ISA 401: Business Intelligence & Data Visualization

24: A Short Introduction to Clustering

Fadel M. Megahed, PhD

Associate Professor
Department of Information Systems and Analytics
Farmer School of Business
Miami University

Twitter: FadelMegahed
GitHub: fmegahed
Email: fmegahed@miamioh.edu
Office Hours: Automated Scheduler for Virtual Office Hours

Spring 2022

A Recap of What we Learned Last Class

- Describe the goals & functions of data mining
- Understand the statistical limits on data mining
- Describe the data mining process
- What is “frequent itemsets” & the application of this concept
- Explain how and why “association rules” are constructed
- Use R to populate both concepts

Learning Objectives for Today's Class

- Describe the different steps of the k-means algorithm
- Cluster using k-means (by hand)
- Cluster using k-means (software)
    - R
    - Tableau

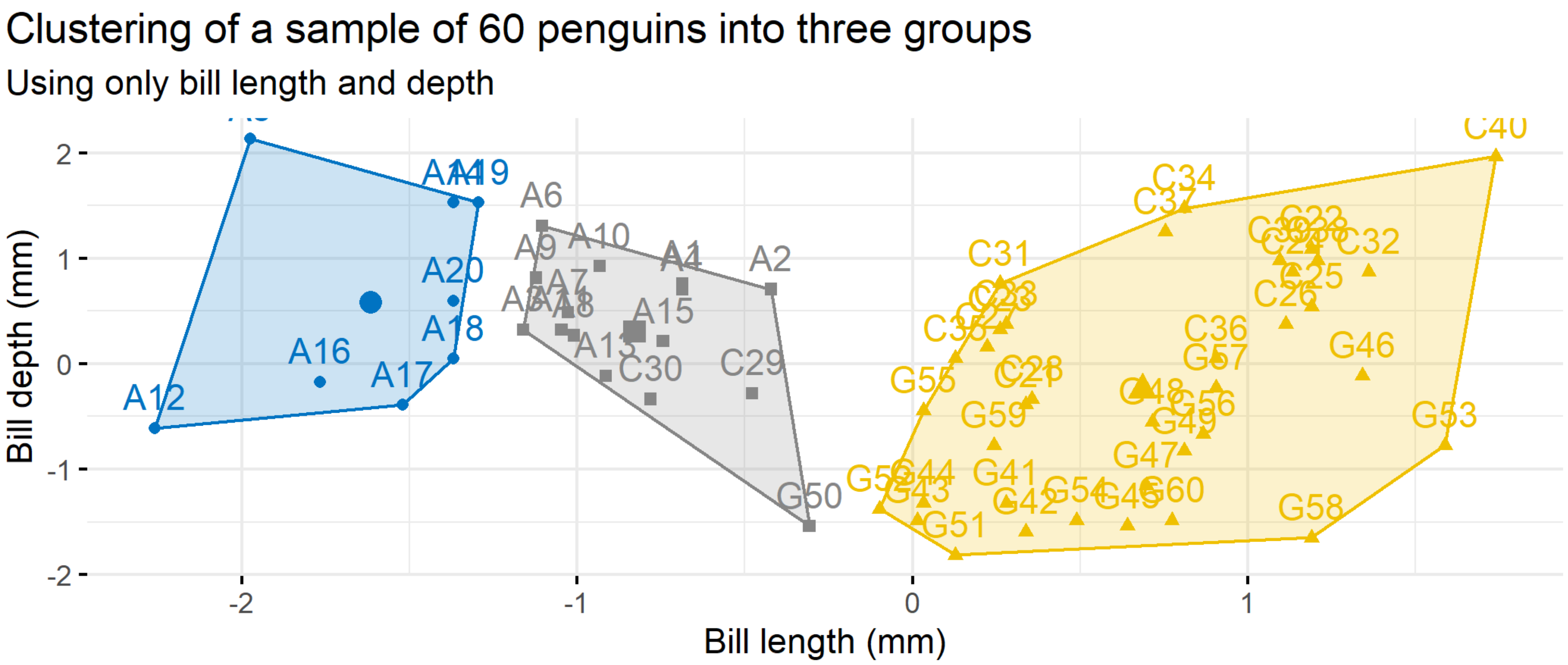
An Overview of Clustering Techniques

The Problem of Clustering

- Given a **set of (high-dimensional) observations**, with a notion of **distance** between observations, **group the observations** into **some number of clusters**, so that:
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar
- **Usually:**
  - The observations are in a high-dimensional space
  - Similarity is defined using a distance measure, e.g.,
    - Euclidean, Cosine, Jaccard, edit distance, etc.

Source: J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Clustering in 2D Space: Clustering Results



Source: The data are available by CC-0 license in accordance with the Palmer Station LTER Data Policy and the LTER Data Access Policy for Type I data. The artwork is by Allison Horst and available at https://allisonhorst.github.io/palmerpenguins/, and the data is downloaded using the palmerpenguins R package.

Comments on the 2D Clustering Problem

Even though the 2D Space clustering problem is the easiest problem to "solve" since we can benefit by plotting the data, **clustering is hard**.

Some important questions:

- With all the variables being numerical, we often assume **Euclidean distance**. This can be problematic when:
  - variables have significantly different scales
  - we are including information that is not pertinent to grouping
- How do you determine the number of clusters (k)?
- How to represent a cluster of many points?
- How do we determine the "nearness" of clusters?

An Overview of Clustering Methods

Categories	Abb. name	Volume			Variety		Velocity	Other criterion
		Size of Dataset	Handling High Dimensionality	Handling Noisy Data	Type of Dataset	Clusters Shape	complexity of Algorithm	Input Parameter
Partitional algorithms	K-Means [25]	Large	No	No	Numerical	Non-convex	O(nkd)	1
	K-modes [19]	Large	Yes	No	Categorical	Non-convex	O(n)	1
	K-medoids [33]	Small	Yes	Yes	Categorical	Non-convex	O(n^2dt)	1
	PAM [31]	Small	No	No	Numerical	Non-convex	O(k$(n-k)^2$)	1
	CLARA [23]	Large	No	No	Numerical	Non-convex	O(k$(40+k)^2$+k(n-k))	1
	CLARANS [32]	Large	No	No	Numerical	Non-convex	O(kn^2)	2
	FCM [6]	Large	No	No	Numerical	Non-convex	O(n)	1
Hierarchical algorithms	BIRCH [40]	Large	No	No	Numerical	Non-convex	O(n)	2
	CURE [14]	Large	Yes	Yes	Numerical	Arbitrary	O(n^2log n)	2
	ROCK [15]	Large	No	No	Categorical and Numerical	Arbitrary	O(n^2+nmmma+n^2logn)	1
	Chameleon [22]	Large	Yes	No	All type of data	Arbitrary	O(n^2)	3
	ECHIDNA [26]	Large	No	No	Multivariate Data	Non-convex	$O(N * B(1 + \log_B m))$	2
Density-based algorithms	DBSCAN [9]	Large	No	No	Numerical	Arbitrary	O(nlogn) If a spatial index is used Otherwise, it is O(n^2).	2
	OPTICS [5]	Large	No	Yes	Numerical	Arbitrary	O(nlogn)	2
	DBCLASD [39]	Large	No	Yes	Numerical	Arbitrary	O(3n^2)	No
	DENCLUE [17]	Large	Yes	Yes	Numerical	Arbitrary	O(log$\|D\|$)	2
Grid- based algorithms	Wave-Cluster [34]	Large	No	Yes	Special data	Arbitrary	O(n)	3
	STING [37]	Large	No	Yes	Special data	Arbitrary	O(k)	1
	CLIQUE [21]	Large	Yes	No	Numerical	Arbitrary	O(Ck + mk)	2
	OptiGrid [18]	Large	Yes	Yes	Special data	Arbitrary	Between O(nd) and O(nd log n)	3
Model- based algorithms	EM [8]	Large	Yes	No	Special data	Non-convex	O(knp)	3
	COBWEB [12]	Small	No	No	Numerical	Non-convex	O(n^2)	1
	CLASSIT [13]	Small	No	No	Numerical	Non-convex	O(n^2)	1
	SOMs [24]	Small	Yes	No	Multivariate Data	Non-convex	O(n^2m)	2

Source: A. Fahad, et al.,"A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis" in IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 03, pp. 267-279, 2014. https://doi.org/10.1109/TETC.2014.2330519

k-means Algorithm

General Idea

The K-Means algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the **inertia** or **within-cluster sum-of-squares** (see below). This algorithm requires the **number of clusters to be specified**.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

Inertia is a measure of how internally coherent clusters are; however, it suffers from various drawbacks:

- Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes.
- Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated.

Source: Clustering — scikit-learn 1.0.2 documentation https://scikit-learn.org/stable/modules/clustering.html#k-means

The Steps of the K-Means Algorithm

In basic terms, the algorithm has three steps.

- Step 0 chooses the initial centroids, with the most basic method being to choose $k$samples from the dataset X . After initialization, K-means consists of looping between the remaining two steps.
- Step 1 assigns each sample to its nearest centroid.
- Step 2 creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed.

  The algorithm repeats these last two steps the centroids do not move significantly.

Source: Clustering — scikit-learn 1.0.2 documentation https://scikit-learn.org/stable/modules/clustering.html#k-means

Class Activity

Use the k-means algorithm to cluster the following observations. Use $k = 2$ and Euclidean distance. **We will use the class handout to walk you through the process.**

Observation	X1	X2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | Optimal k | Clusters By Metrics | Viz Clusters

```r
pacman::p_load(tidyverse, palmerpenguins, magrittr)

penguins_tbl = penguins # our data for today

penguins_tbl # printing it out
```

```
## # A tibble: 344 x 8
##    species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##    <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
##  1 Adelie  Torgersen           39.1          18.7               181        3750
##  2 Adelie  Torgersen           39.5          17.4               186        3800
##  3 Adelie  Torgersen           40.3          18                 195        3250
##  4 Adelie  Torgersen           NA            NA                  NA          NA
##  5 Adelie  Torgersen           36.7          19.3               193        3450
##  6 Adelie  Torgersen           39.3          20.6               190        3650
##  7 Adelie  Torgersen           38.9          17.8               181        3625
##  8 Adelie  Torgersen           39.2          19.6               195        4675
##  9 Adelie  Torgersen           34.1          18.1               193        3475
## 10 Adelie  Torgersen           42            20.2               190        4250
## # ... with 334 more rows, and 2 more variables: sex <fct>, year <int>
```

Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | Optimal k | Clusters By Metrics | Viz Clusters

```r
penguins_tbl %<>%
  # selecting relevant cols
  select(species, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g) %>%
  na.omit() %>% # removing NAs
  mutate_at(vars(-species), scale) # scaling numeric variables

penguins_tbl # printing it out
```

```
## # A tibble: 342 x 5
##    species bill_length_mm[,1] bill_depth_mm[,1] flipper_length_~ body_mass_g[,1]
##    <fct>                <dbl>             <dbl>            <dbl>           <dbl>
##  1 Adelie              -0.883            0.784            -1.42          -0.563
##  2 Adelie              -0.810            0.126            -1.06          -0.501
##  3 Adelie              -0.663            0.430            -0.421         -1.19
##  4 Adelie              -1.32             1.09             -0.563         -0.937
##  5 Adelie              -0.847            1.75             -0.776         -0.688
##  6 Adelie              -0.920            0.329            -1.42          -0.719
##  7 Adelie              -0.865            1.24             -0.421          0.590
##  8 Adelie              -1.80             0.480            -0.563         -0.906
##  9 Adelie              -0.352            1.54             -0.776          0.0602
## 10 Adelie              -1.12            -0.0259           -1.06          -1.12
## # ... with 332 more rows
```

Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | Optimal k | Clusters By Metrics | Viz Clusters

```
km_res = kmeans(x = penguins_tbl %>% select(-species), # input data with no label
                centers = 3) # k =3

# tabulating the results with rows corresponding to true labels and the columns correspondin
table(penguins_tbl$species, km_res$cluster)
```

```
##
##                1   2   3
##   Adelie       0   0 151
##   Chinstrap    0   1  67
##   Gentoo      66  57   0
```

Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | **Optimal k** | Clusters By Metrics | Viz Clusters

```r
pacman::p_load(NbClust)

km_res_nbclust = NbClust(
  data = penguins_tbl %>% select(-species),
  distance = "euclidean",
  min.nc = 2, max.nc = 10,
  method = "kmeans", index ="all")

table(penguins_tbl$species, km_res_nbclust$Best.partition)
```

```
## *** : The Hubert index is a graphical method of determining the number of clusters
##                 In the plot of Hubert index, we seek a significant knee that corre
##                 significant increase of the value of the measure i.e the significa
##                 index second differences plot.
##

## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significan
##                 second differences plot) that corresponds to a significant increas
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 3 proposed 5 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************

##
##               1   2   3
##   Adelie      8   0 143
##   Chinstrap  63   0   5
##   Gentoo      0 123   0
```

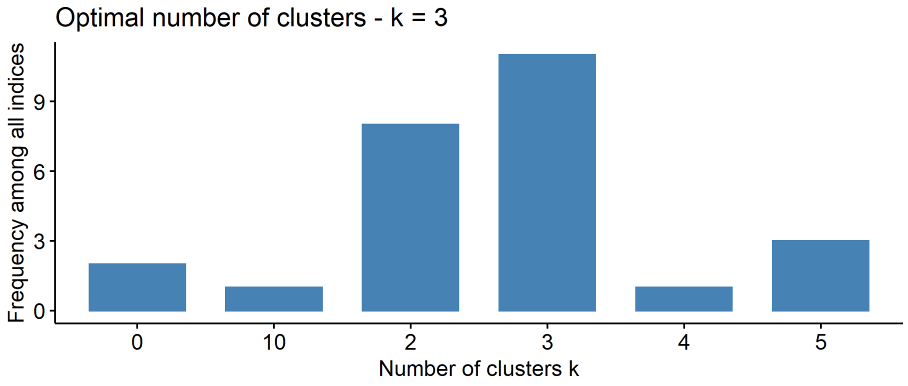
Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | Optimal k | **Clusters By Metrics** | Viz Clusters

```r
pacman::p_load(factoextra)

fviz_nbclust(km_res_nbclust, ggtheme = theme_minimal())
```

```
## Among all indices:
## ===================
## * 2 proposed  0 as the best number of clusters
## * 8 proposed  2 as the best number of clusters
## * 11 proposed  3 as the best number of clusters
## * 1 proposed  4 as the best number of clusters
## * 3 proposed  5 as the best number of clusters
## * 1 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  3 .
```

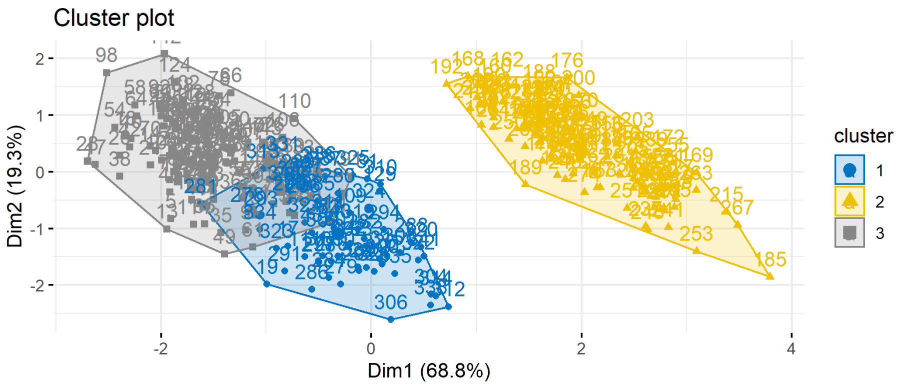


Practical Issues with K-Means Clustering

Data | Prep | K-means (k=3) | Optimal k | Clusters By Metrics | **Viz Clusters**

```
pacman::p_load(factoextra)

fviz_cluster(object =
               list(
                 cluster = km_res_nbclust$Best.partition,
                 data = penguins_tbl %>% select(-species)
               ),
             ellipse.type = "convex",
             palette = "jco",
             ggtheme = theme_minimal()
)
```



Summary of Practical Issues

- Rescale numeric data prior to k-means implementation. The scaling can be:
    - a z-transformation similar to what we did in the example
    - a 0-1 scaling
    - converting count data into percentage or counts per a certain number of the population
    - etc.
- Use more than one metric to determine k when using k-means clustering
- Your cluster solution is not the end result, you will need to:
    - visualize it in appropriate way (simple representation as in the previous slide, spatially, time-based, etc.)
    - Attempt to explain the cluster membership using an appropriate binomial/multinomial model (e.g., see this analysis)

k-means in Tableau

Let us use Tableau to implement the k-means clustering implementation on the 60 sample observations from the penguins dataset as shown in Slide 6 of this presentation.

Recap

Summary of Main Points

- Describe the different steps of the k-means algorithm
- Cluster using k-means (by hand)
- Cluster using k-means (software)
  -
  - Tableau