Strategies for Handling Missing Data

Stephen Haptonstahl
Berico Technologies

2012/08/30

The Problem

Data, in theory

Suppose we want to mine sales records to see what kind of customer buys more from us.

sales	female	education	income
1209	0	high school	35000
2587	1	4yr degree	45000
13265	1	grad degree	65000
2298	0	some college	49000

Data, in practice

Suppose we want to mine sales records to see what kind of customer buys more from us.

sales	female	education	income
1209	0	high school	35000
2587	NA	4yr degree	45000
13265	1	NA	65000
NA	0	some college	49000

Who cares?

- Best case if you ignore missing data: throwing away data
- Same as loss of efficiency
- Less certainty (less statistical significance)
- **Some patterns will be completely missed**
- Data scientists are replaced by equally incompetent algorithms, ie.
- Mass hysteria

Today's talk

- The Problem
- Typical Method Versus a Better Method
- Naive Imputation
- Home-Rolled Imputation
- Existing Tools
  - *Small Data Sets*
  - *Large Data Sets*
  - *Real-time*
- When Imputation Fails
- Fin

Typical Method Versus a Better Method

(The Evil that is) Row Deletion

- Idea: Only include full rows in the analysis.
- This is the most common method.
  - *Default for most software, inluding `lm` and other R functions.*
- Throws away data, perhaps a lot.
  - *Ex: 50 fields for each customer*
  - *How many have all 50 observed? (Ans: none except test data)*
- **Some patterns will be completely missed**
- We know where this leads

What is imputation?

Imputation

Filling in missing data with guesses, good or bad, for what would have been observed.

- Example 1: “These scientists all say the earth is warming, but today it’s cold. They must be in a conspiracy!”
- Example 2: “These scientists all say the earth is warming, but today it’s cold. Perhaps they only measured on warm days.”
- Both are wrong, but one is a better (more likely to be true) guess than the other.
- The better guess will be less likely to lead you to bad decisions, algorithms, hysteria, etc.

When is imputation bad

If only we knew what was there. Then we would be able to make a good guess at what was there.

- Missingness process **not ignorable**, which means
- Missingness depends on unobserved data
- Types of data-generating processes
  1. *Missing Completely At Random (MCAR): no data will help you predict which values are missing*
     - Random noise
  2. *Missing at Random (MAR): observed values can help you predict which values are missing*
     - Faulty switches
  3. *Not ignorable (NMAR): unobserved values can help you predict which values are missing*
     - Losses reported as zero income (censored)
- Imputing when missingness is not ignorable leads to bias, bad decisions, etc.
- cf. Donald Rubin (1976)

Naive Imputation

Mean, Median, Mode

We just filled in the missing values with zero.

How we guess (hopefully) depends on what we expect to be there.

- Continuous variable
  - *floating point*
  - *Use `mean`*
- Ordinal (disagree, neither agree nor disagree, agree): `median`
- Count (or other integer): `median`
- Categorical (states; colors): mode

```
StatMode <- function(x) names(table(x))
[which.max(table(x))]
```

Mean, Median, Mode (continued)

Suppose we want to mine sales records to see what kind of customer buys more from us.

sales	female	education	income
1209	0	high school	35000
2587	NA	4yr degree	45000
13265	1	NA	65000
NA	0	some college	49000

Mean, Median, Mode (continued)

Suppose we want to mine sales records to see what kind of customer buys more from us.

sales	female	education	income
1209	0	high school	35000
2587	**0**	4yr degree	45000
13265	1	some college	65000
5687	0	**some college**	49000
mean	**mode**	**median**	

Mean, Median, Mode (continued)

- Pros
  - *Fast*
  - *Principled*
  - *Better than throwing away data*
  - *Tends to work*
- Cons
  - *We can do better.*
  - *"Making up data" makes us more certain than we should be.*

Home-Rolled Imputation

Imputation with Regression

1. Identify a cell `X[i,j]` to impute.
2. Find all rows `prediction.rows` that:
    1. *have an observed value for column `j`*
    2. *have observed values for all other columns `obs.cols` that row `i` has*
3. Using those rows: `reg <- lm(X[prediction.rows,j] ~ X[prediction.rows,obs.cols])`
4. Fill with `predict(reg, names(X)[obs.cols]=X[i,obs.cols])`

Imputation with Regression (continued)

y	x1	x2
7.44	1.21	1.95
3.25	-1.35	0.11
3.90	NA	0.63
5.19	0.16	2.27
5.57	0.17	1.53
…	…	…

Imputation with Regression (continued)

y	x1	x2
7.44	1.21	1.95
3.25	-1.35	0.11
3.90	-0.89	0.63
5.19	0.16	2.27
5.57	0.17	1.53
…	…	…

- Missing value: -0.89
- Predicted value: -0.89

Type of Regression Varies by Variable Type

Continuous

Linear regression with `lm()`

Ordinal

Ordered logit/probit regression with `MASS::polr`

Count

Poisson regression with `glm(…, family=poisson(link = “log”))`

Categorical

Multinomial logit with `nnet::multinom`

Safety First!

- Look out for having too little data to estimate. (Default to mean/median/mode?)
- Data is dirty in other ways (illegal values).
- Iterative algorithms sometimes don't converge.
- Still have problem that "Making up data" makes us more certain than we should be.

Existing Tools

Small Data Sets

`Amelia` (King) and `mi` (Gelman) work differently from home-rolled solutions.

Multiple Imputation

1. Generate m copies of the data set, each with **different** imputed values.
2. Perform the desired analysis on each imputed data set to get the quantity of interest q .
3. Aggregate the results:
    1. *If* $m = 5$ *or so*
        - Point-estimates: `mean(q)`
        - SEs: fairly simple formula
    2. *If* m *large*
        - Treat as simulation result
        - `mean` and `sd`
4. Check distribution of observed versus imputed values (`plot`, `ks.test`)

Amelia Example

```
library(Amelia)
data(africa)
```

```
a.out <- amelia(x = africa, cs = "country", ts = "year", logs = "gdp_pc")
```

```
## -- Imputation 1 --
##
##  1  2  3
##
## -- Imputation 2 --
##
##  1  2  3
##
## -- Imputation 3 --
##
##  1  2  3
##
## -- Imputation 4 --
##
##  1  2
##
## -- Imputation 5 --
##
##  1  2
##
```

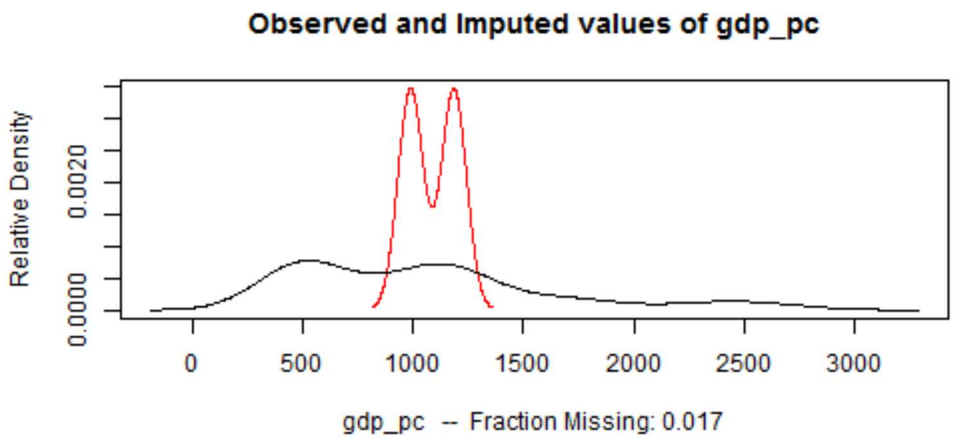
Amelia Example (continued)

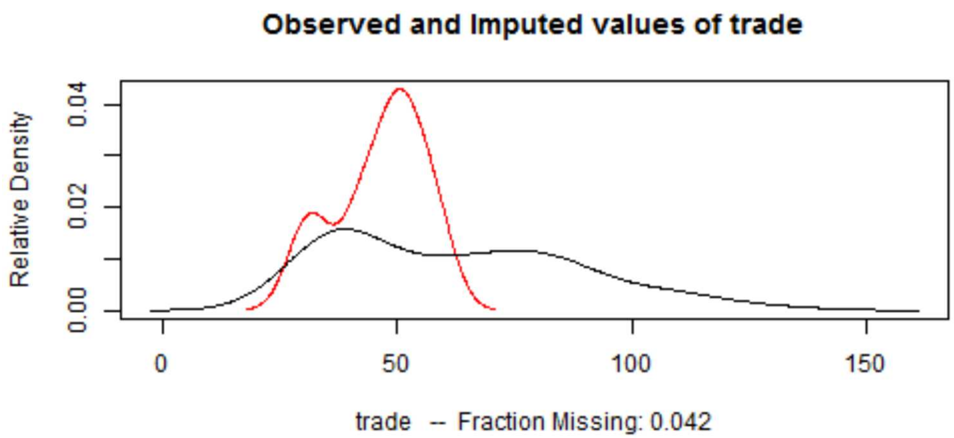
```r
summary(a.out)
```

```
##
## Amelia output with 5 imputed datasets.
## Return code:  1
## Message:  Normal EM convergence.
##
## Chain Lengths:
## --------------
## Imputation 1:  3
## Imputation 2:  3
## Imputation 3:  3
## Imputation 4:  2
## Imputation 5:  2
##
## Rows after Listwise Deletion:  115
## Rows after Imputation:  120
## Patterns of missingness in the data:  3
##
## Fraction Missing for original variables:
## -----------------------------------------
##
##            Fraction Missing
## year                0.00000
## country             0.00000
## gdp_pc              0.01667
## infl                0.00000
## trade               0.04167
## civlib              0.00000
## population          0.00000
##
```

Amelia Example (continued)

```
plot(a.out)
```





That's hard. What have we gained or lost?

- Correct reporting of uncertainty (SEs)
- Actually, it's easier than rolling your own
  - *Smart about pathologies*
  - *Smart about variable types*
  - *Code is shorter if you are implementing in R*
  - *Very fast on reasonably sized data sets*
- If you are **not** implementing in R, you have a lot more work
  - *Reading the paper(s), implementing your own version*
  - *Quite do-able, not for faint at heart*
- Not scalable

Large Data Sets: Bayes

Bayesian data augmentation

- When using Bayes to estimate a model, put a prior on the *data*.
- The algorithm will give draws from the posterior distribution of the missing values.
- It uses the model defined (whatever it is) to impute.

Large Data Sets: Bayes (continued)

Problems

- Slow
- Convergence can be tricky
- Expertise required for custom samplers, but libraries for implementing custom solutions: `rcppbugs` in R, others for C++, Python, etc.

Large Data Sets: Bayes (continued)

Example: Using Bayes for IRT on network links to generate DILS (data-informed link strength)

linkid	**net1**	**net2**	**net3**	**…**	**netk**
1	1	0	0	…	1
2	0	0	1	…	0
3	1	NA	1	…	1
4	1	0	NA	…	0
5	NA	0	0	…	1

becomes…

Large Data Sets: Bayes (continued)

Example: Using Bayes for IRT on network links to generate DILS (data-informed link strength)

linkid	dils
1	.12
2	.01
3	.34
4	.08
5	.03

Large Data Sets: Random Forest Classifier

Easy solution: use `missForest`

```
library(Amelia)  # to provide the data
data(africa)
library(missForest)
```

```
africa.rf <- missForest(africa)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
```

- Yields very good point predictions (single imputation).
- Scales well.
- Requires a **lot** of data (think non-parametric).
- Without care might give silly values.
- Still should check imputed distributions against observed distributions.

Large Data Sets: Random Forest Classifier (continued)

- If you want good measures of uncertainty (multiple imputation) then you'll have to find some way to keep the `ntrees` individual trees used in each of the `nvars` forests.
- Run `randomForest(…, keep.forest=TRUE)` with each column as dependent variable.
  - *Result:* `ntrees` × `nvars` *trees*
- Use `lapply` to get `ntrees` imputed data sets.
  - *Create* `ntrees` *target copies of the original data set.*
  - *For each column use the forest which had that column as the dependent variable.*
  - *For each empty cell generate a predicion for each tree in that forest.*
  - *Store the result for each tree in the corresponding target copy.*
- Run your analysis over each of the imputed data sets.
- Aggregate the results using `mean` and `sd`.

Real-Time: `FastImputation`

- Project: Improve fraud detection classifier
- Same model as Amelia
- Learns (slowly) using lots of data
- Imputes a single row at a time (very, well, fast)
- Available on CRAN
- Caveat emptor (as with all R packages)

When Imputation Fails

To Imputate or Not?

- Assumptions of Imputation
  - *Data is MCAR or MAR+PD*
  - *"Small" fraction of data missing*
    - < 10%: usually no problem
    - 10%-20%: Be careful
    - *20%: There be dragons*
- Always* better than row deletion

What to Do When Imputation Fails

- Explicitly model the reason observations are missing
    - *Tobit (incomes censored, displayed as zero)*
    - *Truncated regression (heights in military)*
    - *Abstentions in voting (more likely if you disagree with party)*

  or

- Impute anyway and warn the consumer (worth considering)

Fin

Links and Contact Info

- This presentation and code: https://github.com/shaptonstahl/MissingData_2012-08
- `Amelia`: http://gking.harvard.edu/amelia (the 2001 paper is a good place to start)
- `mi`: http://cran.r-project.org/web/packages/mi
- Great ref on missing data, imputation: http://gking.harvard.edu/files/evil.pdf
- `FastImputation`: http://cran.r-project.org/web/packages/FastImputation
- Data-Informed Link Strength (DILS): https://github.com/shaptonstahl/dils
- Great place to work: http://www.bericotechnologies.com
- Stephen Haptonstahl – srh@haptonstahl.org – @polimath