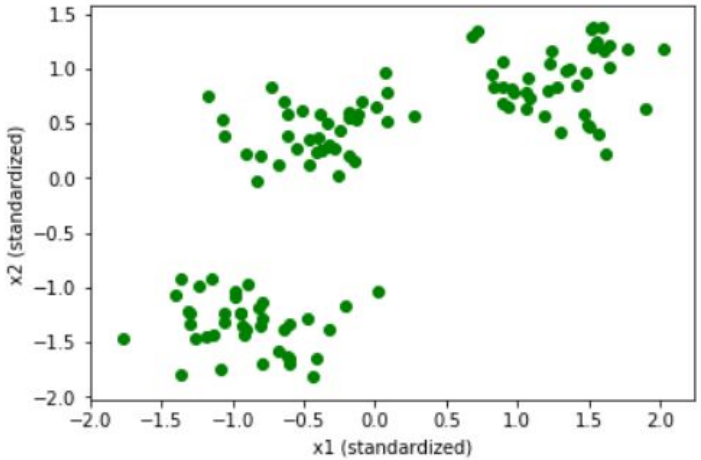
THINKFUL

Centroid-based Algorithms

DATA SCIENCE

Warm Up

Consider the following visualization of a toy dataset:



How many clusters intuitively jump out? If you had to characterize each of these clusters by a single point, what would that point be?

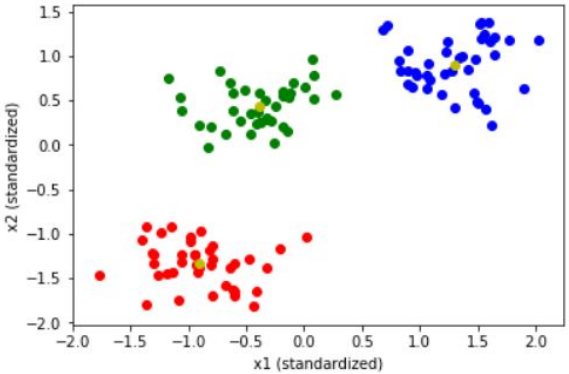
Agenda

- Overview of k-means clustering
- Step-by-step walkthrough of the algorithm
- Advantages and disadvantages

Overview

K-Means clustering is one of the simplest, most popular methods for clustering quantitative data

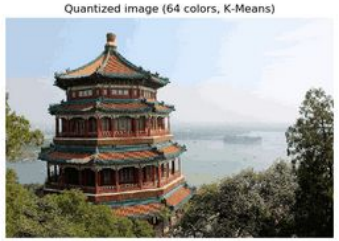
- The “K” in “K-Means” refers to the number of clusters, which must be specified in advance
- The “Means” in “K-Means” refers to the fact that each cluster is characterized by a single point called a *centroid*, which is derived by taking the mean of the points assigned to that cluster
- Objective function: **minimize the within-cluster variance** (equivalently, the sum-of-squared distances from observations to corresponding centroids)

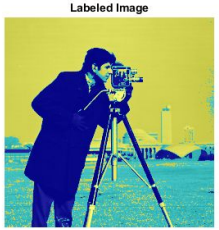


Use Cases

Common use cases for k-means clustering include:

- Data compression
  - Observations are represented by their corresponding centroids
  - Reducing an image to a fixed number of colors
  - Image segmentation

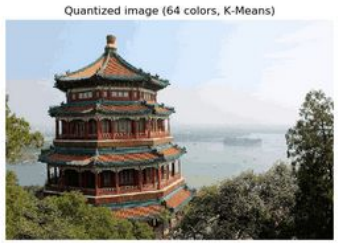


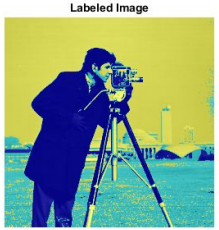


Use Cases

Common use cases for k-means clustering include:

- Segmentation
  - Identifying groups of common observations
  - E.g. products, customers, etc.
  - Geospatial data
  - For example, identifying common pickup spots for rideshares





UP NEXT

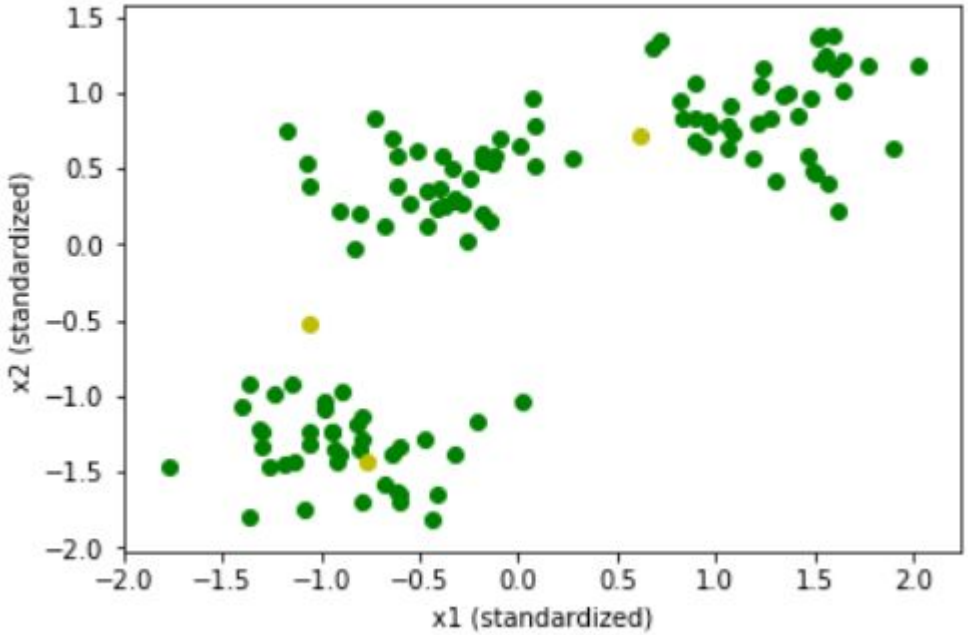
Algorithm Walkthrough

Algorithm Summary

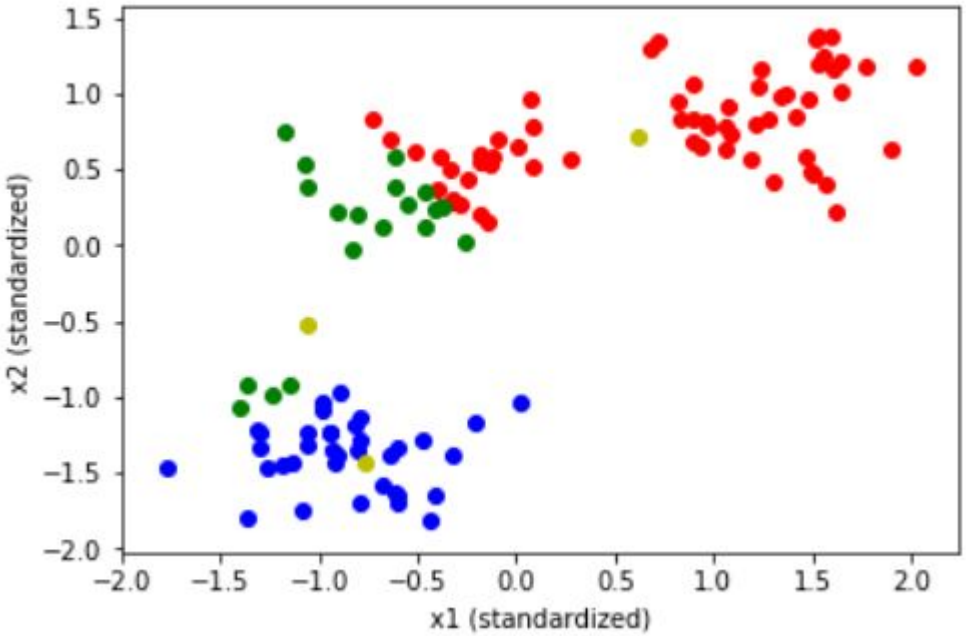
The simplest and most popular implementation of k-means, **Lloyd's algorithm,** clustering involves an iterative approach that alternates between two steps (after choosing an initial set of k centroids):

- **Assignment:** Assign each point to a cluster. For a specific point, this assignment is done by calculating the Euclidean distance from said point to each the k centroids, then assigning the point to the cluster with the nearest centroid
- **Update:** Recalculate the centroid for each cluster by taking the means of the points in the cluster.
- This process is repeated until convergence (until the assignments and centroids no longer change)
- Lloyd's algorithm is an approximation/heuristic
    - Finding the exact optimal solution is very computationally intensive and infeasible for any dataset of reasonable size

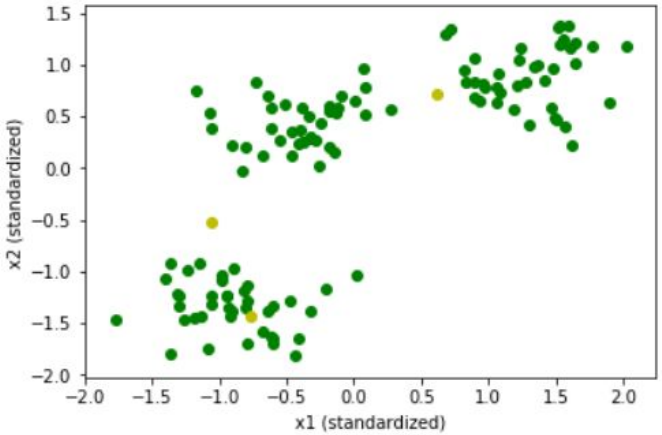
K-Means: Initial State



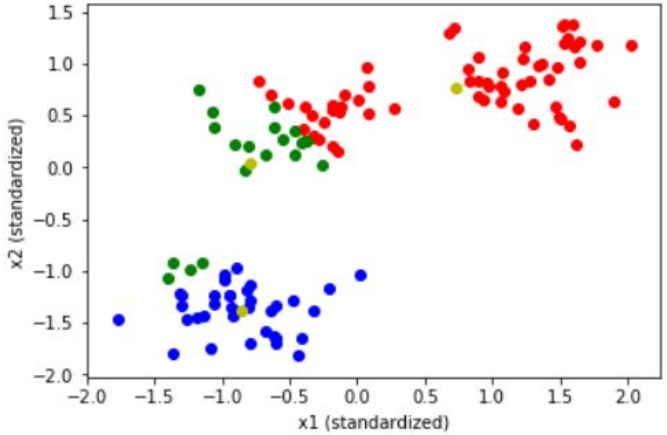
K-Means: Assignment Step 1



K-Means: Update Step 1

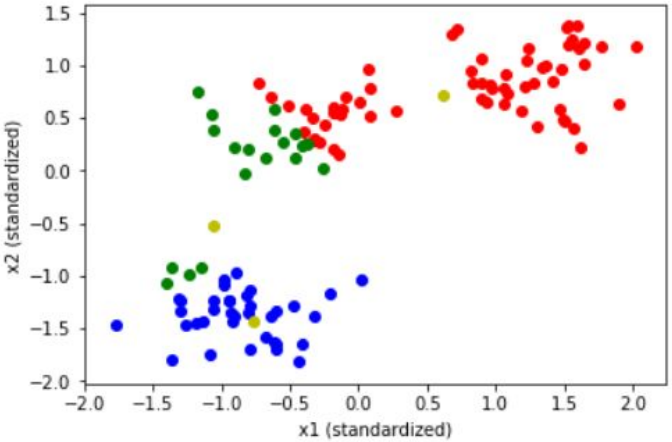


Initial centroids

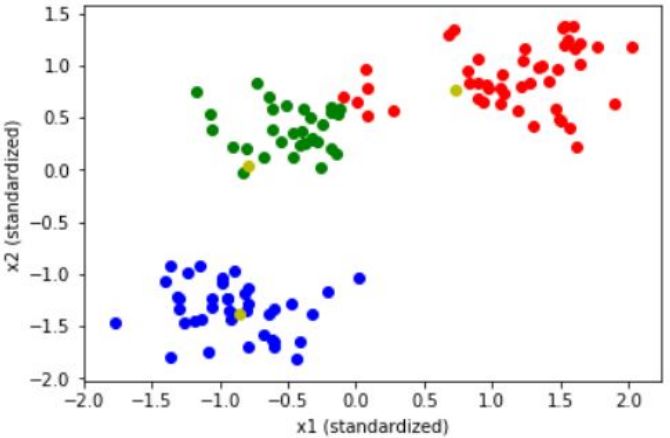


New centroids after
first assignment step

K-Means: Assignment Step 2

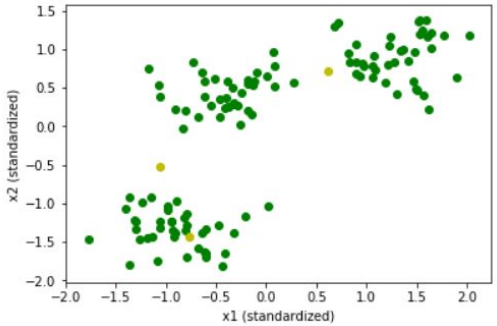


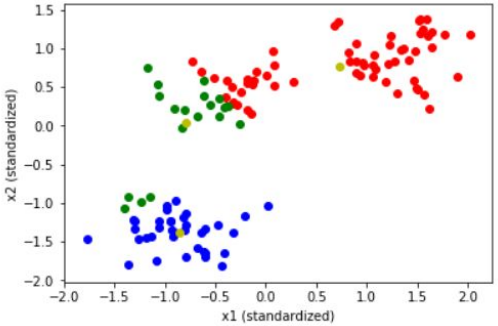
Initial assignments

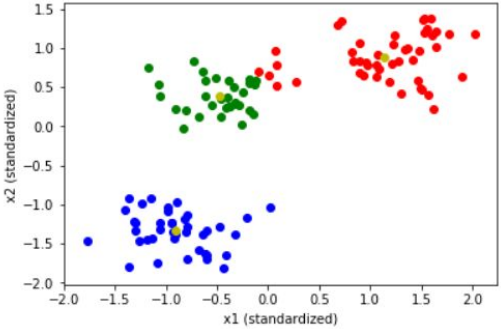


New assignments after first centroid update

K-Means: Update Step 2







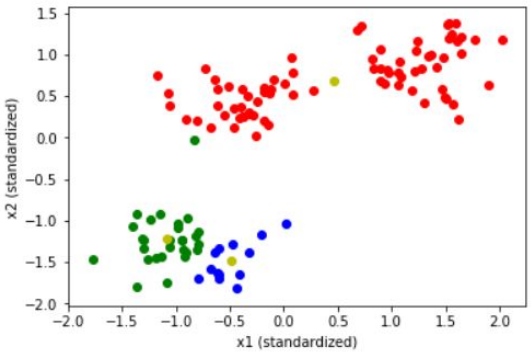
Initialization Techniques

K-means is sensitive to the choice of initial centroids; it can converge to a local minimum rather than the global minimum depending on the initial centroids. Various techniques exist for initialization; three of the most popular are:

- **Random:** uniform random assignment of each point to a cluster, followed by normal mean centroid calculation
- **Forgy:** select k random data points from dataset as initial centroids
- **Kmeans++**: select the first centroid at random from the dataset, then select subsequent centroids from the dataset using a weighted probability distribution. The probability a point is chosen is proportional to the square of its distance from the nearest previously chosen centroid

Initialization Techniques

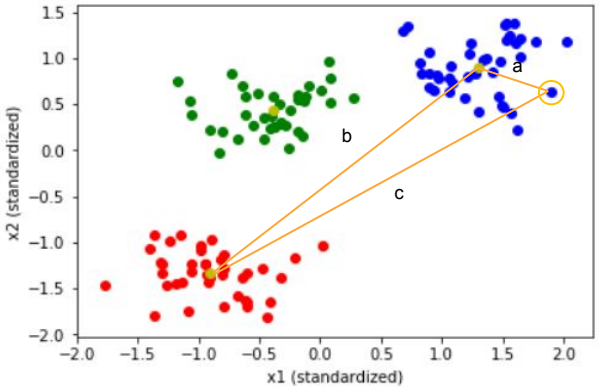
```
k_means = KMeans(n_clusters=3, random_state=123, n_init=1, init='random')
k_means.fit(all_data)
y_pred = k_means.predict(all_data)


# Plot the solution.
colors = 'rbg'
for i in range(all_data.shape[0]):
   plt.scatter(all_data[i,0], all_data[i,1], c=colors[y_pred[i]])
for k in range(k_means.cluster_centers_.shape[0]):
   plt.scatter(k_means.cluster_centers_[k,0], k_means.cluster_centers_[k,1], c='y')
plt.xlabel('x1 (standardized)')
plt.ylabel('x2 (standardized)')
plt.show()
```



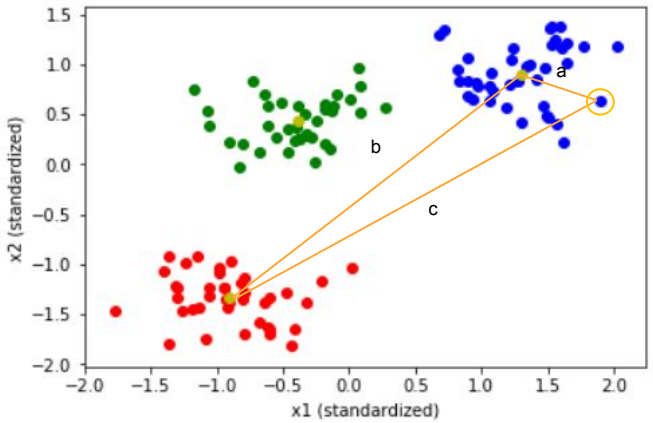
Beyond Naive K-Means

The previously discussed algorithm for k-means clustering can be inefficient, as it requires recomputing all of the observation-centroid distances at every iteration. Two of the most widely implemented improvements on Naive k-means are:

- **Elkan's algorithm**
  - Intuition: if the centroids move very little after an update step, most of the observation-centroid distances do not need to be recalculated
  - Uses the triangle inequality to eliminate many unnecessary computations



Beyond Naive K-Means



After a centroid update, one of our goals for the circled point is to determine if c > a. The naïve approach would be to calculate both a and c, but what if we could do better?

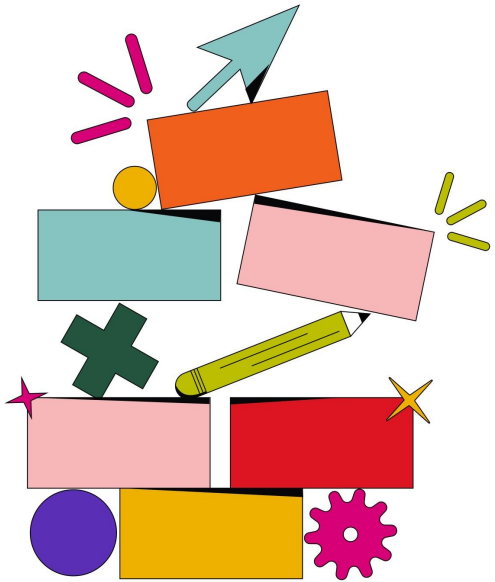
The triangle inequality states that |a-b| ≤ c ≤ a+b. Therefore, if b > 2a, c must be larger than a, so there is no need to consider reassigning the circled point to centroid corresponding to the red cluster.

By keeping track of the distances between centroids (b in the example above), we can eliminate several unnecessary distance calculations.

- **Mini-batch k-means**
  - Instead of using the entire dataset, uses randomly selected samples for each iteration
  - Updates to centroids are done on a per-sample basis in a manner that mimics gradient descent

UP NEXT

Advantages & Disadvantages



Advantages

K-Means is one of the most popular clustering algorithms for several reasons:

- Simplicity
  - Easy to implement
  - Easy to interpret clusters by examining centroids
  - Easy to understand algorithm and explain to others
- Efficiency/Scalability
  - Runtime of naive algorithm scales linearly with size of dataset (on average)
  - Improvements on naive algorithm (e.g. mini-batch) provide significant additional improvements to algorithm runtime

Disadvantages

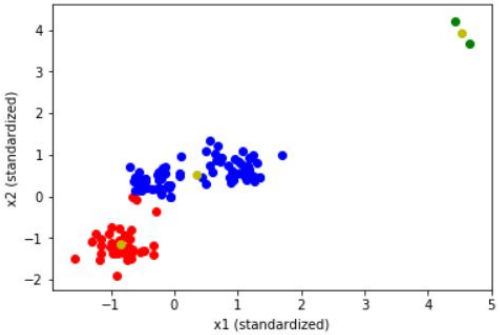
However, k-means also has several drawbacks

- User must specify k in advance
  - No “one true approach” to choosing k; most approaches are a combination of heuristics and domain knowledge
- Sensitivity to outliers
  - Outliers can drag the centroid away from natural clusters, resulting in suboptimal assignments
- Not suitable for categorical data
  - Objective function is grounded in minimizing variance, which only makes sense in the context of quantitative data

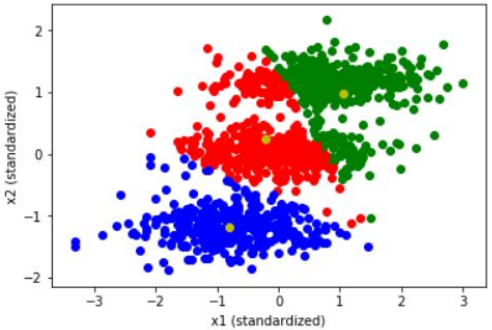
Disadvantages

- Strong assumptions about data and clusters
  - Isotropically (spherically) distributed data → bias towards spherical clusters
  - Equal variance (i.e. spheres of equal radius)

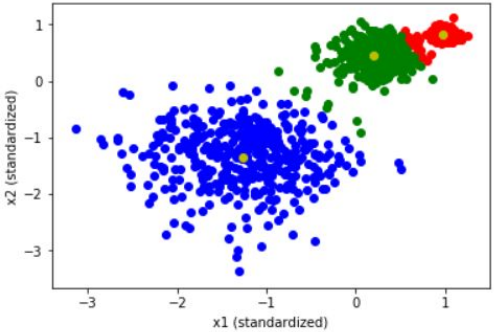
Disadvantages



Outliers



Non-spherical
(anisotropic)



Spherical but
unequal variance

Summary

- K-means clustering is a very popular centroid-based unsupervised learning algorithm
- The simplest implementation of algorithm alternates between an assignment step (assigning observations to centroids) and an update step (recalculating centroids based on newly assigned observations)
- K-means is fast, scalable, and easy to explain; however, it requires knowing the number of clusters in advance and makes several strong assumptions that are frequently violated in reality

EXERCISE

1. Jupyter notebook
2. Data set: NBA player-seasons from 15-16 to 18-19
3. Filter players that do not play very much (i.e. few games started, few minutes player per game)
4. Isolate a subset of columns that would be useful for identifying player archetypes
5. Run K-Means clustering and interpret the results

Thank You