My Comprehensive Evaluation

A Comprehensive Evaluation Report
Presented to
The Statistics Faculty
Amherst College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts
in
Statistics

Kaitlyn E. Haase

February 2019

Acknowledgements

I want to thank my family.

Table of Contents

Introduction **1**

0.1 Why Analyze Spatial Data? 1

0.2 Big Picture Clustering Algorithms 1

0.2.1 Classification vs. Clustering 1

Chapter 1: Spatial Clustering Methods **3**

1.1 Types of Clustering: Partitioning and Hierarchical 3

1.2 How to Create Clusters: K-Means vs K-Medoids 4

1.3 PAM 4

1.4 CLARA 4

1.5 CLARANS (?) 5

Chapter 2: Example **7**

2.1 Exploring the Data 7

2.2 Appplying CLARA 7

2.3 Evaluation of CLARA 7

2.3.1 Model to Predict Cluster 7

Chapter 3: Tables, Graphics, References, and Labels **9**

3.1 Tables 9

3.2 Figures 11

3.3 Footnotes and Endnotes 15

3.4 Bibliographies 15

Conclusion **17**

Appendix A: The First Appendix **19**

Appendix B: The Second Appendix, for Fun **21**

References **23**

List of Tables

3.1 Correlation of Inheritance Factors for Parents and Child 9

List of Figures

3.1 Amherst logo 11
3.2 Mean Delays by Airline 13
3.3 Subdiv. graph 14
3.4 A Larger Figure, Flipped Upside Down 14
3.5 Subdivision of arc segments 15

Abstract

In recent years, the amount of geographic data has increased immensely. With new technology, the accuracy and complexity of data has also improved. This has provoked statisticians to create techniques to best analyze and draw conclusions from this new-found data. Earlier techniques of spatial data were not equipped to handle the complexity and amount of present data. This project first explores how and why we analyze data based on geographic information. The project will then focus on the CLARANS (Clustering Large Applications based on RANdomized Search) algorithm, which is an extension of both the PAM (Partitioning Around Medoids) and CLARAS (Clustering LARge Applications). Example data will be used to demonstrate CLARANS, and the project will conclude by testing how accurate CLARANS clustered the data.

Introduction

Data includes: latitude and longtitude, zip code, street address, etc.

0.1 Why Analyze Spatial Data?

We want to analyze spatial data because there is so much of it available.

0.2 Big Picture Clustering Algorithms

There are many clustering algorithms out there.

0.2.1 Classification vs. Clustering

Two categories of how to analyze spatial data include classification and clustering. Classification is BLANK Clustering is organizing a set of data items into groups so that items in the same group are similar to each other and different from those in other groups [Rec 1]. Clustering is helpful in finding patterns and similarities/differences between data points and groups; however it can be quite subjective, as we will discuss later on in the project.

Chapter 1

Spatial Clustering Methods

There are many factors to consider when chosing a clustering algorithm, such as the application of the problem (what do you want to find out about this data?), quality vs speed trade off (size of data plays a role), characteristics of the data (i.e. numeric distance measures), dimensionality (typically as dimension increases the time it takes to run the method increases and quality of the data clusters decrease), and outliers (some methods are very sensitive to outliers) [Rec 2].

1.1 Types of Clustering: Partitioning and Hierarchical

There are many types of clustering algorithms, two of which are: partitioning and hierarchial.

Hierarchial clustering organizes data items into a hierarchy with a sequence of nested partions or groupings [Rec 1, p. 405]. There is the bottom-up approach: There is also the top-down approach:

Partitioning cluster methods divide a set of data items into a number of non-overlapping clusters. A data item is typically assigned to a cluster based on a proximity or dissimilarity measure [Rec 2, p. 405]. Partitioning clustering algorithms classifies the data into K groups by satisfying both that each group has at least one data point, and that each data point belongs to exactly one group. [Rec 5, p. 18].

1.2 How to Create Clusters: K-Means vs K-Medoids

K-means algorithm and k-medoid algorithm are two examples of partitioning algorithms. They both ise iterative processes to find K clusters; however, they use different ways to represent these clusters.

K-means algorithm represents its n observations in k groups, with the center of the groups being the mean/average observation.

Steps on p. 4, Rec 2 Also, it tries to minimize the objective function, steps on this page: * Rec 5, p. 18

Instead of taking the mean value of the objects in a cluster, the k-medoid method uses the most centrally located object in a cluster to be the cluster center [Rec 2]. This causes the method to be less sensitive to outliers, but also requires more time to run.

same steps as K-means except BLANK (p. 6 in Rec 2) steps in Rec 5, p. 19 for k-medoids

1.3 PAM

Partitioning Around Medoids (PAM) is a k-medoid method that iterates through all the k cluster centers and tries to replace the center with one of the other objects (n-k possibilities). [rec 2]. For a replacement to occur, the squared error function must decrease (if it does not decrease, there is no replacement). The algorithm eventually terminates with a local optimum.

The total complexity of PAM in one iteration is **formula: O(k(n-k)ˆ2) (o= each non-medoid data point, k= # of cluster centers, (n-k) objects to compare to, and (n-k) operations for calculating E). This makes for a costly computation when n is large. Works best for n= 100, k=5.

1.4 CLARA

Because PAM does not scale well to large data sets, Clustering LARge Applications (CLARA) was developed to deal with larger data sets.

CLARA is a sampling based method, meaning a sample of the data is used to represent the entire data set. Medoids are chosen from this sample data using PAM and then “the average dissimilarity is computed using the whole dataset” (**don’t know what "average dissimilarity" means or how it is calculated). If a new set of medoids gives a lower dissimilarity than a previous best solution, then the best solution is replaced with a new set of medoids [Rec 2, p. 7].

1.5 CLARANS (?)

Chapter 2

Example

2.1 Exploring the Data

2.2 Appplying CLARA

2.3 Evaluation of CLARA

2.3.1 Model to Predict Cluster

Chapter 3

Tables, Graphics, References, and Labels

3.1 Tables

In addition to the tables that can be automatically generated from a data frame in **R** that you saw in [R Markdown Basics] using the `kable` function, you can also create tables using *pandoc*. (More information is available at `http://pandoc.org/README.html#tables`.) This might be useful if you don't have values specifically stored in **R**, but you'd like to display them in table form. Below is an example. Pay careful attention to the alignment in the table and the use of the hyphens to create the rows and columns.

Table 3.1: Correlation of Inheritance Factors for Parents and Child

Factors	Correlation between Parents & Child	Inherited
Education	-0.49	Yes
Socio-Economic Status	0.28	Slight
Income	0.08	No
Family Size	0.18	Slight
Occupational Prestige	0.21	Slight

We can also create a link to the table by doing the following: Table 3.1. If you go back to [Loading and exploring data] and look at the `kable` function code, you'll see that I added in a similar `\\label` to be able to reference that table later. (The extra backslash there is a way that *Markdown* interfaces with LaTeX.) We can create a reference to the max delays table: **??**.

The addition of the `\label{}` option to the end of the table caption allows us to then use the LaTeX `autoref` function to produce the link. The `ref` function in **R** allows for tables and figures to be referenced in the document easily without having to directly use the `autoref` function. It will automatically add “Table” before your number if you add the “tab:” prefix to your label. Note that this reference could appear anywhere throughout the document.

3.2 Figures

If your thesis has a lot of figures, *R Markdown* might behave better for you than that other word processor. One perk is that it will automatically number the figures accordingly in each chapter. You'll also be able to create a label for each figure, add a caption, and then reference the figure in a way similar to what we saw with tables earlier. If you label your figures, you can move the figures around and *R Markdown* will automatically adjust the numbering for you. No need for you to remember! So that you don't have to get too far into LaTeX to do this, a couple **R** functions have been created for you to assist. You'll see their use below.

In the **R** chunk below, we will load in a picture stored as `amherst.png` in our main directory. We then give it the caption of "Amherst logo", the label of "amherst", and specify that this is a figure. Note again the use of the `results = "asis"` specification to automatically include and compile the LaTeX code.

```r
label(path = "figure/amherst.png", caption = "Amherst logo",
      label = "amherst", type = "figure")
```

Figure 3.1: Amherst logo

Here is a reference to the Amherst logo: Figure 3.1. Note the use of the inline **R** code here. By default "figure" is specified as the type. For clarity, we could have also added the `label` and `type` to the parameter specifications and this would give us the same result: Figure 3.1.

Below we will investigate how to save the output of an **R** plot and label it in a way similar to that done above. Recall the `flights` dataset from . (Note that we've shown a different way to reference a section or chapter here.) We will next explore a bar graph with the mean flight departure delays by airline from Portland for 2014. Note also the use of the `scale` parameter which is discussed on the next page.

```r
delay_airline <- flights %>% group_by(carrier) %>%
  summarize(mean_dep_delay = mean(dep_delay)) %>%
  ggplot(aes(x = carrier, y = mean_dep_delay)) +
  geom_bar(position = "identity", stat = "identity", fill = "red")
ggsave("figure/delays.png", plot = delay_airline,
       width = 5, height = 3)
```

```r
label(path = "figure/delays.png",
      caption = "Mean Delays by Airline",
      label = "delays", type = "figure",
      scale = 0.3)
```

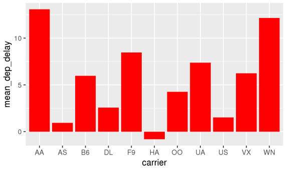

Figure 3.2: Mean Delays by Airline

A table linking these carrier codes to airline names is available at `https://github.com/ismayc/pnwflights14/blob/master/data/airlines.csv`.

Next, we will explore the use of the `scale` parameter which can be used to shrink or expand an image. Here we use the mathematical graph stored in the "subdivision.pdf" file. Note that we didn't specify the `caption =` or `label =` here, but we could have.

```
label("figure/subdivision.pdf", "Subdiv. graph", "subd",
      scale = 0.75)
```

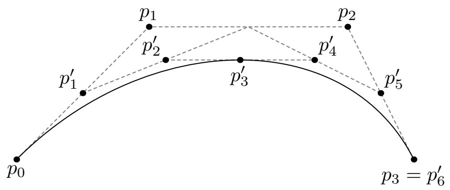

Figure 3.3: Subdiv. graph

Here is a reference to this image: Figure 3.3. (Move this around throughout the document as you wish.)

More Figure Stuff

Lastly, we will explore how to rotate figures using the `angle` parameter.

```
label("figure/subdivision.pdf",
      "A Larger Figure, Flipped Upside Down",
      scale = 1.5,
      angle = 180,
      label = "subd2")
```

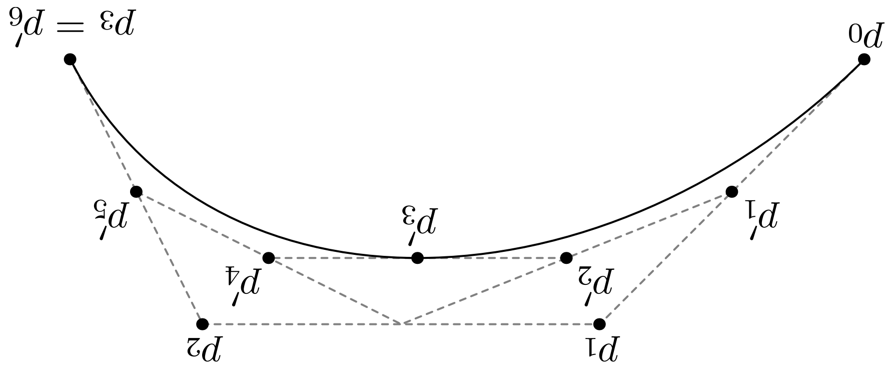

Figure 3.4: A Larger Figure, Flipped Upside Down

As another example, here is a reference to this figure: Figure 3.4.

Common Modifications

The following figure features the more popular changes thesis students want to their figures. We can add math to the caption that displays below the picture, specify the size of our caption to display below the figure (list of sizes available at this link), and also specify that a different caption `alt.cap` be what appears in the Table of Figures for this figure.

If you'd like to make further tweaks to figures, you might need to invoke some LaTeX code.

```
label("figure/subdivision.pdf",
      caption = "Subdivision of arc segments",
      alt.cap = "You can see that $p_3 = p_6^\\prime$",
      cap.size = "footnotesize",
      label = "subd3")
```

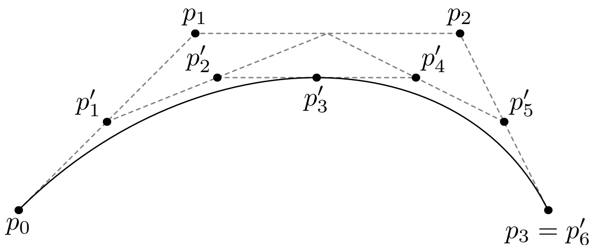

Figure 3.5: You can see that $p_3 = p'_6$

3.3 Footnotes and Endnotes

You might want to footnote something.[1] The footnote will be in a smaller font and placed appropriately. Endnotes work in much the same way.

3.4 Bibliographies

Of course you will need to cite things, and you will probably accumulate an armful of sources. There are a variety of tools available for creating a bibliography database (stored with the .bib extension). In addition to BibTeX suggested below, you may want to consider using the free and easy-to-use tool called Zotero.

[1] footnote text

R Markdown uses *pandoc* (`http://pandoc.org/`) to build its bibliographies. One nice caveat of this is that you won't have to do a second compile to load in references as standard LaTeX requires. To cite references in your thesis (after creating your bibliography database), place the reference name inside square brackets and precede it by the "at" symbol. For example, here's a reference to a book about worrying: (Molina & Borkovec, 1994). This `Molina1994` entry appears in a file called `thesis.bib` in the `bib` folder. This bibliography database file was created by a program called BibTeX. You can call this file something else if you like (look at the YAML header in the main .Rmd file) and, by default, is to placed in the `bib` folder.

For more information about BibTeX and bibliographies, see Reed College's CUS site (`http://web.reed.edu/cis/help/latex/index.html`)[2]. There are three pages on this topic: *bibtex* (which talks about using BibTeX, at `http://web.reed.edu/cis/help/latex/bibtex.html`), *bibtexstyles* (about how to find and use the bibliography style that best suits your needs, at `http://web.reed.edu/cis/help/latex/bibtexstyles.html`) and *bibman* (which covers how to make and maintain a bibliography by hand, without BibTeX, at `http://web.reed.edu/cis/help/latex/bibman.html`). The last page will not be useful unless you have only a few sources.

If you look at the YAML header at the top of the main .Rmd file you can see that we can specify the style of the bibliography by referencing the appropriate csl file. You can download a variety of different style files at `https://www.zotero.org/styles`. Make sure to download the file into the csl folder.

Tips for Bibliographies

- Like with thesis formatting, the sooner you start compiling your bibliography for something as large as thesis, the better. Typing in source after source is mind-numbing enough; do you really want to do it for hours on end in late April? Think of it as procrastination.
- The cite key (a citation's label) needs to be unique from the other entries.
- When you have more than one author or editor, you need to separate each author's name by the word "and" e.g. `Author = {Noble, Sam and Youngberg, Jessica},`.
- Bibliographies made using BibTeX (whether manually or using a manager) accept LaTeX markup, so you can italicize and add symbols as necessary.
- To force capitalization in an article title or where all lowercase is generally used, bracket the capital letter in curly braces.

[2] Reed College (2007)

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{.unnumbered}` attribute. This has an unintended consequence of the sections being labeled as 3.6 for example though instead of 4.1. The LaTeX commands immediately following the Conclusion declaration get things back on track.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file:

```r
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
}
library(acstats)
```

In :

```r
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(dplyr))
    install.packages("dplyr", repos = "http://cran.rstudio.com")
if(!require(ggplot2))
    install.packages("ggplot2", repos = "http://cran.rstudio.com")
```

```r
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
  }
library(acstats)
flights <- read.csv("data/flights.csv")
```

Appendix B

The Second Appendix, for Fun

References

Angel, E. (2000). *Interactive computer graphics : A top-down approach with opengl.* Boston, MA: Addison Wesley Longman.

Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with quicktime.* Boston, MA: Wesley Addison Longman.

Angel, E. (2001b). *Test second book by angel.* Boston, MA: Wesley Addison Longman.

Molina, S. T., & Borkovec, T. D. (1994). The Penn State worry questionnaire: Psychometric properties and associated characteristics. In G. C. L. Davey & F. Tallis (Eds.), *Worrying: Perspectives on theory, assessment and treatment* (pp. 265–283). New York: Wiley.

Reed College. (2007, March). LaTeX your document. Retrieved from `http://web.reed.edu/cis/help/LaTeX/index.html`