Monte Carlo Simulation - Bayesian

Nikhil Kamath

2024-07-09

Bayesian Probability - Monte Carlo Simulation on Automotive Data

In this project, we will be using R code in order to do an exploratory data analysis of automotive data using the foundations of Bayesian Probability to produce accurate and concise results.

Questions and Objectives

1.)Use Monte Carlo approximation to estimate the marginal probability of a compact car (manufacture == 'honda').

2.) Use Gibbs sampling of Binomial-Beta conjugate prior-posterior to estimate the marginal probability of a 'honda' car.

3.)Use Naive Bayes to estimate the conditional probability of a 'honda' car given the MPGs of city (cty) and highway (hwy).

4.)Besides the city and highway MPGs, what else features are useful to predict a car manufacture?

#Loading in Libraries

#Monte Carlo Simulation - Honda

Next, we are going to simulate a monte carlo simulation using R code foor the mpg data

```
# Monte Carlo approximation to estimate the marginal probability of a Honda car
set.seed(123)  # For reproducibility
n_samples <- 10000
samples <- sample(mpg$manufacturer, n_samples, replace = TRUE)
honda_count <- sum(samples == 'honda')
honda_prob <- honda_count / n_samples

print(honda_prob)
```

```
## [1] 0.0336
```

Based on this simulation, the probability above shows the probability that a car is drawn is a Honda is 0.0336.

#Gibbs Sampling for Binomial-Beta Conjugate Prior-Posterior

Next we will be conducting a Gibbs sampling with the given code:

```
# Function for Gibbs sampling of Binomial-Beta
gibbs_sampler <- function(n_iter, a, b, data) {
  # Initialize storage for samples
  samples <- numeric(n_iter)

  # Initial value for theta
  theta <- rbeta(1, a, b)

  for (i in 1:n_iter) {
    # Sample from Beta posterior
    theta <- rbeta(1, a + sum(data), b + length(data) - sum(data))
    samples[i] <- theta
  }

  return(samples)
}

# Filter data for Honda cars
honda_data <- mpg %>% filter(manufacturer == 'honda')
n_honda <- nrow(honda_data)

# Assume prior parameters a and b
a <- 1
b <- 1

# Run Gibbs sampler
n_iter <- 10000
samples <- gibbs_sampler(n_iter, a, b, rep(1, n_honda))

# Posterior mean estimate
posterior_mean <- mean(samples)
print(posterior_mean)
```

```
## [1] 0.9071739
```

The result of the gibbs value is around 0.91, which is the best estimate of the joint distribution of variables.

Naive Bayes to Estimate Conditional Probability of a ‘Honda’ Car Given MPGs of City (cty) and Highway (hwy)

```
# Load necessary libraries
install.packages("e1071")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(e1071)

# Prepare data
mpg$manufacturer <- as.factor(mpg$manufacturer)

# Fit Naive Bayes model
model <- naiveBayes(manufacturer ~ cty + hwy, data = mpg)

# Predict the probability of a car being a Honda
pred <- predict(model, mpg, type = "raw")

# Print the first few probabilities for Honda
print(head(pred[, "honda"]))
```

```
## [1] 3.744117e-04 3.708975e-02 3.713792e-02 7.210436e-02 5.428574e-07
## [6] 1.518399e-05
```

The values provided appear to be the conditional probabilities estimated by the Naive Bayes model for a 'Honda' car given different combinations of city (cty) and highway (hwy) miles per gallon (MPG) features. Here's how we can interpret these values:

First value: 3.744117e-04: This represents a very small probability, indicating that given the specific combination of cty and hwy MPG values in this instance, the likelihood of the car being a Honda is quite low.

Second value: 3.708975e-02: This represents a probability of approximately 0.037, suggesting a higher likelihood compared to the first value, but still relatively low.

Third value: 3.713792e-02: This is similar to the second value, indicating a slightly higher probability for a different combination of cty and hwy MPG values.

Fourth value: 7.210436e-02: This value is approximately 0.072, indicating a higher probability compared to the previous ones, suggesting a greater likelihood of the car being a Honda given the particular cty and hwy MPG values.

Fifth value: 5.428574e-07: This represents an extremely small probability, almost negligible, indicating that for this combination of cty and hwy MPG values, the likelihood of the car being a Honda is extremely low.

Sixth value: 1.518399e-05: This is also a very small probability, indicating a very low likelihood of the car being a Honda for the given combination of cty and hwy MPG values.

#Feature Engineering - Prediction

There are two ways to go about the feature conditional probability, the first is a machine learning algorithm known as a random forest classifier, but seeing as we are statisticians, we must use an ANOVA table in order to deterimine the most prominent and predicitive features

```
# Load necessary libraries
install.packages("randomForest")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# Fit Random Forest model
rf_model <- randomForest(manufacturer ~ ., data = mpg, importance = TRUE)

# Get feature importance
importance <- importance(rf_model)

# Print feature importance
print(importance)
```

```
##              audi chevrolet     dodge       ford      honda    hyundai
## model 41.628084040 24.477049 35.354177 35.0905923 22.8513523 41.1931477
## displ 13.051823487 23.126848 24.977564 20.2145612 20.5537774 16.9119869
## year  -0.002935136 -2.611007  4.847719 -0.6786996 -0.4686718 -0.2185937
## cyl    2.174046127  9.419773 11.626494  7.6719051  9.2299740  1.1325775
## trans -0.218906040  8.601563  6.725494  0.2278058 -0.7822242 -4.1984314
## drv    4.945953623 19.173579 16.804630 22.7293338 11.7627698 18.8205682
## cty    4.583397922  4.561387 19.089092  9.8850872 15.3015429  7.8245314
## hwy    9.642393569  4.705357 21.040037 11.5538497 14.5300833  8.7348467
## fl    19.130976105  6.285627  5.468553  8.2047632 -1.9585243  9.8808384
## class 16.938334030 16.739152 22.464341 14.9772800 18.3903970 11.0610485
##             jeep land rover    lincoln   mercury     nissan   pontiac
## model 16.4841573  18.656177 11.3235537 10.822481 18.1810812 13.789080
## displ  0.7677092  12.663393  9.7751319  4.046017 10.0300418  7.364881
## year  -0.2175213   3.494028 -1.9877061 -3.043071  0.7599166 -2.896723
## cyl    5.8430528   9.404748  3.6392587 -3.215355  6.1304004  2.531817
## trans  4.8210270   5.643531  0.8826038  1.142456 -6.1658559  3.671678
## drv    5.7046806   7.320375 11.3476928  3.845111  9.0664653  8.561159
## cty   -0.4936311  15.998072  6.5471840  4.463838  7.0498352  3.919294
## hwy    3.6370871  11.422490  4.1212765  4.255604  7.6664513  7.375775
## fl    -1.4787902   4.366943 -4.2603974  3.856657  4.3006794 -3.891100
## class 12.8177636  15.144899  7.3879021  9.146712 13.2750111  9.841350
##           subaru     toyota volkswagen MeanDecreaseAccuracy MeanDecreaseGini
## model 21.8555799 43.2076798  35.398625            68.837553        72.265537
## displ 20.9531681 18.1495376  25.380781            39.323145        32.330986
## year   2.9641204 -0.2709814   3.827677             1.884913         2.672382
## cyl   12.8040586  5.4363921  10.911992            16.251996         6.209181
## trans -0.4448939 -4.0046429   5.809556             7.444258         8.528418
## drv   29.2261913 19.6773221  24.079610            40.985484        17.745981
## cty   13.4583672 15.7021873  11.948571            25.393256        17.372602
## hwy   14.9558230 19.5278000  13.284218            25.923298        18.740097
## fl     0.3425922 17.2527912  10.391665            22.777941         8.906733
## class  4.4579065 24.4846543  13.326934            33.961343        21.251739
```

ANOVA variance section

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Load the mpg dataset
data(mpg)

# Function to perform ANOVA
anova_test <- function(feature) {
  model <- aov(mpg[[feature]] ~ mpg$manufacturer)
  anova_result <- summary(model)
  return(anova_result[[1]]$`Pr(>F)`[1])
}

# Select numerical features
numerical_features <- mpg %>% select_if(is.numeric) %>% names()

# Apply ANOVA to each numerical feature
anova_p_values <- sapply(numerical_features, anova_test)

# Print ANOVA p-values
print(anova_p_values)
```

```
##        displ         year          cyl          cty          hwy
## 5.089788e-44 8.844286e-01 8.202262e-35 2.031554e-31 9.220796e-30
```

Based on the ANOVA values, we can see that the features that predict mpg are displacement, the year, cylinder, city, and highways of the given honda at hand. This also corresponds to the random forest classifer that was run in the Honda section. Cross refrencing these values, we can see that these are the biggest indicators of MPG.