R Homework 2

Vinh Nguyen

***Abstract*—The purpose of this homework is to understand several imputation methods, how they work in some specific cases, Using the iris dataset, randomly create missing values to occupy x% of the data. Use R to apply 3 different methods of imputation and compare their performance when x=2, 5, 10 , 15, 20, 25. Use Root mean square error (RMSE) between imputed and true values and Supervised classification error (Use k-NN classifier). Discussions on the results are also provided**

***Keywords*—R, imputation methods, missing values**

I. Imputation methods

When the number of observation is big or the dataset is too large, one of the easiest approach is to drop the observations containing missing values. A good indicator is that the number of observations having missing values should be less than 5 percent of the total records. But there is a case that, omitted data may contain useful information so it is wisely to *impute* the missing values instead of excluding them from the data.

A. Mean Imputation Method

Perhaps, this is the easiest way to impute missing values when one wants to maintain the size of the dataset. This method substitutes averaging value of all available cases for the missing data. This method is simple but it reduces the variability of the data. It is often used in the case when adding data does not change so much to the analysis such as in questionnaire manuals.

B. k nearest neighbor imputation

This method is based on donor observation, that is, it defines each individual of a given variable a set of its k–nearest neighbours and then the missing value will be replaced by averaging values of its neighbours. I found this approach is very interesting compared to the mean imputation method because it takes the mean from its neighbour rather than the whole population.

C. Regression imputation

In this method, the imputed value is predicted from a linear regression equation. For this method, other variable with completed data is used to predict the values of the missing observations.

Vinh Nguyen. Phd Student at Computer Science Department, Texas Tech University. email: vinh.nguyen@ttu.edu

D. Predictive Mean Matching (PMM)

PMM has been introduced for a long time [2] and is considered to be very effective method since it produces imputed values that are very close to the real values. Suppose our dataset contains variable or feature Y that has some missing values, and a set of variable Xs that have complete data. These variables Xs will be used to impute missing values in Y.

- **Step 1:** When there is no missing data, the algorithm performs a linear regression of Y on Xs, estimating a set of coefficients c.
- **Step 2:**: Randomly draw from the posterior predictive distribution of c to create a new set of coefficients c^*
- **Step 3:** From the coefficients c^*, the algorithm generates the predicted values for Y for all cases, meaning that imputed values are generated for both missing values and presented values.
- **Step 4:** For each case with missing value Y, PMM identifies a set of other cases with observed Y in which predicted values are close to the missing data's predicted value.
- **Step 5:** Randomly draw one from these close cases and assigns its observed value for the missing value.
- **Step 6:** Repeats Step 2 to Step 6 until data is completed.

Unlike many other linear regression methods that substitute missing values for imputed values generated directly from the model, PMM provides a list of matching cases that have predicted value close to the missing data's predicted value.

II. Dataset

In this experiment, we use a very popular dataset for machine learning, that is, the IRIS dataset introduced by Fisher [1]. This dataset contains 150 observations for three species of flowers (setosa, versicolor, and virginica) with 4 variables (Sepal.Length, Sepal.Width, Petal.Length, Petal.Width)

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

Fig. 1. IRIS DATASET SUMMARY

III. Implementation with random missing value

In this section, we are going to randomly create some missing values to occupy x percent (2, 5, 10, 15, 23, 25) of the available dataset and perform three different methods to impute missing data as shown in Table I, Table II and Table III.Missing values are randomly created by using $sample$ methods with seeding. For the consistency and comparison between the three imputation methods, we will work on the first column only, which is $Sepal.Length$. The second column on each table shows the RMSE between the imputed values and the true values. Ori. Accuracy shows the accuracy of the kNN classifier of the true values whereas Imputed Accuracy shows the accuracy of the kNN classifier of the imputed values. We split the data into training and testing with ratio 0.7 and 0.3 respectively. It can be seen from the three tables that, as the number of missing values increase, RMSE also increases linearly. kNN Imputation method seems to performs best of the three, and Mean Imputation method gives the worst result. Classification of the flowers of the original fluctuates in three result but consistent, this can be explained by the initial of k, leading to different results. But overall we expect that these results should be the same since we don't make any change in the data. Imputed data accuracy, on the other hand shows a trend of small decreasing and consistent with Table 1 and Table 3, we expect these values since RMSE increases.

TABLE I. Performance Comparison with Mean Imputation method

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.1167	0.9333	0.9556
5	0.1645	0.9333	0.9556
10	0.2607	0.9556	0.9556
15	0.3158	0.9778	0.9556
20	0.3370	0.9333	0.9111
25	0.4126	0.9556	0.9111

TABLE II. Performance Comparison with kNN Imputation method

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.0527	0.9333	0.9333
5	0.0759	0.9333	0.9333
10	0.0890	0.9556	0.9333
15	0.1103	0.9778	0.9333
20	0.1230	0.9333	0.9333
25	0.1666	0.9556	0.9333

TABLE III. Performance Comparison with Linear Regression Imputation method

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.1088	0.9333	0.9556
5	0.1559	0.9333	0.9556
10	0.2529	0.9556	0.9556
15	0.3082	0.9778	0.9556
20	0.3307	0.9333	0.9111
25	0.4067	0.9556	0.9111

IV. Implementation with non-random missing value

In this section, we conduct an experiment to create a missing values not randomly. This time, we want to perform on the last column, the $Petal.Length$. As analyzed in classroom, this variable plays the most important role in classification so it is interesting to know if this feature contains missing values and to know how well the kNN classifier can work on this problem.

TABLE IV. Performance Comparison with Mean Imputation method Non Random

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.3451	0.9777	0.9333
5	0.5252	0.9777	0.9333
10	0.8227	0.9777	0.9333
15	1.0373	0.9777	0.9333
20	1.2797	0.9777	0.9333
25	1.5142	0.9777	0.9333

TABLE V. Performance Comparison with kNN Imputation method Non Random

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.0154	0.9777	0.9333
5	0.0258	0.9777	0.9333
10	0.0458	0.9777	0.9333
15	0.0556	0.9777	0.9333
20	0.0866	0.9777	0.9333
25	0.0915	0.9777	0.9333

TABLE VI. Performance Comparison with Linear Regression Imputation method No Random

Missing percent	RMSE	Ori. Accuracy	Imputed Accuracy
2	0.0252	0.9778	0.9333
5	0.0496	0.9778	0.9333
10	0.0670	0.9778	0.9333
15	0.1309	0.9778	0.9333
20	0.1768	0.9778	0.9333
25	0.2337	0.9778	0.9333

Results from the three tables did not give too much interesting prediction since all imputed accuracy in all table give the same results. Only RMSE increases linearly with the percentage of missing value. Complete R codes are found at https://github.com/Alex-Nguyen/CS5331R

References

[1] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.

[2] R. J. Little. Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, 6(3):287–296, 1988.