GOV51 Section

Text as Data

Pranav Moudgalya

Harvard University

2025-04-24

Reminder

- Missing data is everywhere!
- Three possible mechanisms:
  - Missing completely at random $\rightsquigarrow$ listwise deletion
  - Missing at random $\rightsquigarrow$ multiple imputation
  - Missing not at random $\rightsquigarrow$ more careful modeling
- Dealing with missing values often leads to different study results!

Multiple Imputation

- Say we start with missing value in ideology variable only.
  - Observed: age, gender, education, ideology (only partially)
  - Missing: ideology (only partially)

Multiple Imputation

- Say we start with missing value in ideology variable only.
  - Observed: age, gender, education, ideology (only partially)
  - Missing: ideology (only partially)
- We train a linear regression model using complete cases:

$$\text{Ideology}_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{gender}_i + \beta_3 \cdot \text{edu}_i + \epsilon_i$$

Multiple Imputation

- Say we start with missing value in ideology variable only.
  - Observed: age, gender, education, ideology (only partially)
  - Missing: ideology (only partially)
- We train a linear regression model using complete cases:

$$\text{Ideology}_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{gender}_i + \beta_3 \cdot \text{edu}_i + \epsilon_i$$

- We impute / predict missing ideology answers using this linear model.

Multiple Imputation

- Say we start with missing value in ideology variable only.
  - Observed: age, gender, education, ideology (only partially)
  - Missing: ideology (only partially)
- We train a linear regression model using complete cases:

$$\text{Ideology}_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{gender}_i + \beta_3 \cdot \text{edu}_i + \epsilon_i$$

- We impute / predict missing ideology answers using this linear model.
- Data is now complete.

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as “place holders.”

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as "place holders."
2. The "place holder" mean imputations for one variable ("var") are set back to missing.

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as "place holders."
2. The "place holder" mean imputations for one variable ("var") are set back to missing.
3. The observed values from the variable "var" in Step 2 are regressed on the other variables in the imputation model. In other words, "var" is the dependent variable in a regression model and all the other variables are independent variables in the regression model.

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as “place holders.”
2. The “place holder” mean imputations for one variable (“var”) are set back to missing.
3. The observed values from the variable “var” in Step 2 are regressed on the other variables in the imputation model. In other words, “var” is the dependent variable in a regression model and all the other variables are independent variables in the regression model.
4. The missing values for “var” are then replaced with predictions (imputations) from the regression model.

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as "place holders."
2. The "place holder" mean imputations for one variable ("var") are set back to missing.
3. The observed values from the variable "var" in Step 2 are regressed on the other variables in the imputation model. In other words, "var" is the dependent variable in a regression model and all the other variables are independent variables in the regression model.
4. The missing values for "var" are then replaced with predictions (imputations) from the regression model.
5. Steps 2–4 are then repeated for each variable that has missing data.

Multiple Imputation Procedure

1. A simple imputation, such as imputing the mean, is performed for every missing value in the dataset. These mean imputations can be thought of as "place holders."
2. The "place holder" mean imputations for one variable ("var") are set back to missing.
3. The observed values from the variable "var" in Step 2 are regressed on the other variables in the imputation model. In other words, "var" is the dependent variable in a regression model and all the other variables are independent variables in the regression model.
4. The missing values for "var" are then replaced with predictions (imputations) from the regression model.
5. Steps 2–4 are then repeated for each variable that has missing data.
6. Steps 2–4 are repeated for a number of cycles, with the imputations being updated at each cycle.

Missing Data Implementation

- `mice` package - "Multiple Imputation by Chained Equations''

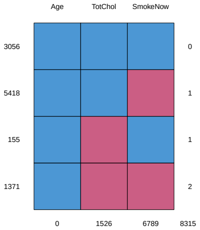
```
library(mice)
library(NHANES)

data(NHANES)
nhanes <- NHANES[c("Age", "SmokeNow", "TotChol")]

# SmokeNow means smokes cigarettes regularly
# TotChol is HDL cholesterol in mmol/L
```

Patterns

```r
md.pattern(nhanes)
```




```
##       Age TotChol SmokeNow
## 3056    1       1        1    0
## 5418    1       1        0    1
## 155     1       0        1    1
## 1371    1       0        0    2
##         0    1526     6789 8315
```

Multiple Imputation

```r
# m specifies the number of imputation cycles
nhanes2MI5 <- mice(nhanes, m = 5)
```

```r
# exports complete data with imputations
df <- complete(nhanes2MI5, 5)
```

Regressions with Imputed Data

```
model1 <- lm(TotChol ~ Age + SmokeNow,
             data = df)
```

	(1)
(Intercept)	4.035***
	(0.028)
Age	0.020***
	(0.000)
SmokeNowYes	0.024
	(0.022)
Num.Obs.	10 000
R2	0.168
R2 Adj.	0.167

$+$ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Comparing with Listwise Deletion

```
model2 <- lm(TotChol ~ Age + SmokeNow,
             data = nhanes)
```

	(1)
(Intercept)	4.775***
	(0.072)
Age	0.006***
	(0.001)
SmokeNowYes	−0.022
	(0.042)
Num.Obs.	3056
R2	0.010
R2 Adj.	0.009

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Multiple Imputation

Questions?

Text as Data: Data Generating Process

- How do we come up with sentences?

Text as Data: Data Generating Process

- How do we come up with sentences?
- One theory is that given a topic (or multiple!), there is a certain probability that words appear

Text as Data: Data Generating Process

- How do we come up with sentences?
- One theory is that given a topic (or multiple!), there is a certain probability that words appear
- In a given *document*, there could be a number of topics
  - Sushi, representation, legislative branch, Felipe's

Text as Data: Data Generating Process

- How do we come up with sentences?
- One theory is that given a topic (or multiple!), there is a certain probability that words appear
- In a given *document*, there could be a number of topics
  - Sushi, representation, legislative branch, Felipe's
- Those topics then dictate the likelihood of which words appear
  - Tuna will probably show up in a document about sushi rather than the legislative branch

Text as Data: Data Generating Process

- How do we come up with sentences?
- One theory is that given a topic (or multiple!), there is a certain probability that words appear
- In a given *document*, there could be a number of topics
  - Sushi, representation, legislative branch, Felipe's
- Those topics then dictate the likelihood of which words appear
  - Tuna will probably show up in a document about sushi rather than the legislative branch
- Pretty rigid framework, but some ground truth
  - We just do the probability weighting implicitly and incredibly quickly

Text as Data: Data Generating Process

- How do we come up with sentences?
- One theory is that given a topic (or multiple!), there is a certain probability that words appear
- In a given *document*, there could be a number of topics
  - Sushi, representation, legislative branch, Felipe's
- Those topics then dictate the likelihood of which words appear
  - Tuna will probably show up in a document about sushi rather than the legislative branch
- Pretty rigid framework, but some ground truth
  - We just do the probability weighting implicitly and incredibly quickly
- Also undergirds the logic under ChatGPT and other large language models (LLMs)
  - Why GPTZero and other programs are easily able to detect because text generated using this framework is rigid!

Key terms in Text as Data

- A corpus (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Key terms in Text as Data

- A corpus (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.
- Within each corpus we will have separate articles, stories, and volumes, each treated as a separate entity or record. Each unit is called a document.

Key terms in Text as Data

- A corpus (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.
- Within each corpus we will have separate articles, stories, and volumes, each treated as a separate entity or record. Each unit is called a document.
- Documents come in a variety of formats, but plain text is often best (e.g. `.txt`, `.csv`)

Key terms in Text as Data

- A corpus (pl: corpora) is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.
- Within each corpus we will have separate articles, stories, and volumes, each treated as a separate entity or record. Each unit is called a document.
- Documents come in a variety of formats, but plain text is often best (e.g. `.txt`, `.csv`)
- Plain text is encoded (i.e., the correspondence between text and binary strings) in different ways. UTF-8 is often easy to use.

Bag of Words Model

- Where do we begin in analysis?

Bag of Words Model

- Where do we begin in analysis?
- One of the most common models is *bag-of-words*
- Text is just a collection of words - the order and structure do not matter particularly when we want to get information like topical relevancy
  - Hence, bag-of-words
- Assumption is that the frequency of words can provide us information about the context in the text

Example

“Does TikTok access the home Wi-Fi Network' ' - Richard Hudson (R-NC)

“Tiktok is fun' ' - Jeremiah Cha (Not in Congress-CA)

Example Table

i	Does	TikTok	access	the	home	Wi-Fi	network	is	fun
1	1	1	1	1	1	1	1	0	0
2	0	1	0	0	0	0	0	1	1

General Preprocessing methods

- One (of many) recipe for preprocessing: retain useful information through dimension reduction.
  - Remove capitalization, punctuation
  - Discard word order (Bag of Words assumption)
  - Discard stop words
  - Combine similar terms: Stem, Lemmatize
  - Discard less useful features $\rightarrow$ depended on application (e.g., numbers)
  - Other reduction, weighting
  - Output: count vector, each element counts occurrence of terms

Remove capitalization

- Assumption: capitalization and punctuation do not provide useful information.
  - `Now we are engaged in a great civil war, testing whether that nation, or any nation`
  - `now we are engaged in a great civil war testing whether that nation or any nation`
- Caution: capitalization might be meaningful given the context (e.g., “Turkey” = “turkey”)

Discard word order (bag of words)

- Assumption: Word Order Doesn't Matter.
  - `now we are engaged in a great civil war testing whether that nation or any nation`
  - `[now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]`
  - `= [a, any, are, civil, engaged, great, in, nation, now, or, testing, that, war, we, whether]`

Tokenization

- **Unigrams**: [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]
- **Bigrams**: [now we, we are, are engaged, engaged in, in a, a great, great civil, civil war, war testing, testing whether, whether that, that nation, nation or, or any, any nation]
- **Trigrams**: [now we are, we are engaged, are engaged in, engaged in a, in a great, a great civil, great civil war, civil war testing, war testing whether, testing whether that, whether that nation, that nation or, nation or any, or any nation]

Tokenization

- **Unigrams**: [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]
- **Bigrams**: [now we, we are, are engaged, engaged in, in a, a great, great civil, civil war, war testing, testing whether, whether that, that nation, nation or, or any, any nation]
- **Trigrams**: [now we are, we are engaged, are engaged in, engaged in a, in a great, a great civil, great civil war, civil war testing, war testing whether, testing whether that, whether that nation, that nation or, nation or any, or any nation]
- Why is this important?

Tokenization

- **Unigrams**: [now, we, are, engaged, in, a, great, civil, war, testing, whether, that, nation, or, any, nation]
- **Bigrams**: [now we, we are, are engaged, engaged in, in a, a great, great civil, civil war, war testing, testing whether, whether that, that nation, nation or, or any, any nation]
- **Trigrams**: [now we are, we are engaged, are engaged in, engaged in a, in a great, a great civil, great civil war, civil war testing, war testing whether, testing whether that, whether that nation, that nation or, nation or any, or any nation]
- Why is this important? $\rightsquigarrow$ Important concepts are encoded in more than one word (e.g., "NOT good")

Stemming and Lemmatizing

- Reduce dimensionality further by combining similar terms through stemming or lemmatizing
- Stemming/Lemmatizing algorithms: Many-to-one mapping from words to stem/lemma
- Stemming algorithm:
  - Simplistic algorithms
  - Chop off end of word (change, changing, changed, changer $\rightarrow$ chang)
  - Types: Porter stemmer, Lancaster stemmer, Snowball stemmer
- Lemmatizing algorithm:
  - Condition on part of speech (noun, verb, etc)
  - Verify result is a word (change, changing, changed, changer $\rightarrow$ change)

Other common preprocessing techniques

- Remove sparse words (rare words)
- Remove other terms (e.g. proper nouns)
- Weight some terms more than others (term frequency–inverse document frequency, or, tf-idf)

Applications

1. Topic modeling
2. Sentiment analysis
3. Text classification

- Also a great application of Lasso, since data can get VERY large

Applications

1. Topic modeling
2. Sentiment analysis
3. Text classification

- Also a great application of Lasso, since data can get VERY large
- **CAUTION**: Text analysis is very useful for getting ideas about general concepts or similarity among documents but is very limited. We lose sarcasm, irony, tone, and more by tokenizing words.

Summary

- Multiple imputation is easily implemented through the `mice` package
  - Introduces a number of easy to use functions that help with descriptive statistics and modeling
- Text as data is incredibly popular and powerful tools in social science
- Data generating process of text is an important foundation for understanding how to tackle text as data
- Bag-of-words is a simple, but powerful model to analyze text
- Remember to schedule a meeting - this is a **requirement** for the final project!