Hw 7

Logan Schmitt

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.

Logan Answer For this scenario, let's imagine a coin with the probability of heads being θ and the probability of tails being 1 - θ.

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

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

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

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

Step 1: Expand $\frac{\pi - (1-\theta)\theta}{\theta}$:

$= \frac{\pi - \theta + \theta^2}{\theta}$

Step 2: Substitute $\theta = \frac{1}{2}$ into the equation:

$= \frac{\pi - \frac{1}{2} + \frac{1}{2}^2}{\frac{1}{2}}$

Step 3: Plug in $\theta = \frac{1}{2}$ and simplify:

$= \frac{\pi - \frac{1}{2} + \frac{1}{4}}{\frac{1}{2}}$

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

$= 2 * (\pi - \frac{1}{4})$

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

Therefore, $\hat{P}$ reduces to $2\pi - \frac{1}{2}$, our result in class.

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

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.

Logan Answer Firstly (giving an expression for the explanation model g in the case where all attributes are meaningless): ϕ_0 ϕ_0 serves as the baseline value or the intercept. This represents the prediction of the model when no features are present. Given if all features are meaningless, we would be left with just this value.

Secondly (giving an expression for the relative contribution of feature i to the explanation model): $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_s(x_s)$ This difference gives the marginal impact of feature i, as we are computing the difference in model output where in one instance we are using feature i and the other we are not.

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.

```
# Logan input:

# chebychev function
chebychev = function(x, y){
    max(abs(x - y))
}

# nearest_neighbors function
nearest_neighbors = function(x, obs, k, dist_func){
  dist = apply(x, 1, dist_func, obs) #apply along the rows
  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
```

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)
```

```r
# Logan input

# knn_classifier function
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 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
```

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

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

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

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

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

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

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

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 71          5.9         3.2          4.8         1.8 versicolor
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 84            6         2.7          5.1         1.6 versicolor
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 102          5.8         2.7          5.1         1.9 virginica
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 127          6.2         2.8          4.8         1.8 virginica
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 128          6.1           3          4.9         1.8 virginica
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 139            6           3          4.8         1.8 virginica
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 143          5.8         2.7          5.1         1.9 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 data frame?

Logan Answer The k-Nearest Neighbors classifier assigned the class label virginica to the unknown observation, which is the correct classification. Based on the sepal length, sepal width, petal length, and petal width of the plant species used as our known observations in the kNN classifier, the unknown plant species's characteristics are most similar to those of the virginica species. This means that the differences in these factors are the smallest between our unknown observation and our known virginica observations. Therefore, this also implies that the majority of the nearest neighbors also shared the virginica classification.

Although I specified a $K = 5$, there are 7 observations included in the output. This is because two of the observations, 102 and 143, are identical in all the features. Therefore, the algorithm, rather than removing these two observations, added two additional ones instead. Alternatively, the algorithm could have removed one of the repeat observations or added an additional observation, however, this would result in an even-numbered K. This has the potential to be problematic, as even-numbered K's can result in ties between class label classifications.

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.

Logan Answer I believe data transfer is allowable if the company managing the software is subsumed or at least if strict guidelines and security measures are established. From a utilitarian standpoint, access to this sensitive healthcare data can be leveraged to benefit the maximum number of patients possible. In regards to current patients, it could save much needed time in regards to the process of diagnosing and treating them, as there would be a much higher accuracy of correctly treating an ailment on the first attempt. This allows for more time to be dedicated towards the care of patients with more complex issues, effectively enabling the most amount of patients to be seen. Additionally, the information can benefit future patients with specific healthcare issues based on historical data of similar patients with the same issues. Overall, retaining sensitive health care data carries the ability to minimize long-term pain among patients. However, as previously mentioned, guidelines must be set in place to minimize access to the information as well as the amount of sensitive data retained.

In order to minimize those who have access to this sensitive information, I believe the access should only belong to two main groups: those assisting in the collection and management of the data (i.e., DeepMind) and the healthcare professionals directly involved in the care and treatment of patients. While the acquisition and retention of this data is useful, it is still important to establish a sense of trust among patients that their data will remain confidential and private. As a result, it's necessary to limit this to only the groups crucial to maintaining and leveraging this information. With this limited access, there is less risk in regards to breaches, unauthorized alterations, or misuse of the data. Additionally only information proved to be necessary in the treatment of future patients should be kept. Given the scenario where an outside source does obtain access to the data, there should not be any identifying information that is able to be linked back to a specific patient. Lastly, transparency is critical in establishing trust. Therefore, patients should be fully-informed in regards to which of their personal information is being retained in the healthcare sector, who has access to it, and how it will be used.

Based on the previous discussion, I believe that this sensitive data should not be made available to insurance companies. Granting insurers access to this information could lead them to deny coverage or increase premiums towards clients based on similarities in health profiles derived from historical data. However, these similarities between profiles do not guarantee similar health outcomes; as a result, this has the strong potential to lead to unfair treatment among certain groups. In regards to healthcare, many individuals have predispositions to certain medical conditions beyond their control. Denying coverage or increasing premiums effectively punishes this group due to luck-based differences, which is not fair or equitable. In an industry that affects the livelihood of others such as the healthcare sector, equality should be prioritized to ensure that everyone has equal access to needed care, regardless of the conditions in which they may come from.