }) = P(\text{ "yes" } \mid \text{ heads }) \times P(\text{ heads }) + P(\text{ "yes" } \mid \text{ tails }) \times P(\text{ tails })$
$P(\text{ "yes"})) = \pi \times \theta + 1 \times (1 - \theta) = \pi\theta + 1 - \theta$ **solving for**π $\pi = \frac{\hat{p} - (1-\theta)}{\theta}$ So

$$\widehat{P} = \frac{\hat{p} - (1 - \theta)}{\theta}$$

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

Plugging $\theta = \frac{1}{2}$

$$\widehat{P} = \frac{\hat{p} - \left(1 - \frac{1}{2}\right)}{\frac{1}{2}}$$

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

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

$$\hat{P} = 2\hat{p} - 1$$

# 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.

When All Attributes Are Meaningless:

$$g(x') = \phi_0$$

This represents the baseline output of the model where none of the input features impact the prediction, indicating a constant output regardless of input changes.

The contribution of each feature i to the explanation model is:

$$\phi_i x_i'$$

This expression shows how the value of feature i, scaled by its attribution coefficient ϕ_i, affects the prediction relative to the baseline ϕ_0.

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
#nearest_neighbors function

# Define the Chebychev distance function
chebychev <- function(x, y) {
  max(abs(x - y))
}

# Define the nearest neighbors function
nearest_neighbors <- function(data, query_point, k, distance_func) {
```

```r
  distances <- apply(data, 1, function(row) distance_func(row, query_point))
  nearest_indices <- order(distances)[1:k]
  list(nearest_indices = nearest_indices, nearest_points = data[nearest_indices, , drop = FALSE])
}


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

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)
#student input
knn_classifier <- function(neighbors, class_column) {
  class_labels <- neighbors[[class_column]]
  mode <- function(x) {
    count <- unique(x)
    count[which.max(tabulate(match(x, count)))]
  }

  mode(class_labels)
}

#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])
obs[,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?

The output matrix presents the sepal length, sepal width, petal length, and petal width of the nearest neighbors to our target observation. Interestingly, this matrix includes seven neighbors instead of the specified five. This deviation occurs because our function is designed to return all data points that share any of the five smallest Chebychev distances to the observation. Given that there were ties at the fifth closest distance, all corresponding points were included, resulting in seven neighbors being returned. Following this, the output includes a dataframe detailing the features of the observation being classified. The classification process doesn't successfully identifies this observation as belonging to the 'virginica' species, with the majority of its neighbors also labeled 'virginica'. This classification is not validated by the final output, not confirming the observation's actual species as 'virginica'.

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 Sensitive health care data, especially in situations of company transition, or where insurance companies may access it, necessitates stringent ethical guidelines. It is vital to obtain informed consent; patients must be fully aware and explicitly told how their health information could be employed in future including possibly being transferred to new corporate entities. This approach ensures patients' autonomy and enables people to have a hold over their own details.

There are significant ethical issues around the possible misuse of such data by insurance firms including using it to limit care or unfairly adjust premiums. These practices not only violate the principle of non-maleficence which lies at the core of medical ethics but also have implications for society as a whole including discrimination outcomes. What happened with COMPAS algorithm and its racial biases is an acid example of how data can reinforce social divisions.

I further assert that giving out personal information for use in medical research should be on a voluntary basis and ought to be made after one has gathered enough facts about it. Making these contributions should not feel like an obligation but rather as something done under full knowledge of what will happen with the data involved. The principle of justice demands equal treatment for all people thereby respecting their decisions.

In short, strict ethics must guide healthcare institutions dealing with sensitive healthcare information while robust data protection regulations should be put in place. This helps protect individual privacy as well as improving public trust that is very paramount for continuity or sustainability of health systems and research endeavors.