# Lecture 11: Imputation

Alan Lee

Department of Statistics
STATS 784 Lecture 11

August 28, 2017

# Outline

# Today's agenda

In this lecture we present a further discussion of imputation, filling in "missing values" in a data set.

# MCAR, MAR and NMAR

**MCAR** : data is completely missing at random: "missingness" is independent of the data values. In this situation using only the complete data (cases having no missing values) will give an unbiased result, but with a smaller sample size than if no data was missing. Can result in ignoring a large proportion of the data.

**MAR** : data is missing at random: "missingness" depends only on the non-missing data (and thus in principle the missing values can be predicted from them).

**NMAR** : not missing at random - "missingness" depends on the missing and non-missing data. Not much can be done in this situation.

# Basic idea of multiple imputation

- For each variable in turn, impute a missing value by drawing from the conditional distribution of the variable, given the rest of the data.
- This amounts to predicting the missing value, and adding small amount of noise to the prediction.
- Use the imputed data to construct a predictor.
- Repeat this several times to obtain multiple predictors.
- Average or “majority vote” to get a final predictor.

# R packages

- The packages `mice` and `mi` do multiple imputation using a variety of prediction methods.
- The package `missForest` is based on a different idea, see later.
- See the tutorial at https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/ for more information on other packages.

# Summarizing patterns of missing data

This is best done visually.

- Use barcharts of missing value proportions.
- Use the `image` function to show where the missing values are in the data set.
- Use the `md.pattern` in the `mice` package for a text summary of the missing value patterns.

# Example: Fisher iris data

50 samples of iris from each of 3 species: *Setosa*, *Versicolor* and *Virginica*

Variables measured: sepal length and width, petal length and width



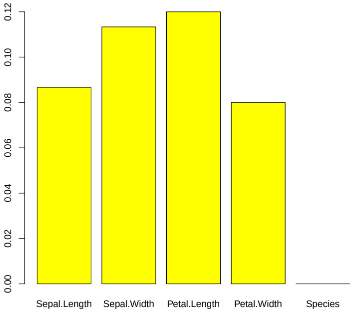
# Example: the (missing) iris data

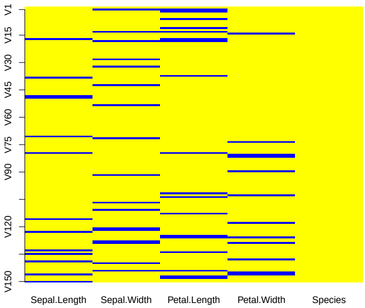
For the setosa data, we made data values go missing (replaced with NA's) in 10% of the size data. Similarly, for versicolor and virginica, 5% and 15% were set to NA. This we have a dataset that is MAR, none of the species values are missing.

# Example: missingness patterns

```
> library(mice)
> md.pattern(iris.miss)
    Species Sepal.Length Sepal.Width Petal.Width Petal.Length
102       1            1           1           1            1  0
  7       1            0           1           1            1  1
  7       1            1           0           1            1  1
 21       1            1           1           1            0  1
  8       1            1           1           0            1  1
  2       1            1           0           1            0  2
  2       1            0           1           0            1  2
  1       1            0           0           1            0  3
          0           10          10          10           24 54
```

# Plots


0.00
0.02
0.04
0.06
0.08
0.10
0.12
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species


V1
V15
V30
V45
V60
V75
V90
V120
V150
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species

# Example: Code

```
# plot of missings
par(mfrow=c(1,2))
k = dim(iris.miss)[2]
freq = numeric(k)
for(i in 1:k) freq = apply(iris.miss, 2,
     function(x)mean(is.na(x)))
barplot(freq, col="yellow")
```

# Example: Code, pt 2

```
NAvec = as.vector(is.na(iris.miss))*1
reverseRows = function(A) A[rev(row(A)[,1]),]
image(t(reverseRows(NAmat)), col=my.col, axes=FALSE)
axis(1, at = seq(0,1, length=k), labels = colnames(NAmat),
    tick=FALSE)
vars = seq(0,150, by=15)
vars[1]=1
ticks = 1-vars/150
axis(2, at = ticks, labels = paste("V",vars, sep=""),
    tick=TRUE)
```

Or, use the functions in the VIM package.

# Example: Code, pt 2

```
NAvec = as.vector(is.na(iris.miss))*1
reverseRows = function(A) A[rev(row(A)[,1]),]
image(t(reverseRows(NAmat)), col=my.col, axes=FALSE)
axis(1, at = seq(0,1, length=k), labels = colnames(NAmat),
    tick=FALSE)
vars = seq(0,150, by=15)
vars[1]=1
ticks = 1-vars/150
axis(2, at = ticks, labels = paste("V",vars, sep=""),
    tick=TRUE)
```

Or, use the functions in the VIM package.

# Example: Imputing the Fisher iris data

```
# do imputation with mice package
imputed_Data = mice(iris.miss, m=5, maxit = 50,
method = "pmm", seed = 500)

for(i in 1:5){
data = complete(imputed_Data,i)
# Discriminant analysis of imputed data

fit.qda = qda(Species~., data = data)
predClasses = predict(fit.qda)$class
qdaTable = table(predClasses, iris$Species)
print(qdaTable)
}
```

# Example: Fisher iris data (cont)

```
predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         1
  virginica       0          2        49

predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48

predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         2
  virginica       0          3        48

predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         49         4
  virginica       0          1        46

predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48
```

# Imputation: Another idea

- Start with a guess for the missing values, using one of the simple imputation methods. Or, alternatively, keep the missing values - RF's can handle them.
- For each variable in turn, predict the missing values using a random forest with the other variables as targets. Fill in the missing values.
- Iterate this until no change.
- Use the imputed data to construct a predictor.

# Advantages and disadvantages

- Works for any data set. Modeling the conditional distributions can be tricky for mixed data sets, since the conditional distribution needs to be estimated.
- Doesn't take account of the uncertainly in the imputation process.
- No method of adjusting the PE to account for the imputation.

# How trees cope with missing values

- When considering splits, we only consider splits of the form $X < c$ where $c$ is one of then non-missing values of $X$.
- We evaluate the splitting criterion ignoring the missing values.
- For each split, we identify "surrogate splits" - splits using different variables that result in similar partitions of the feature space.
- We use these if a case has a missing value in the primary split, when assigning cases to regions.
- When calculating the value of the tree in a region, we ignore missing values in the target.

Since trees cope, so do random forests.

# Using the missForest package

- The function `missForest` in the package of the same name cycles through the variables in the data set, predicting the missing values of that variable using a random forest, using the other variables as features. This process may be repeated several times.
- The outputs are a imputed data set with the missing values filled in, and a set of prediction errors, giving the OOB prediction error for each variable. These give a measure of the success of the imputation.
- See the article "Using the missForest Package" and the Bioinformatics article by Stekhoven and Bühlmann on the web page for more information.
- The function is very easy to use with sensible defaults. You can tweek the random forest settings if desired.

# Doing the imputation

```
ntrees=100
iris.imp = missForest(iris.miss, variablewise=TRUE,
                      ntree=ntrees)
iris.imp$OOB
      MSE       MSE       MSE       MSE       PFC
0.1071421 0.0938676 0.0619656 0.0284400 0.0000000
```

The imputed data is in the data frame `iris.imp$ximp`.

# Example: QDA

```
fit.qda = qda(Species~., data = iris.imp$ximp)
predClasses = predict(fit.qda)$class
qdaTable = table(predClasses, iris$Species)

predClasses  setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         1
  virginica       0          2        49
  > mean(is.na(iris.miss))
[1] 0.08133333
```

In fact, this is the same table that results from the complete data, so we have paid no price for the 8% of missing data.

# An experiment

Suppose we have data following a multivariate normal distribution with mean zero and covariance matrix $\Sigma$ where

$$\Sigma_{ij} = \begin{cases} 1, & i = j \\ \rho, & i \neq j. \end{cases}$$

The data consist of $n = 200$ draws from this distribution. The target is the first variable in the data set and the remaining 19 variables are the features.

# An experiment (cont)

We have four data sets:

1. A training set, as above.
2. A test set generated in the same way.
3. A "missing" data set where the values of the training set have been set to NA with probability $\pi$.
4. An "imputed" data set where the missing values have been inputed using `missForest`.

We calculate (1) the test set estimate of prediction error, using a linear predictor and the complete data, (2) the test set estimate of prediction error, using a linear predictor and the imputed data, and (3) a CV estimate of the prediction error, using the imputed data.

# Results

We show these 3 quantities for different values of $\rho$ and $\pi$.

| | $\pi = 0.05$ | | | $\pi = 0.10$ | | |
|---|---|---|---|---|---|---|
| | Comp | Imp | CV | Comp | Imp | CV |
| $\rho = 0.5$ | 0.60 | 0.60 | 0.54 | 0.57 | 0.57 | 0.53 |
| $\rho = 0.6$ | 0.47 | 0.47 | 0.42 | 0.47 | 0.47 | 0.42 |
| $\rho = 0.7$ | 0.35 | 0.36 | 0.30 | 0.34 | 0.35 | 0.31 |
| $\rho = 0.8$ | 0.24 | 0.24 | 0.21 | 0.24 | 0.24 | 0.22 |
| $\rho = 0.9$ | 0.12 | 0.12 | 0.11 | 0.12 | 0.12 | 0.11 |

| | $\pi = 0.15$ | | | $\pi = 0.20$ | | |
|---|---|---|---|---|---|---|
| | Comp | Imp | CV | Comp | Imp | CV |
| $\rho = 0.5$ | 0.59 | 0.60 | 0.53 | 0.57 | 0.58 | 0.52 |
| $\rho = 0.6$ | 0.48 | 0.48 | 0.41 | 0.47 | 0.47 | 0.40 |
| $\rho = 0.7$ | 0.34 | 0.35 | 0.32 | 0.36 | 0.36 | 0.30 |
| $\rho = 0.8$ | 0.23 | 0.24 | 0.21 | 0.24 | 0.24 | 0.21 |
| $\rho = 0.9$ | 0.12 | 0.12 | 0.11 | 0.12 | 0.12 | 0.11 |

# Points to note

- The effect of the missing values is negligible: Imp error is never much more than the the Comp error.
- The CV estimate under-estimates, so while the predictions aren't much affected, the estimate of error is.
- The degree of underestimation doesn't seem to depend much on either $\rho$ or $\pi$, and is around 85-90%.

# References


1. Analytics Vidhya. Tutorial on 5 Powerful R Packages used for imputing missing values. https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/
2. Stekhoven, D.J. (2011). Using the missForest Package. https://stat.ethz.ch/education/semesters/ss2013/ams/paper/missForest_1.2.pdf
3. Stekhoven, D.J. and Bühlman, P. (2012). MissForest - non-parametric missing value imputation for mixed-type data. `Bioinformatics`, 28, 112-118.
4. van Buuren, S. and Groothuis-Oudshoorn, K. (2011) mice: Multivariate Imputation by Chained Equations in R. `Journal of Statistical Software`, 45 (3).