Hw 7

Swagat Adhikary

12/01/2024

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

Generalizing this process for a biased coin with probability of landing heads equal to θ means that the proportion of people answering affirmative to the incriminating question would be a sum of those implicated who flipped heads and responded truthfully ($\theta\hat{P}$) and those who flipped tails first ($1-\theta$) and then heads (θ). This gives us:

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

Solving for $\hat{P}$ gives the generalized formula:

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

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

Substituting 1/2 for θ gives:

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

Simplified we get the result established in class:

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

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.

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

```r
#student input
#chebychev function
chebychev <- function(vec1, vec2) {
  return(max(abs(vec1 - vec2)))
}
#nearest_neighbors function
nearest_neighbors <- function(data, observation, k, distance_function) {
  # Calculate distances between the observation and each point in the dataset
  distances <- apply(data, 1, function(row) distance_function(row, observation))

  # Sort distances to find the k-th smallest distance
  sorted_distances <- sort(distances)
  kth_distance <- sorted_distances[k]

  # Include all indices with distances <= k-th smallest distance
  nearest_indices <- which(distances <= kth_distance)

  # Return the indices and corresponding distances
  return(list(indices = nearest_indices, distances = distances[nearest_indices]))
}

x<- c(3,4,5)
y<-c(7,10,1)
#cheby(x,y) <- provided function name != name used for testing
chebychev(x,y)
```

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

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_data, class_column) {
  # Extract class labels of the nearest neighbors
  class_labels <- neighbors_data[[class_column]]

  # Return the mode of the class labels
  return(names(which.max(table(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])
```

```
##     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
```

```
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] "virginica"
```

```
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?

The classification of the observation as *virginica* is correct, matching the true class of the observation. This indicates that the K-Nearest Neighbors (KNN) algorithm performed well, as the majority class among the nearest neighbors (including any ties) correctly predicted the class label.

The output includes 7 observations instead of the specified K=5 because there were ties, meaning that the 5 smallest distances included more than 5 neighbors. The `nearest_neighbors` function includes all data points closer than or equidistant to the 5th furthest neighbor to ensure fairness and consistency. This behavior prevents the arbitrary exclusion of tied points, which could otherwise affect the 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.

Algorithmic assistance in healthcare, like Google DeepMind's clinical alert app for acute kidney injury, raises crucial ethical concerns regarding autonomy, harm, and informed consent. Sensitive health data, while essential for such innovations, must be handled carefully to avoid harm and protect patient autonomy.

Patients have the *personal autonomy* to control how their data is used. The *harm principle*, as articulated by J.S. Mill, states that autonomy may only be limited to prevent harm to others. In the case of DeepMind, the unconsented transfer of patient data could be considered a violation of autonomy because patients were not given the opportunity to decide how their information would be used. Even though the project aimed to benefit patients, its potential to harm, such as misuse of data for non-clinical purposes, justifies invoking the harm principle to restrict such transfers.

Google could argue that the data transfer was an act of *paternalism*, intended for the long-term benefit of patients by enabling faster diagnoses and treatment. However, as discussed in class, paternalistic actions must balance potential benefits against immediate harm and risks. Without explicit consent, patients lose sovereignty over their data, creating a slippery slope where future breaches could lead to adverse consequences, such as misuse by third parties (e.g., insurance companies denying care).

The unconsented transfer of data violates the harm principle by undermining patient autonomy and exposing them to potential harm. While the intended benefits align with paternalistic justifications, the lack of informed consent makes the transfer ethically questionable. The situation underscores the importance of protecting autonomy through clear consent mechanisms while weighing harms and benefits using consequentialist principles.

I have described our responsibility to proper interpretation as an *obligation* or *duty.* How might a Kantian Deontologist defend such a claim?

A Kantian deontologist would argue that the responsibility to properly interpret and substantiate claims is a moral *duty*, rooted in the formulations of the categorical imperative. Failing to uphold this responsibility violates both key formulations of Kant's ethical framework: the *universalization principle* and the *humanity principle.*

The universalization principle states that one must act only according to maxims that can be consistently willed as universal laws. If advancing claims without proper interpretation and substantiation were universalized, the practice of making claims would become meaningless. People would lose trust in all claims, rendering the act of communication self-defeating. Thus, the maxim of advancing unsupported claims cannot be universalized, making it a violation of this formulation of the categorical imperative.

The humanity principle requires treating all moral agents as ends in themselves, never as mere means to an end. Presenting claims without proper interpretation exploits researchers, readers, or collaborators by misleading them for personal or professional gain (e.g., advancing one's work or reputation). This instrumentalization disrespects their rational capacities and moral agency, violating the humanity principle.

A Kantian deontologist would defend the obligation to proper interpretation as a strict moral duty. Upholding this duty respects the rationality and autonomy of others while preserving the integrity of intellectual and ethical discourse. Failing to do so not only erodes trust but also undermines the very foundations of moral and intellectual interaction.