Cluster Analysis and K-Means

MATH/CMPT 370

Amanda Landi

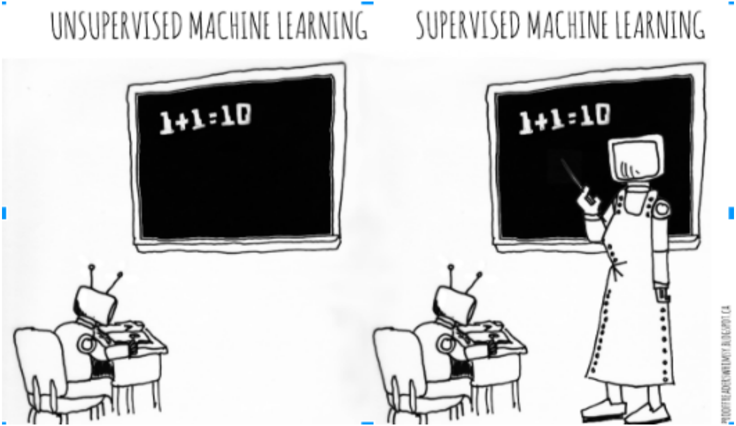
March 7, 2017

Outline

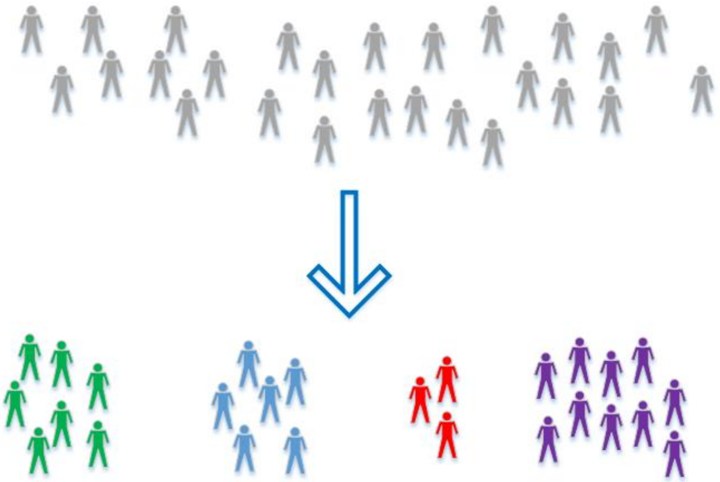
In this lesson, we discuss cluster analysis and explore the unsupervised machine learning method known as K-means.

1. Unsupervised Machine Learning?
2. Clustering?
3. Some Real Examples
4. K-Means Method via Example: Iris Dataset
5. In-class Activity
6. Analyzing Results
7. How to Choose Number of Clusters

Unsupervised Machine Learning



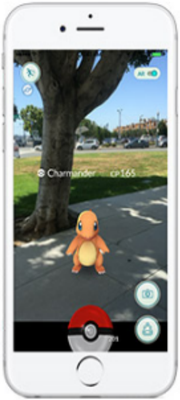
Clustering



User Retention

Consider Pokémon Go.

- Collects data on the user: gender, location, pokémon captured, user's start playing, user's last play
- Want to figure out the common features among users who continue to play versus users who play one month



Application to Genomic Data

Consider a data array - rows represent cells, columns represent time points

- Each element is a measurement of cancer cell activities after treatment by solution M
- Want to find common patterns among cells over time
- Can inform doctors of effectiveness of solution M and, perhaps, determine on which types of cells

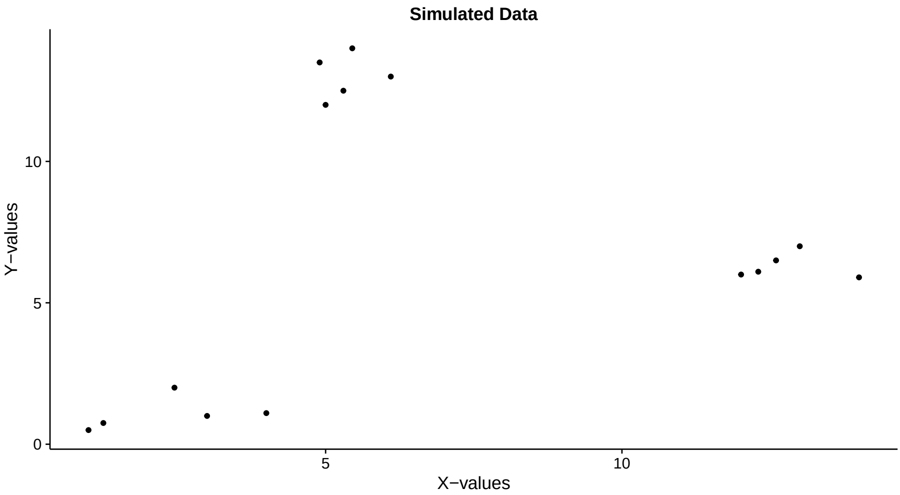
Recommendation Engines

Group users based on similarities

- Movie Recommendations: ratings on already-viewed movies, genres of already-viewed movies
- Advertisements: products purchased, products viewed
- Other features?

Clustering Example

We will consider first simulated data where clusters are clearly defined.



K-Means, The Problem

Given a set of observations $\{x_1, x_2, \dots, x_n\}$, where each observation is an m-dimensional real vector, k-means clustering aims to partition the n observations into k sets $P = \{P_1, P_2, \dots, P_k\}$ so as to minimize the Euclidean distance within clusters. In other words, we want to find

$$\arg\min_{P} \sum_{i=1}^{k} \sum_{x \in P_i} ||x - \mu_i||_2^2$$

where μ_i is the mean of the points in P_i.

- The idea has been around since the late 1950s
- The standard method wasn't published until 1982
- Lloyd's algorithm converges in $O(nkmi)$, where i is the number of iterations.

K-Means, Lloyd's Method

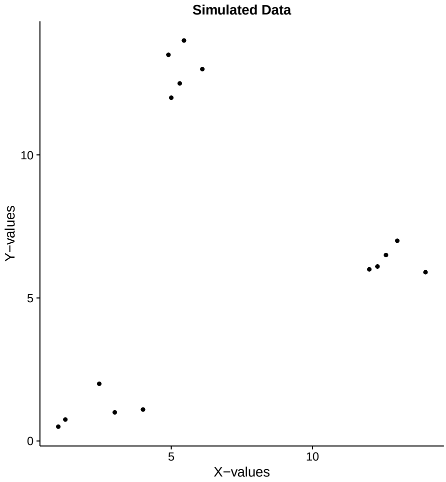
Outline for Code

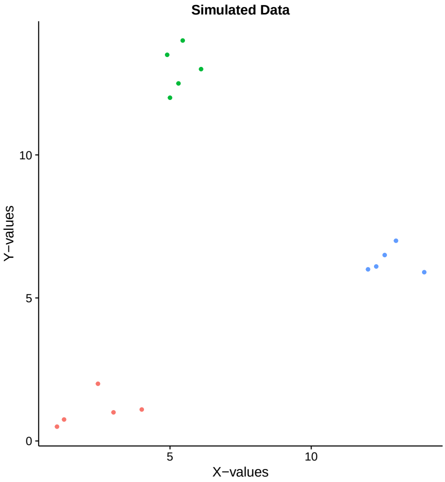
1. Parameters: data, k, maxiter
2. Initialize the k centers
3. Create a vector for cluster assignments
4. Then, within our maxiters
    - we loop over each data point
        - calculate the distance of each point from each of the centers
        - assign cluster (mean) number to data points where distances are smallest
    - update the means

Note:

- Algorithm has "converged" when assignments no longer change
- There is no guarantee a global optimum is found. So, when to stop?
- Many different implementations
    - Forgy method initializes means by choosing from the original data

Clustering Example Results

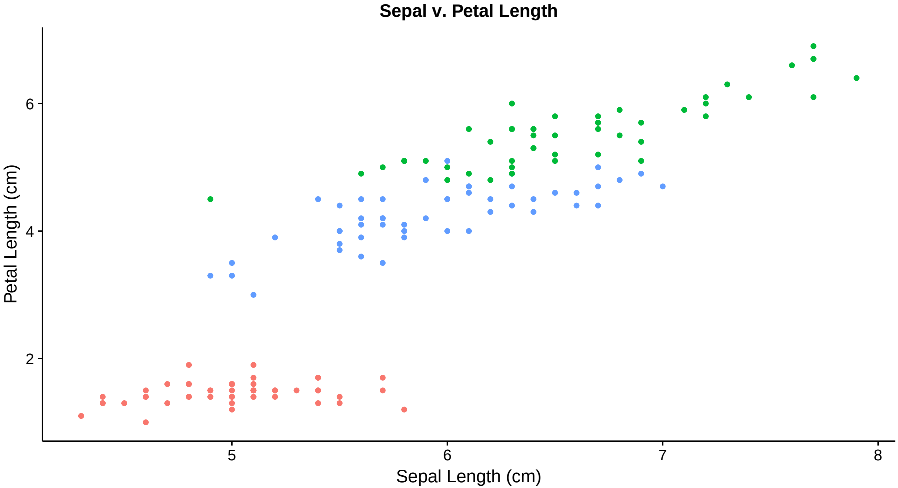




Note, K-means run 10 iterations.

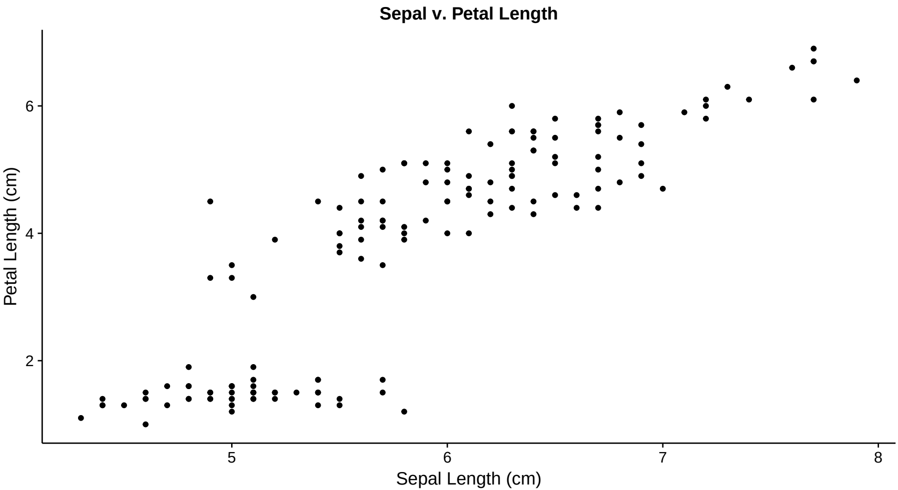
Iris Example

Recall the built-in Iris dataset.

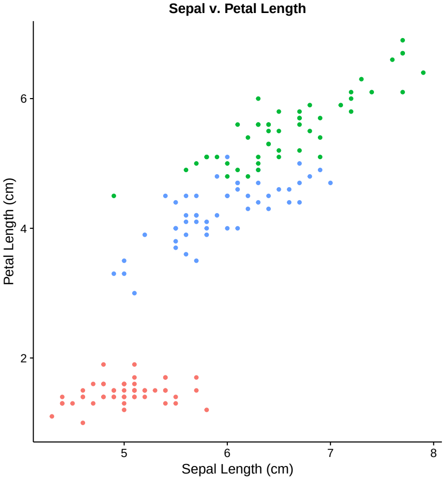


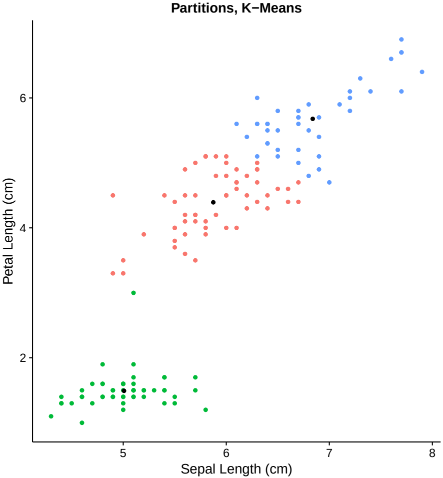
Iris Example

We know the truth, but let's pretend we don't. We want to cluster (or group) individual data points based on similarity.



K-Means Results - Iris Dataset





Note, K-means run 10 iterations.

In-class Activity (20 - 25 minutes)

- Using the description of K-Means, attempt to implement your own K-means.
- For a guide on how to begin, please look at the outlined code on Git

`https://src-code.simons-rock.edu/git/MATH_CMPT_370_S17/K_means`

- Practice your data visualization – Others need to be able to understand your results.

What you don't finish in class, please do for homework.

Analyzing Results

- If the method has found the appropriate partitions, then we expect there to be less variation **within** groups and more variation **between** groups.
- Before defining our measures for within and between, we define the total sum of squares, where μ is the mean of all data points, to be

$$TSS = \sum_{i=1}^{n} (x_i - \mu)^2,$$

- To measure the variance within groups, we examine the within sum of squares

$$WSS = \sum_{i=1}^{k} \sum_{x \in P_k} (x - \mu_j)^2$$

Analyzing Results

- To measure the variance between groups, we examine the between sum of squares

$$BSS = TSS - WSS$$

- We want the ratio $\frac{BSS}{TSS}$ to be large (close to 1).
- For our iris results, we have
  - `TSS = 566.493733333333`
  - `WSS Cluster 1 20.4078048780488, WSS Cluster 2 23.5084482758621, WSS Cluster 3 9.89372549019607`
  - `BSS/TSS = 0.905012226123878`
- What else might we care about when it comes to K-means clustering?

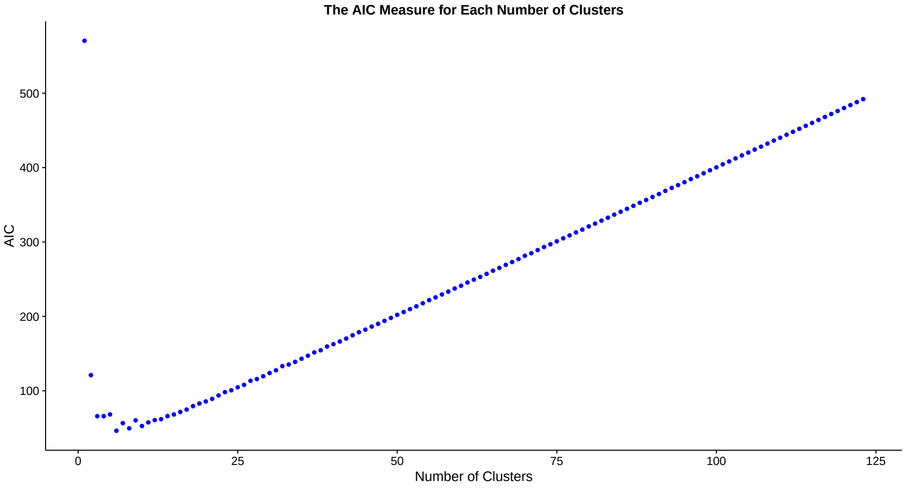
How to Choose Appropriate k

- Typically chosen *a priori*.
- Can compare against the cost function.
- Information Criterion, AIC - Akaike Information Criterion:
  - Used in model selection; trade off between goodness of fit of model and complexity of model
  - For K-means,

$$AIC = 2C + WSS_C$$

  - where C is the amount of complexity (for K-means, $C =$ number of features per data point x number of clusters)
  - WSS_c is the within sum of squares measure at complexity C
  - Notice as k gets closer to n, the variation within each cluster will decrease; first term will overpower the second term.
  - As k is smaller, variation within a group will be larger; second term will overpower the first.
  - Need to calculate AIC for all $k = 1 : n$
  - The point at which AIC is minimal gives an estimate for the number of clusters.

Graph AIC



For our experiment, $k = 6$. Very close to reality!

References

- Raschka, S. **Python Machine Learning**. 2015
- image on slide 3:
  `http://www.frankichamaki.com/`
  `data-driven-market-segmentation-more-effective-marketing-to-seg`
- AIC on slide 16:
  - `http://sherrytowers.com/2013/10/24/k-means-clustering/`
  - `http://nlp.stanford.edu/IR-book/html/htmledition/cluster-cardinality-in-k-means-1.html`

The Mean as 2-Norm Squared Minimizer

We look at the optimization problem

$$\min_y J(y) = \min_y \sum_{i=1}^{n} (x_i - y)^2$$

To minimize we find the derivative with respect to y

$$\frac{dJ}{dy} = -2(x_1 - y) - 2(x_2 - y) - \ \ldots \ - 2(x_n - y)$$

We set this equal to 0 and can immediately cancel out the -2 from every term. Notice, then

$$\begin{aligned}
(x_1 + x_2 + x_3 + \cdots + x_n) - n \cdot y = 0 \\
(x_1 + x_2 + x_3 + \cdots + x_n) = n \cdot y \\
(x_1 + x_2 + x_3 + \cdots + x_n)/n = y
\end{aligned}$$

The AIC for K-Means

- AIC = 2C - 2ln(L)
- C = m x k, m = number of features per individual, k = number of clusters in model
- L is the likelihood the data can be obtained from the model
- We assume data has Gaussian distribution where each cluster has its own mean with standard deviation 1, and each cluster is independent of one another

$$
\begin{aligned}
p(y|M) &= \Pi_{j=1}^{k} p(y|\mu_j, \sigma_j = 1) \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^k e^{-\frac{1}{2}\sum_{j=1}^{k}\sum_{i=1}^{t}|x_i - \mu_j|^2} \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^k e^{-\frac{1}{2}WSS}
\end{aligned}
$$

- Thus, the natural log gives $-\dfrac{1}{2}WSS$ plus some constant.