Player Modelling: Clustering 2

Matthew Guzdial

guzdial@ualberta.ca

Announcements

- Practice Quiz Today + Office Hours directly after
- Monday (Nov 1) watch-along video from Dr. Alex Zook, then a Data Scientist at Blizzard + Player Modeling Review
- Wednesday (Nov 3): Intro to PCG + Future of Game AI Voting
- Friday (Nov 5): Quiz 4 (no class)
- Reading Week!

Decision Tree Quick Review
Decision Tree Example (Pokémon Go)

Level	Collection Percentage	Cosmetics Unlocked	Buddy Pokémon	Steps Taken	Purchase Sale?
38	85%	55%	Eevee	100K	T
10	35%	5%	Charizard	12K	F
5	10%	1%	Bulbasaur	6K	T
40	99%	100%	Infernape	150K	F
25	45%	95%	Gardevoir	20K	T

So far: K means and K medoids Clustering

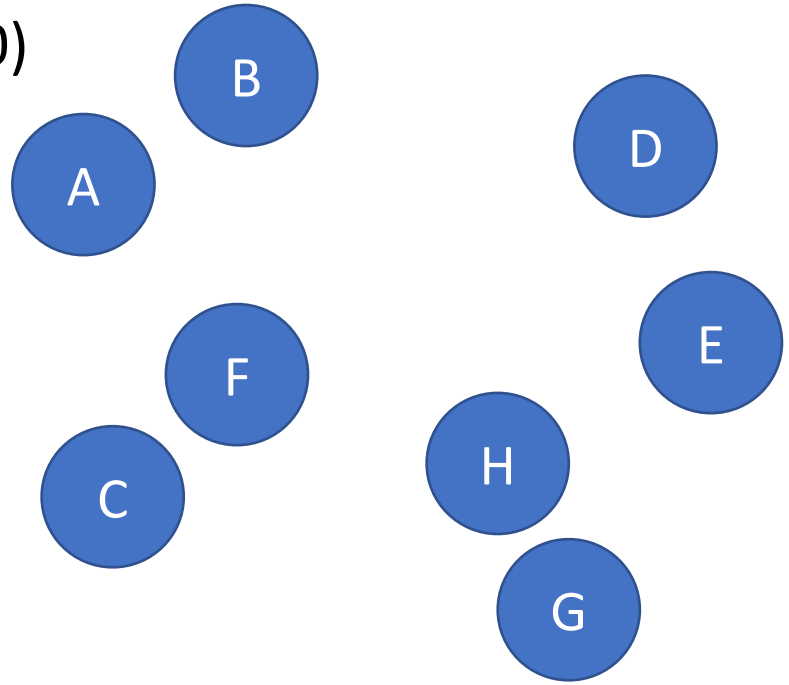
An Even Simpler Clustering Method: Single-linkage Clustering

We don’t need to find K.

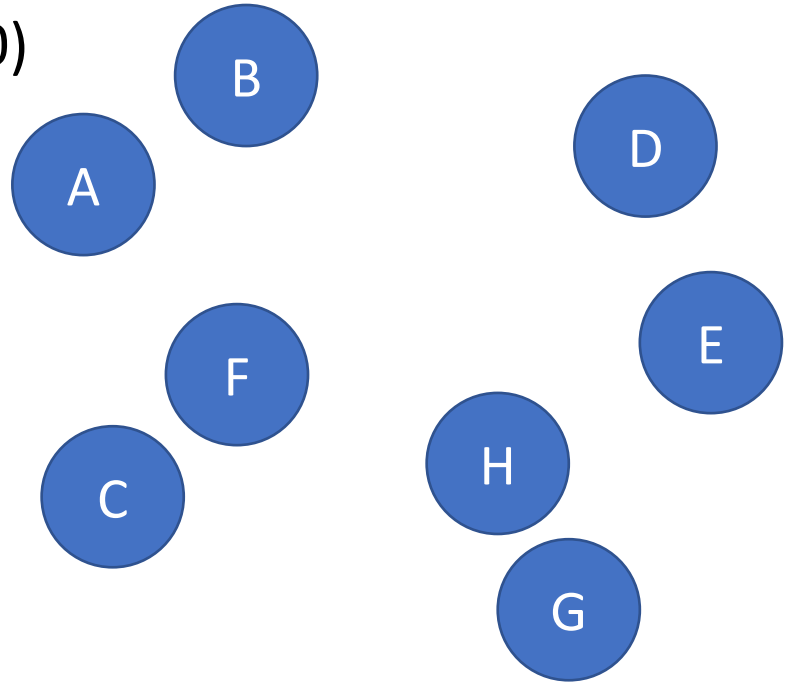
Algorithm:

- At every step, link the closest unlinked points, this builds up a tree.
- After every point is linked together, determine where the best “cut” is.

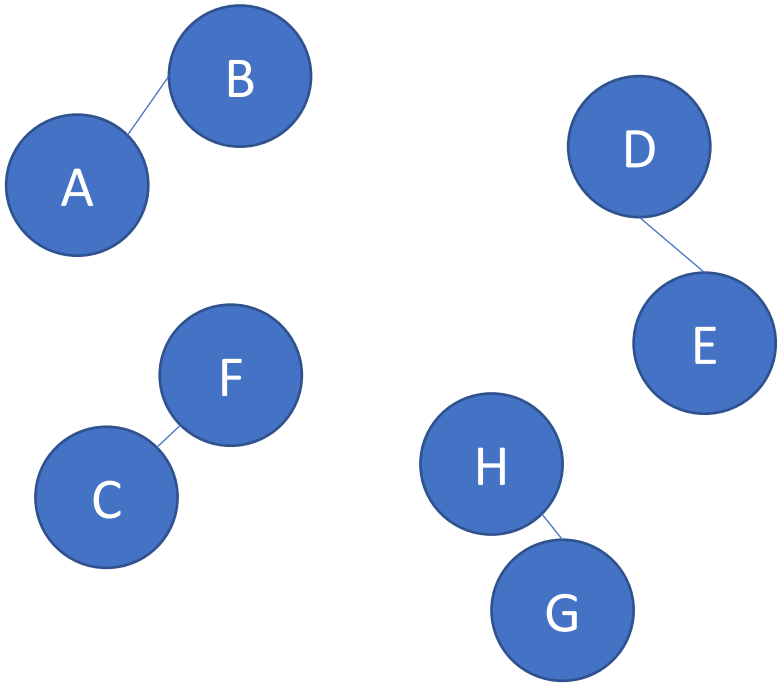
0)



0)

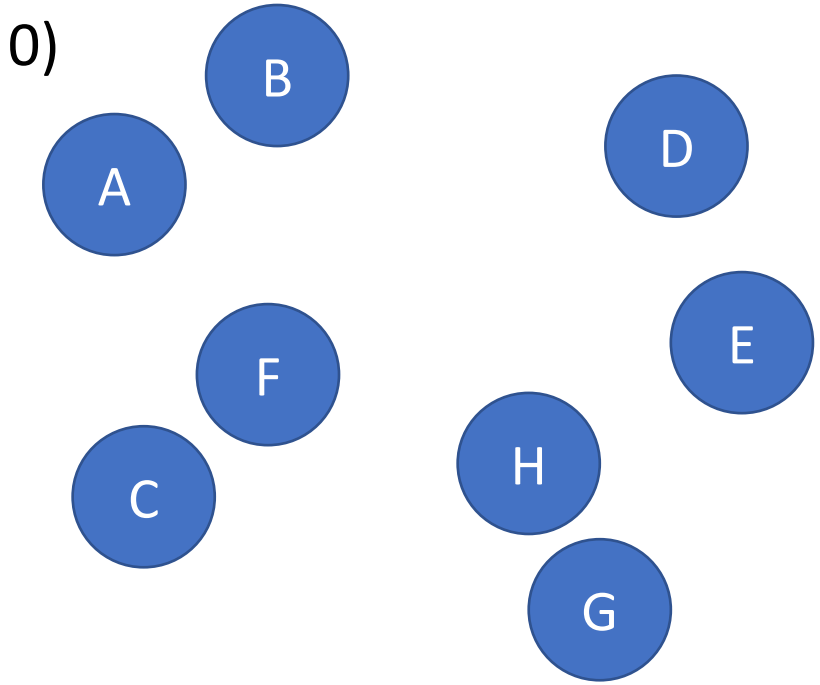


1)

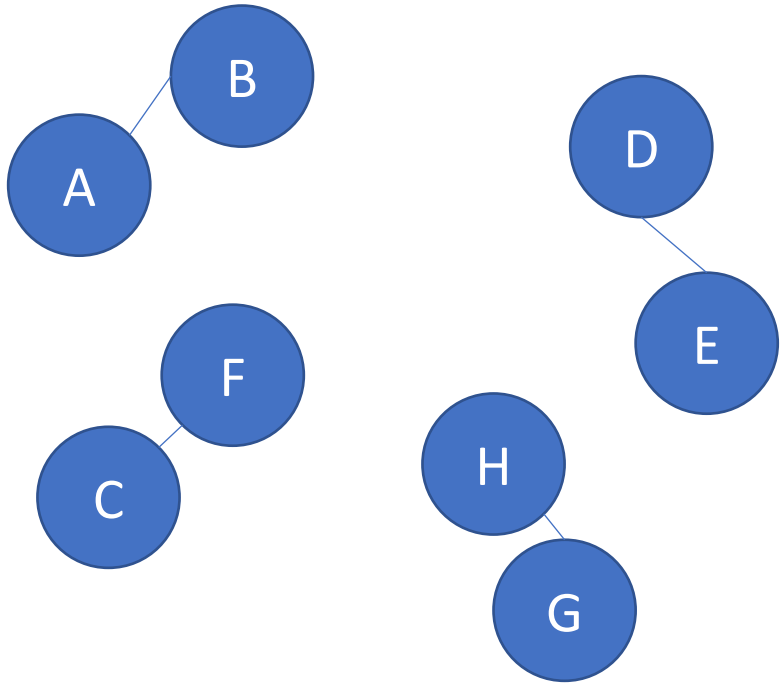


1)

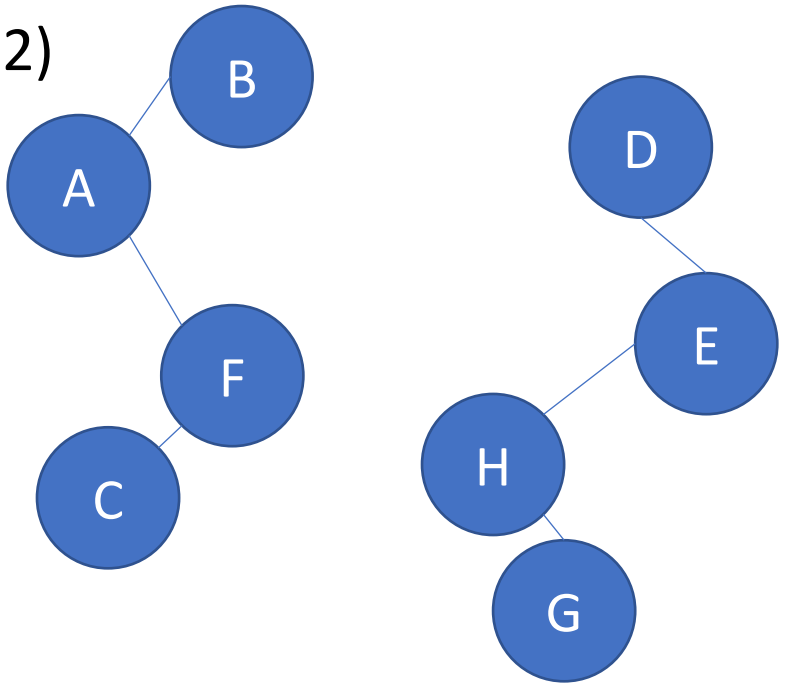
0)

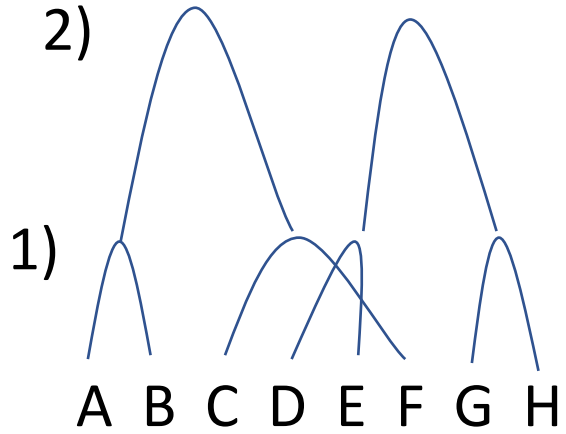


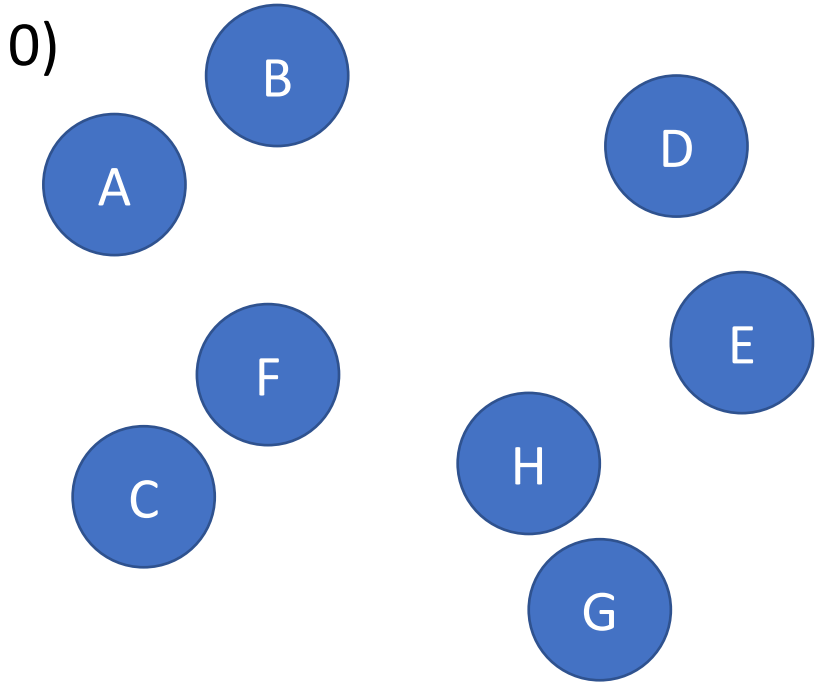
1)

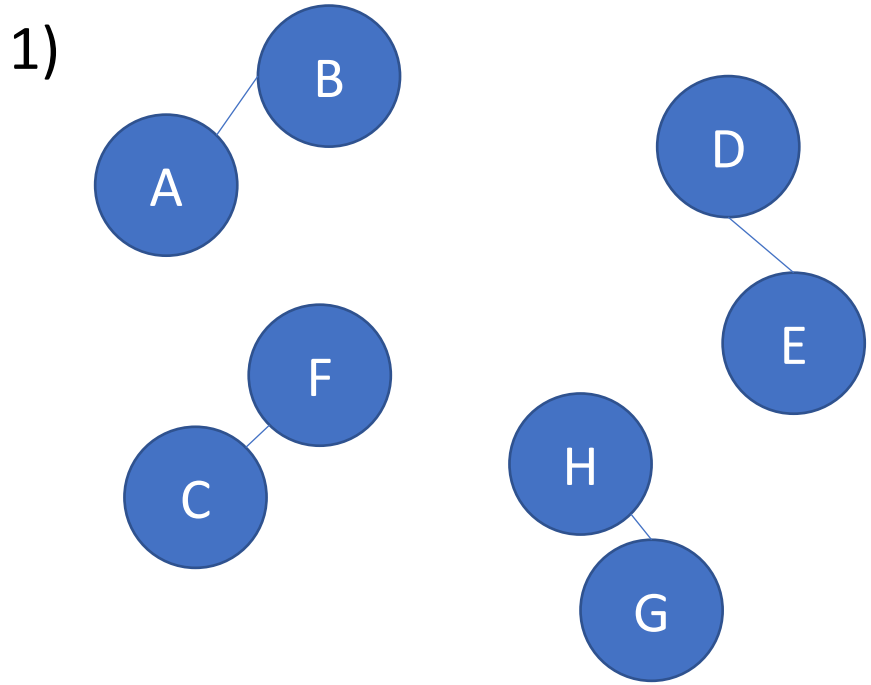


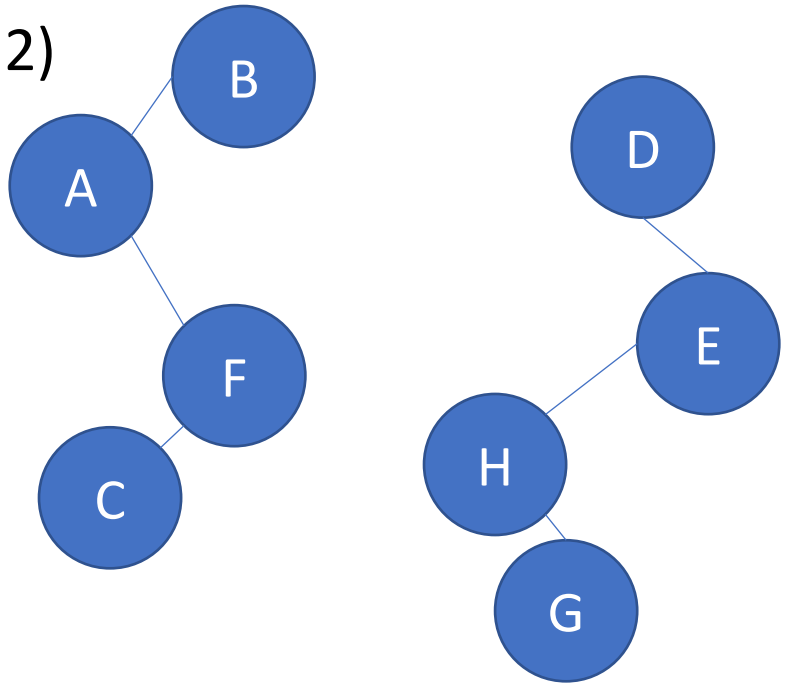
2)

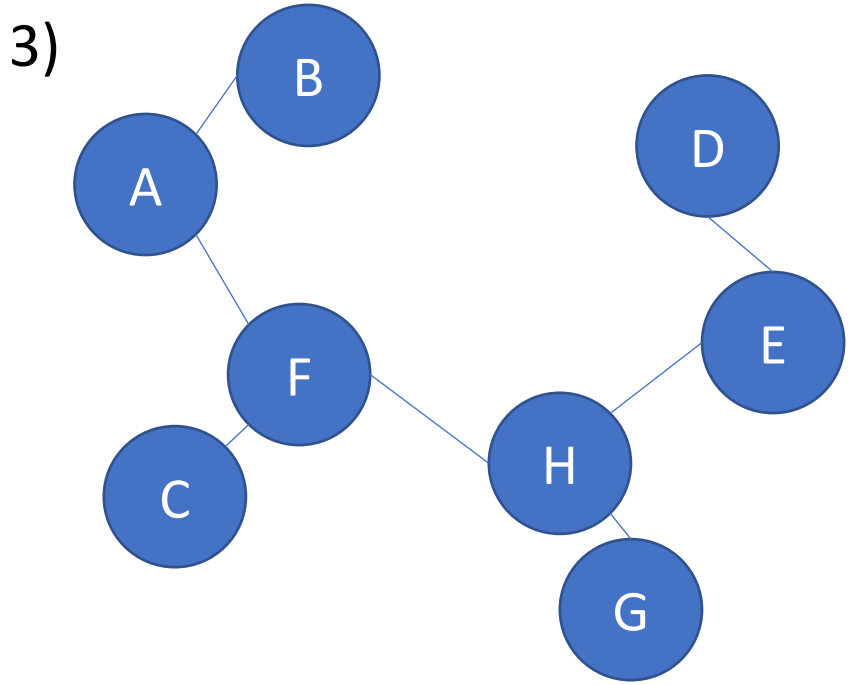


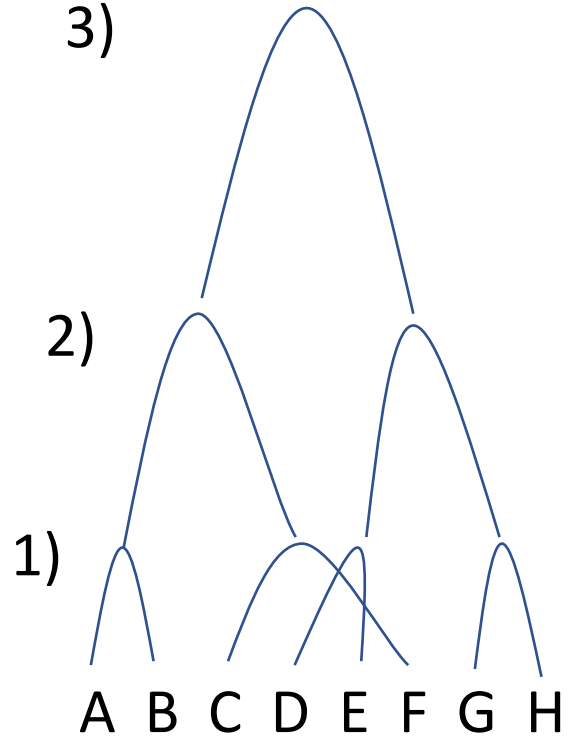



0)
B
A
F
C
D
E
H
G


1)
B
A
F
C
D
E
H
G

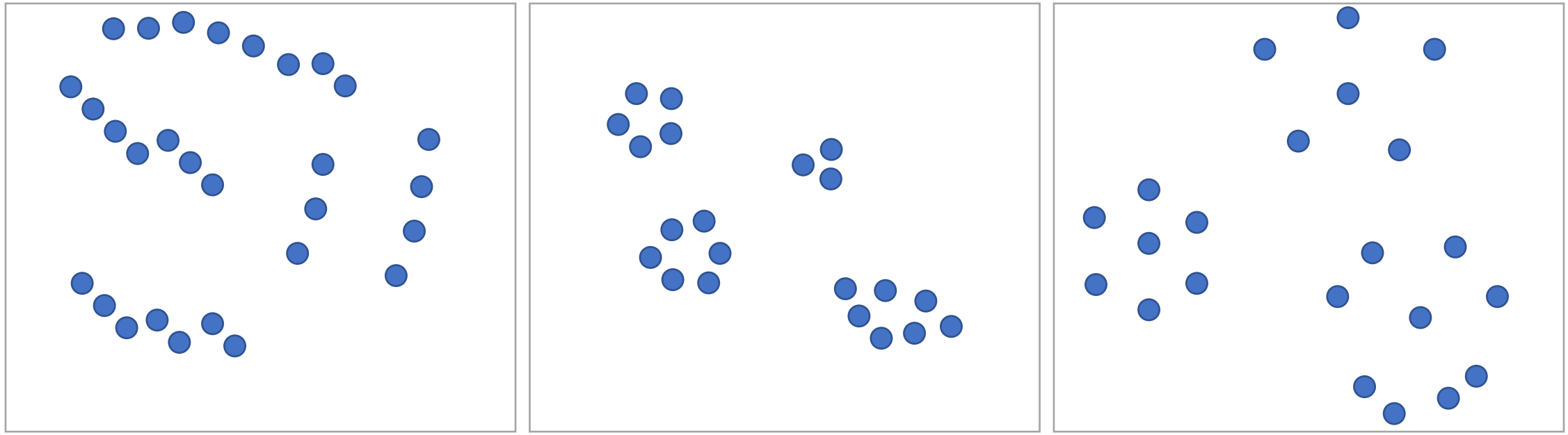

2)
B
A
F
C
D
E
H
G


3)
B
A
F
C
D
E
H
G


3)
2)
1)
A B C D E F G H

PQ1: https://forms.gle/Y9cqzgSK1xfndDPBA https://tinyurl.com/guz-pq24

What clustering approach discussed so far would you use for each of the following data sets? Why?



A B C

When to use what clustering technique?

- Single-linkage
  - When our data like lots of small line segments.
  - When we don’t know K and want to see possible clusters fast.
  - When we want a consistent algorithm.
- K means
  - When theoretical “average” points make sense and our data looks ”blobby”
- K medoids
  - When “average” points don’t make sense and our data looks “blooby” with points in the center of the blobs.

Assumption: So far, we've assumed every data point has one unique center.

Question: Can you think of a situation where this assumption doesn't hold (doesn't make sense)?

My Answer: Identifying Player Types

- It’s probably not the case that a single person only ever displays one “player type”.
- E.g. Sometimes a player might act more like a collector, more like a killer, more like an achiever, etc.

Solution: Mixture Models

For every data point (D), we will say it belongs to a particular cluster (C) with some probability (P(C|D)).

- | means “given” when talking about probabilities.
- So we can guess C given the information in data point D given some probability

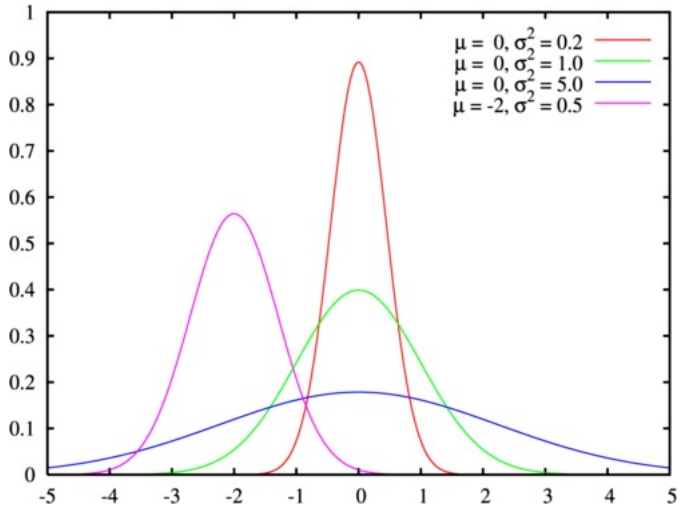
Mixture Models Method

1. Choose a probability distribution
2. Choose a K
3. Instantiate K random distributions (clusters)
4. Calculate P(C|D) for each datapoint.
5. Iterate until convergence
   - For each distribution, re-calculate its parameters given its current clustered datapoints.
   - Calculate P(C|D) for each datapoint.
6. (Optional) Run again but with a different K

Choose a Probability Distribution (There are many, I don't expect you to know them all)

https://en.wikipedia.org/wiki/Mixture_model

- Binomial distribution, for the number of "positive occurrences" (e.g., successes, yes votes, etc.) given a fixed number of total occurrences
- Multinomial distribution, similar to the binomial distribution, but for counts of multi-way occurrences (e.g., yes/no/maybe in a survey)
- Negative binomial distribution, for binomial-type observations but where the quantity of interest is the number of failures before a given number of successes occurs
- Poisson distribution, for the number of occurrences of an event in a given period of time, for an event that is characterized by a fixed rate of occurrence
- Exponential distribution, for the time before the next event occurs, for an event that is characterized by a fixed rate of occurrence
- Log-normal distribution, for positive real numbers that are assumed to grow exponentially, such as incomes or prices
- Multivariate normal distribution (aka multivariate Gaussian distribution), for vectors of correlated outcomes that are individually Gaussian-distributed
- Multivariate Student's-t distribution (aka multivariate t-distribution), for vectors of heavy-tailed correlated outcomes[1]
- A vector of Bernoulli-distributed values, corresponding, e.g., to a black-and-white image, with each value representing a pixel; see the handwriting-recognition example below

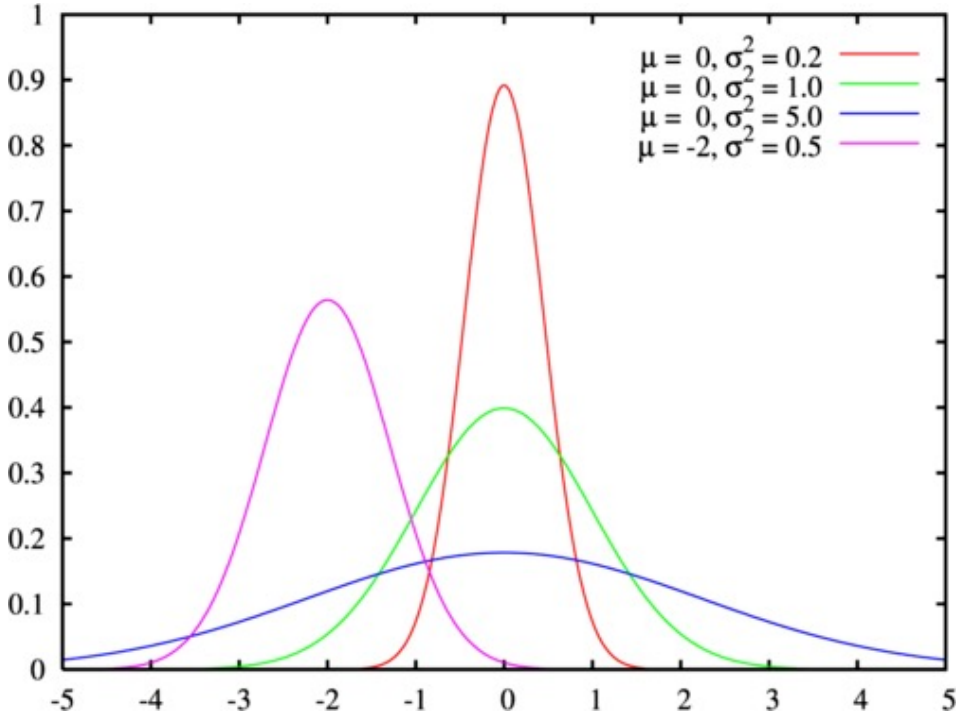


Normal Distribution Most Common
"Gaussian Mixture Models" (GMMs)

Calculate P(C|D) (Gaussian)

$$f(x) = a \cdot \exp\left(-\frac{(x-b)^2}{2c^2}\right)$$

- a: Height of the curves peak (is the top of the curve 100%, 50%, etc.?)
- b: The position of the center of the peak
- c: The width of the “bell” (what area do we cover?)



Example on Virtual Board (time permitting)

Practice Quiz 4

https://forms.gle/ZQ8ETXexyzHu8Jbu8

https://tinyurl.com/guz-quiz4

1. Which of the following takes the longest time to train?
   A.) Decision Trees B.) Linear Regression
   C.) K means D.) All about the same
2. Imagine you are clustering data from an education game meant for children. The data has the form (name, age, favorite animal, favorite colour, favorite shape). Would it be better to use K means or K medoids, and why?
3. Write five fake data points of the form
   (deaths (int), level (int), timePlayed (int), churned (T or F)).
4. Using the data from 3, assume you trained a decision tree to predict churned, what would the decision at the root be?
5. Using the data from 3, still assuming we're training a decision tree, what would the decisions at depth 1 be if there are any?

1. C. In the worst case decision tree runtime is O(m*n) where m is the number of features of each datapoint and n is the number of data points. $O(m^2n)$ for linear regression and O(n*K*I*m) for K means clustering (I is iterations). Except in the case where (K*I)<=m (which is rare), K means is the slowest.
2. K medoids. It likely doesn't make sense to treat colours, animals, and shapes as something you can average between.
3. Variable, see table for example
4. Level<=4
5. N/A

ID	Deaths	Level	Time Played	Churned
1	0	0	0	T
2	100	1	100	T
3	10	4	10	T
4	100	10	100	F
5	1	100	1000	F

Any remaining time for examples