Monday Group Presentation on Missing Data

Sam Shi

Ryerson University

November 14, 2016

Overview

1. Missing data
   - Types of Missing Data
   - Why it is important
   - Common ways to deal with missing data
2. Methods
   - Listwise deletion
   - Single imputation
   - Multiple imputation
3. Examples with Python and R

Types of Missing Data

supposed you want to model weight Y as a function of gender X and you do a survey asking for Y and X, in the end there are some missing values(data point missing or data attribute missing), below are possible scenarios,

- Missing completely at random(MCAR)
  No particular reasons why the data is missing, such as, it can be someone dropped the survey paper, hardly recognizable hand-writing, etc. (data are rarely MCAR)
- Missing at random(MAR)
  It can happen that one gender X would less likely to disclose their weight information than the other.
- Missing not at random(MNAR)
  Missing value itself is related to why it is missing, e.g. a person with higher weight Y would more likely not fill out the weight blank on survey.

- Easy to occur very common

	Division		
	Alpha	Bravo	Charlie
Sales		$ 11,500,000	
Net operating income		$ 920,000	$ 210,000
Average operating assets	$ 800,000		
Margin	4 %	%	7 %
Turnover	5		
Return on investment (ROI)	%	20 %	14 %

- Nearly all standard statistical methods presume complete information for all the variables included in the analysis; Machine learning models need to have complete input.[Wiki, C4.5]

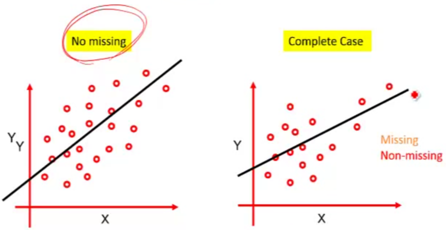
Improvements from ID.3 algorithm [edit]

C4.5 made a number of improvements to ID3. Some of these are:

- Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the thresh
- Handling training data with missing attribute values - C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations.
- Handling attributes with differing costs.
- Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

- Create bias [Missing Data Analysis]

Informative missing



1. Loss of statistical power
2. Regression slope is biased

Common ways to deal with missing data

A quick summary before we introduce some methods, [Computerphile]

- Listwise deletion (or complete case analysis)
- Imputation methods
- Multiple Imputation
- Maximum Likelihood
- Bayesian simulation methods
- Hot deck imputation methods

Listwise deletion

Subject	Age	Gender	Income
1	29	M	$40,000
2	45	M	$36,000
3	81	M	--missing--
4	22	--missing--	$16,000
5	41	M	$98,000
6	33	F	$60,000
7	22	F	$24,000
8	--missing--	F	$81,000
9	33	F	$55,000
10	45	F	$80,000

– picture from Wiki

Advantage:

- Easy to implement, no special computation method requires
- It is valid if the missing data is MCAR
- If the proportion of deleted data is small, e.g. $< 5\%$

Disadvantage:

- Can exclude a large portion of data
- Missing data are MCAR rarely happens in reality
- Introduce bias

Imputation

(Unconditional) Mean imputation: Use mean value of available data to represent missing data.

- Easy to implement, mean stays the same;
- Decreased variance and introduce bias.

Conditional mean imputation: Suppose we want to build an regression model with multiple attributes, and there are some missing values in one of the attribute a_i, then we use all data with avaliable attributes to perform regression on a_i.

- Conditional mean imputations may generate accurate predictions, the uncertainty or imputation error is estimated at all.

(KNN)

Multiple Imputation through Chained Equations(MICE):

Fill in the missing data with random draws from the observed values.

a	b	c	d	y
2	3	8	-1	0
$\boxed{?}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{?}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{?}$	1	1

$\xrightarrow[\text{mean impute}]{\text{Initialize randomly}}$

a	b	c	d	y
2	3	8	-1	0
$\boxed{2}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{0}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{4}$	1	1

Move through the columns of variables and perform singlevariable imputation using some method

a	b	c	d	y
2	3	8	-1	0
$\boxed{2}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{0}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{4}$	1	1

we perform regression on each column, starting with attribute a using the values of all the rest attribute with filled in values:

$$a \sim b + c + d + y$$

suppose this gives us a new value $\hat{a} = -3$, then we update it and keep doing the same for the other attribute, c

Multiple imputation

repeat until a number of iteration converged

a	b	c	d	y
2	3	8	-1	0
$\boxed{-3}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{0}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{4}$	1	1

now we want perform regression on c using the newly updated $\hat{a}$:

$$c \sim \hat{a} + b + d + y$$

then we will obtain a new $\hat{c}$.

Iterate certain number of times or set up a convergence limit.

Finally we do all above steps m times each with different filled in values, and obtain m imputed data sets.

Multiple imputation(PMM)

Previous slides we talked about simple linear regression method, but there are many ways to do multiple imputation, and the default one R MICE uses is *Predictive Mean Matching* (PMM).

a	b	c	d	y
2	3	8	-1	0
$\boxed{?}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{0}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{4}$	1	1

we still start with regression on a using avaliable data with filled in data:

$$a \sim b + c + d + y$$

then we obtain a new $\hat{a}$ vector $[3.5, 2, 5, -3, 0, 6]$.

we pick 3 the closest values (in terms of distance e.g. Euclidean) and randomly choose one of them as the fill-in value.

Multiple imputation(PMM)

a	b	c	d	y	$\hat{a}$
2	3	8	-1	0	$\underline{3.5}$
$\boxed{?}$	2.5	12	-1	1	$\boxed{2}$
4	4	-1.5	0	0	$\underline{5}$
-5	3	$\boxed{0}$	1.2	1	-3
0.5	3	8	-0.5	0	$\underline{0}$
7	9.2	$\boxed{4}$	1	1	6

so we see $[3.5, 5, 0]$ are three values(you can choose different number of points) that closest to $\hat{a} = 2$, so we ranomly choose one out of their corresponding avaliable data, i.e. $[2, 4, 0.5]$

Again, we iterate until satisfied, and repeat this whole process *m* times each with different initializtion.

This is the default multiple imputation method in MICE for continuous variables.

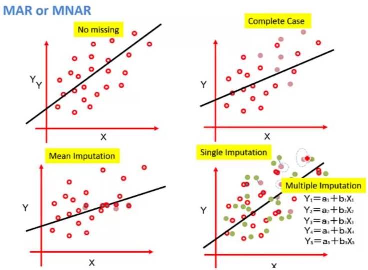
Multiple imputation(PMM)

a	b	c	d	y
2	3	8	-1	0
$\boxed{3.5}$	2.5	12	-1	1
4	4	-1.5	0	0
-5	3	$\boxed{12}$	1.2	1
0.5	3	8	-0.5	0
7	9.2	$\boxed{-1.5}$	1	1

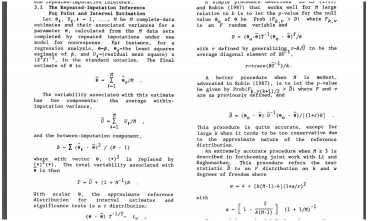
We have obtained one imputated data, then repeat m times, where after you can perform statistical analysis on those m results and you can pool the machine learning parameters from those m results to give you a final model

Use real data to fill in missing data but we are using our imputation model to pick which one we fill in.

Multiple imputation(PMM)



More details on Rubin's paper about statistical analysis:
`http://ww2.amstat.org/sections/SRMS/Proceedings/papers/1988_016.pdf`



one repeated-imputation inference.

3.1 The Repeated-Imputation Inference For Point and Interval Estimation

Let θ_ℓ, U_ℓ, $\ell = 1, \ldots, M$ be M complete-data estimates and their associated variances for a parameter θ, calculated from the M data sets completed by repeated imputations under one model for nonresponse. For instance, for a regression analysis, $\theta = \beta$, θ_ℓ = the least squares estimate of β, and U_ℓ = (residual mean square) x $(X^TX)^{-1}$, in the standard notation. The final estimate of θ is

$$\bar{\theta} = \sum_{\ell=1}^{M} \hat{\theta}_\ell / M \ .$$

The variability associated with this estimate has two components: the average within-imputation variance,

$$\bar{U} = \sum_{\ell=1}^{M} U_\ell / M \ ,$$

and the between-imputation component,

$$B = \sum (\hat{\theta}_\ell - \bar{\theta})^2 / (M - 1)$$

where with vector θ, $(\bullet)^2$ is replaced by $(\bullet)^T(\bullet)$. The total variability associated with θ is then

$$T = \bar{U} + (1 + M^{-1})B \ .$$

With scalar θ, the approximate reference distribution for interval estimates and significance tests is a t distribution:

$$(\theta - \bar{\theta})\, T^{-1/2} \sim t_v \ ,$$

A simple procedure described in Li (1985) and Rubin (1987) that works well for M large relative to k is to let the p-value for the null value θ_o of θ be Prob $(F_{k,v} > D)$ where $F_{k,v}$ is an F random variable and

$$D = (\theta_o - \bar{\theta}) T^{-1} (\theta_o - \bar{\theta})^T / k$$

with v defined by generalizing $r = B/\bar{U}$ to be the average diagonal element of $B\bar{U}^{-1}$,

$$r = \mathrm{trace}(B\bar{U}^{-1})/k.$$

A better procedure when M is modest, advocated in Rubin (1987), is to let the p-value be given by Prob$(F_{k,v(k+1)/2} > \tilde{D})$ where F and v are as previously defined, and

$$\tilde{D} = (\theta_o - \bar{\theta})\, \bar{U}^{-1} (\theta_o - \bar{\theta}) / [(1+r)k] \ .$$

This procedure is quite accurate, except for large k when it tends to be too conservative due to the approximate nature of the reference distribution.

An extremely accurate procedure when $M \geq 3$ is described in forthcoming joint work with Li and Raghunathan. This procedure refers the test statistic $\tilde{D}$ to an F distribution on k and w degrees of freedom where

$$w = 4 + [k(M-1) - 4](1 + a/r)^2$$

with

$$a = \left[1 - \frac{2}{k(M-1)} \right] (1 + 1/M)^{-1}$$

Maximum likelihood

$$p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{N} p(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(m)}|\theta)$$

so now assume first two attribute missing for $i \geq n$ then we have,

$$p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta) = \prod_{i=1}^{n-1} p(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(m)}|\theta) \prod_{j=n}^{N} p(x_i^{(3)}, x_i^{(4)}, ..., x_i^{(m)}|\theta)$$

this can still be maximized.

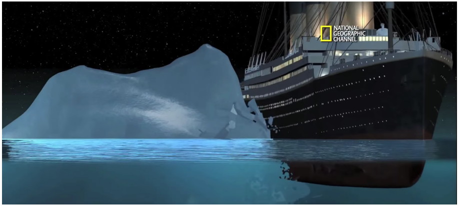
Why ML might be better than MI:

- ML is more efficient than MI.
- For a given set of data, ML always produces the same result. On the other hand, MI gives a different result every time you use it.
- The implementation of MI requires many different decisions, each of which involves uncertainty. ML involves far fewer decisions.
- With MI, there is always a potential conflict between the imputation model and the analysis model. There is no potential conflict in ML because everything is done under one model

[Allison, SAS Global Forum 2012]

Examples

Titanic:



"One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.
In this challenge, we ask you to complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy."
– Kaggle

```
> jupyter notebook Missing_data_demo.ipynb
```

References

- C4.5: https://en.wikipedia.org/wiki/C4.5_algorithm
- Missing Data Analysis: https://www.youtube.com/watch?v=QAvSj2TWZy0
- The Trouble with Missing Data
  https://www.youtube.com/watch?v=oCQbC818KKU
- *Handling Missing Data by Maximum Likelihood*, Paul D. Allison, Statistical Horizons, Haverford, PA, USA