discussion3

Coco_Luo

2023-02-22

```
Data <- read.csv("~/Desktop/528/reading3/Data.csv")
```

There are 5178 missing data and so we have about 20% missing data

```
sum(is.na(Data))/25000
```

```
IsMissingMatrix=is.na(Data)
```

```
install.packages("missForest")
library(missForest)
```

First, try to use the build in package missForest to impute my data. missForest is an algorithm used for imputing missing values in a dataset. It belongs to the class of non-parametric methods, meaning that it does not make assumptions about the distribution of the data. The missForest algorithm is based on random forests, which are an ensemble learning method for classification, regression and other tasks.

In missForest, the random forest algorithm is used to predict the missing values in a dataset based on the values of the other variables. The algorithm first splits the data into two sets: one with missing values and one without. It then builds a random forest model on the non-missing data and uses it to predict the missing values in the other set. This process is repeated multiple times to get more accurate predictions.

One advantage of missForest is that it can handle missing values in both continuous and categorical variables, as well as variables with complex relationships. It also allows for multiple imputations, meaning that it can generate several complete datasets with different imputations, which can be useful for further analysis.

However, missForest also has some limitations, such as being computationally intensive and potentially generating imputations that are biased or inaccurate in some cases. It is important to carefully evaluate the results and consider the characteristics of the data and the research question when using missForest or any other imputation method.

I got an NRMSE 0.318 which is kind of large.

```
imp <- missForest(Data)
missForest_out = as.matrix(imp$ximp)
imp$OOBerror

ImputedValues=as.vector(missForest_out)[as.vector(IsMissingMatrix)]
output=data.frame(Id=c(1:length(ImputedValues)), Predicted=ImputedValues)
#write.csv(output,'missForest_output.csv',row.names = FALSE)
```

I tried a simple mean imputation where I wrotea function to do the imputation. Mean imputation involves replacing missing values with the mean of the observed values in that variable.

```
# use mean to impute
impute.matrix <- function (matrix) {
    missing <- which(is.na(matrix) | !is.finite(matrix), arr.ind=TRUE)
    if (length(missing)!=0){
```

```
        for (j in 1:nrow(missing)){
                mean <- mean(matrix[missing[j,1],][is.finite(matrix[missing[j,1],])], na.rm=TRUE)
                matrix[missing[j,1],missing[j,2]] <- mean
            }
    }
    matrix
}

Mean_out = impute.matrix(as.matrix(Data))

ImputedValues=as.vector(Mean_out)[as.vector(IsMissingMatrix)]
output=data.frame(Id=c(1:length(ImputedValues)), Predicted=ImputedValues)
#write.csv(output,'mean_output.csv',row.names = FALSE)
```

Regression imputation involves using regression models to estimate missing values in a variable based on other variables in the dataset. To perform regression imputation on this dataset, we can use the mice package, which is a popular package for multiple imputation of missing data.

```
library(mice)
#imp <- mice(Data, method = "norm.predict")
Regression_out <- complete(imp)

#mimp <- mice(Data, m = 5) # create 5 imputed datasets
Multiple_out1 <- complete(mimp, 1)
Multiple_out2 <- complete(mimp, 2)
Multiple_out3 <- complete(mimp, 3)
Multiple_out4 <- complete(mimp, 4)
Multiple_out5 <- complete(mimp, 5)

r <- as.numeric(unlist(Regression_out))

ImputedValues=as.vector(r)[as.vector(IsMissingMatrix)]
output=data.frame(Id=c(1:length(ImputedValues)), Predicted=ImputedValues)
write.csv(output,'Regression_output.csv',row.names = FALSE)
```

Multiple imputation involves creating multiple imputed datasets and then combining them to obtain a final imputed dataset. This method takes into account the uncertainty in the imputed values and can provide more accurate estimates. To perform multiple imputation on my datadet, I still use the mice package.

```
# Impute missing data using mice
#about 10% average missing data, so maxit= 20
tempData <- mice(Data,m=5,maxit=20,meth='pmm',seed=500)
summary(tempData)

# Possible imputation models provided by mice() are
methods(mice)

# imputation method I use
tempData$meth

# Get completed datasets (observed and imputed)
completedData <- complete(tempData,1)
completedData <- as.numeric(unlist(completedData))

ImputedValues=as.vector(completedData)[as.vector(IsMissingMatrix)]
```

```
output=data.frame(Id=c(1:length(ImputedValues)), Predicted=ImputedValues)
write.csv(output,'multiple_mice_output.csv',row.names = FALSE)
```

K-nearest neighbor imputation involves using the values of the k-nearest neighbors to estimate missing values in a variable. To perform K-nearest neighbor (KNN) imputation, I used the multiUS package. But it seems that there are some limitations on using this method when we do not have sufficient complete cases for computing neighbors.

```
library(multiUS)
knn_out = KNNimp(data = Data)
```

Expectation-maximization (EM) imputation iteratively estimates the missing values in a variable based on the observed values and the estimated values of the other variables in the dataset. I used the missMethods package. However, we need to be aware that this method does not always converge.

```
library(missMethods)
EM_out <- impute_EM(Data)
```

```
m <- as.numeric(unlist(EM_out))
```

```
ImputedValues=as.vector(m)[as.vector(IsMissingMatrix)]
output=data.frame(Id=c(1:length(ImputedValues)), Predicted=ImputedValues)
write.csv(output,'output.csv',row.names = FALSE)
```

Usually in practice, the choice of imputation method depends on the specific characteristics of the dataset and the research question being addressed. It is important to evaluate the performance of different imputation methods using appropriate criteria such as the mean squared error, correlation coefficients, or the ability to recover the true underlying structure of the data.