k-Means Clustering

Rina Singh

Tennessee Tech. University

April 12, 2017

Outline

1. Clustering
2. *k*-Means Clustering
   - Lloyd's Algorithm
   - *k*-means++
3. Choosing *k*
   - Elbow Method
   - Gap Statistic

Clustering

Clustering is an unsupervised machine learning task that attempts to assign observations into groups (i.e., clusters) so that observations in the same cluster are similar in some sense.

k-Means Clustering

Given a set of observations (represented as points in $\mathbb{R}^n$) and a positive integer k, the k-means clustering task asks to partition the observations into k disjoint clusters in such a way as to minimize the within-cluster sums of squares (WCSS).

Objective Function

More formally, *k*-means clustering ask to find

$$\underset{\boldsymbol{C}=\{C_1,\ldots,C_k\}}{\arg\min} \sum_{l=1}^{k} \sum_{\boldsymbol{x}_i,\boldsymbol{x}_j \in C_l} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \underset{\boldsymbol{C}=\{C_1,\ldots,C_k\}}{\arg\min} \sum_{l=1}^{k} |C_l| \sum_{\boldsymbol{x} \in C_l} \|\boldsymbol{x} - \boldsymbol{\mu}_l\|^2$$

where

$$\boldsymbol{\mu}_l = \frac{1}{|C_l|} \sum_{\boldsymbol{x}_l \in C_l} \boldsymbol{x}_l$$

denotes the centroid of the cluster C_l.

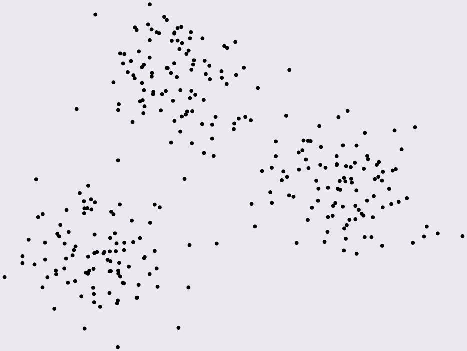
Complexity

k-means clustering is an *NP*-Hard problem, and it is often solved using heuristic or approximation algorithms. One such example is Lloyd's algorithm. Lloyd's algorithm is so commonly used to perform *k*-means clustering that it is often referred to as the *k*-means algorithm.

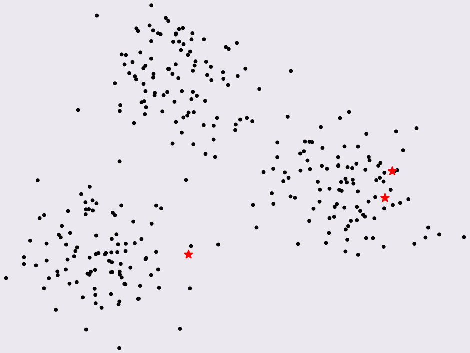
Lloyd's Algorithm

1. "Randomly" pick c initial seeds.
2. Calculate the distance between each point and the seeds.
3. Assign each point to the closest seed.
4. Move the seeds to the centroid of their associated points.
5. Repeat steps 2-4 until seeds no longer move.

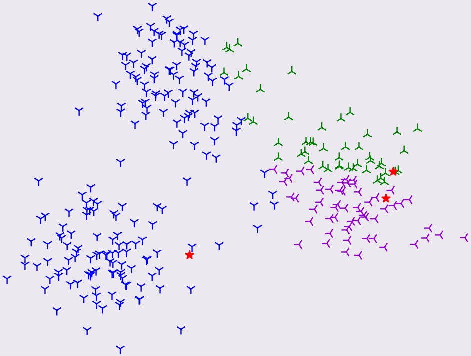
Initial data for *k*-means clustering.



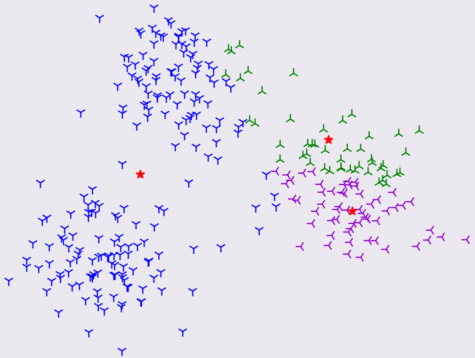
Pick initial seeds for *k*-means clustering.



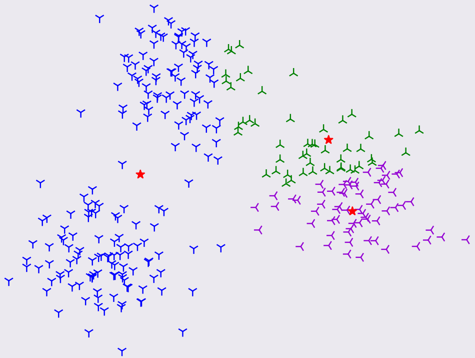
Move the seeds to the centroid of their associated points.



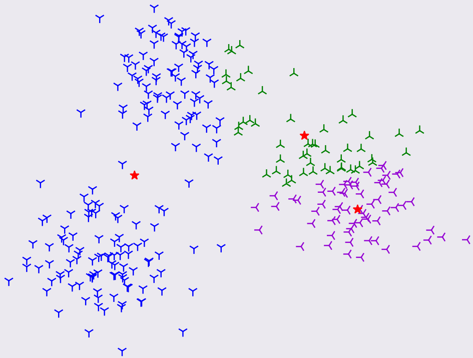
Move the seeds to the centroid of their associated points.



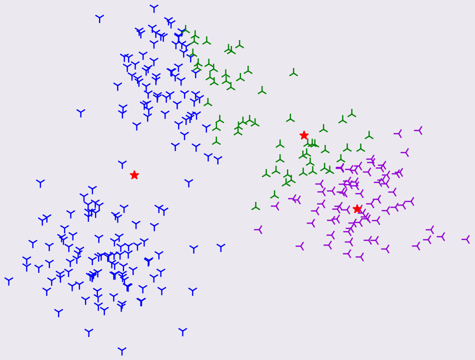
Recalculate point associations.



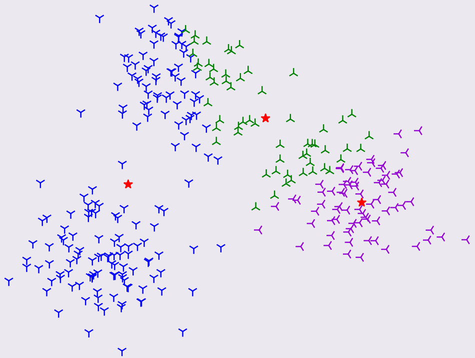
Move the seeds to the centroid of their associated points.



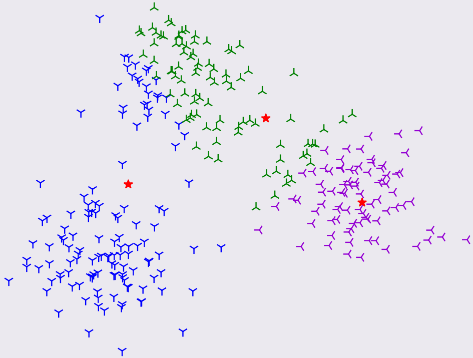
Recalculate point associations.



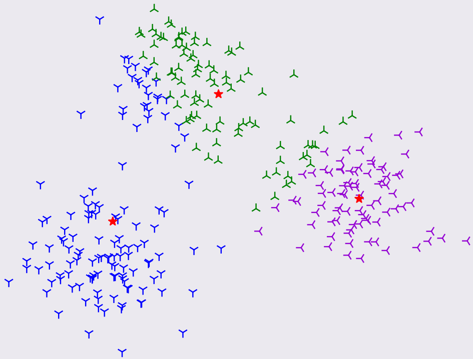
Move the seeds to the centroid of their associated points.



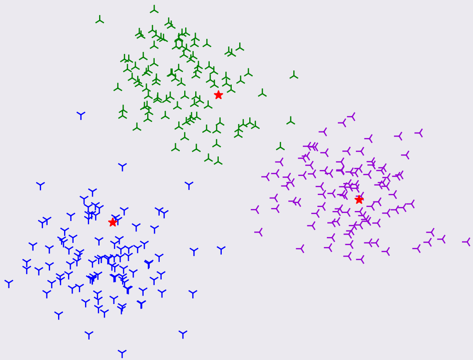
Recalculate point associations.



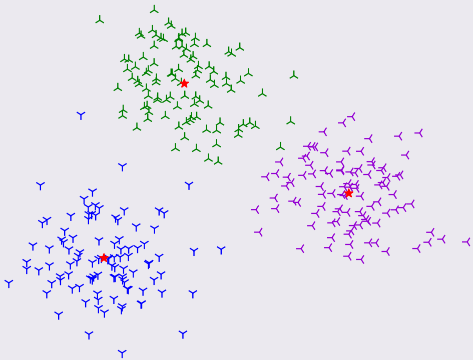
Move the seeds to the centroid of their associated points.



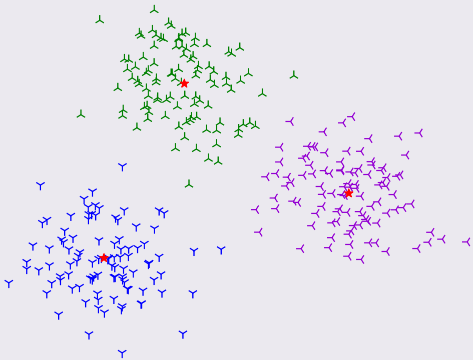
Recalculate point associations.



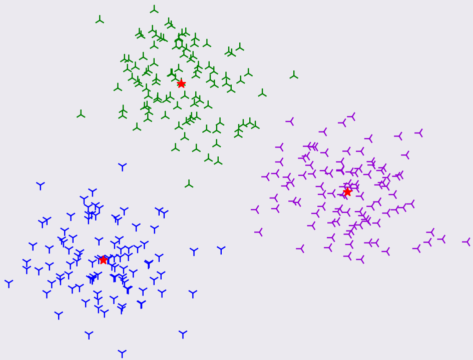
Move the seeds to the centroid of their associated points.



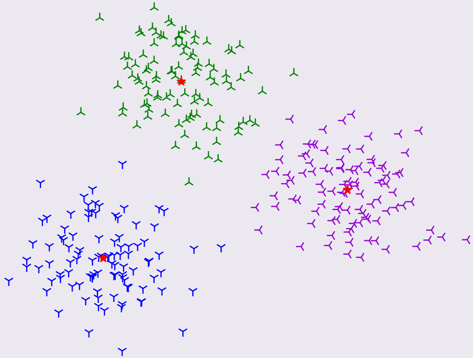
Recalculate point associations.



Move the seeds to the centroid of their associated points.



Stop when the seeds converge.



Animation

http://tech.nitoyon.com/en/blog/2013/11/07/k-means/

Lloyd's Algorithm

- Within-cluster sum of squares will decrease with each iteration of Lloyd's algorithm.
- Lloyd's algorithm is guaranteed to converge within k^n steps, where n is the number of observations.
- Convergence does not imply an optimal solution. Final clusters depend upon the initial seed positions.
- Lloyd's algorithm is often run several times in an attempt to find the clusters with the smallest within-cluster sum of squares.

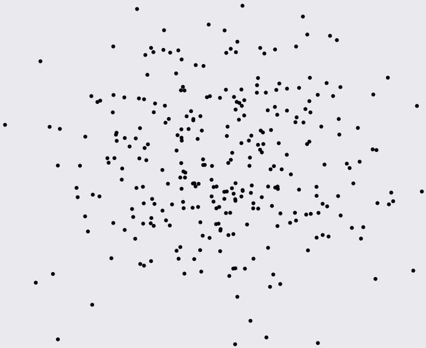
k-means++

Lloyd's algorithm can result in arbitrarily bad clusters, with respect to the objective function, when compared to the optimal solution. k-means++ is an algorithm that attempts to overcome this issue by trying to choose "good" initial seed values for Lloyd's algorithm.

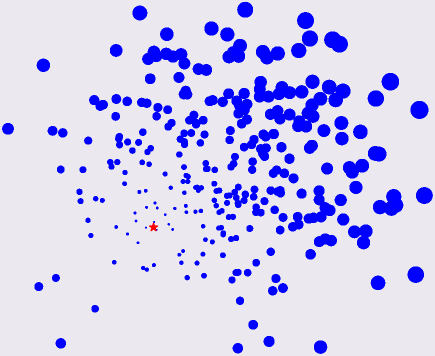
k-means++ Algorithm

1. Choose an initial seed uniformly from among the observations.
2. Calculate the distance between each observation and its nearest seed.
3. Choose a new seed from the observations with probability proportional to the distances calculated in step 2.
4. Repeat steps 2 and 3 until *k* seeds have been chosen.

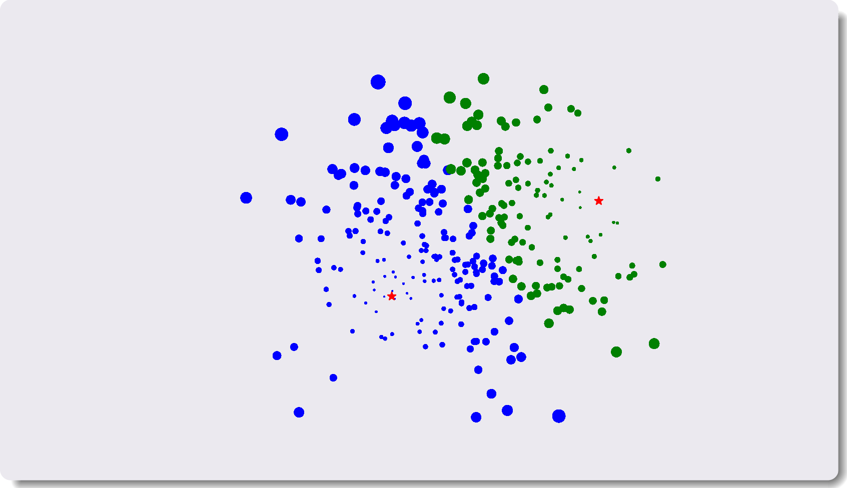
k-means++



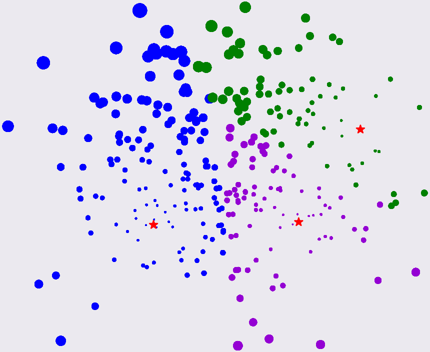
k-means++



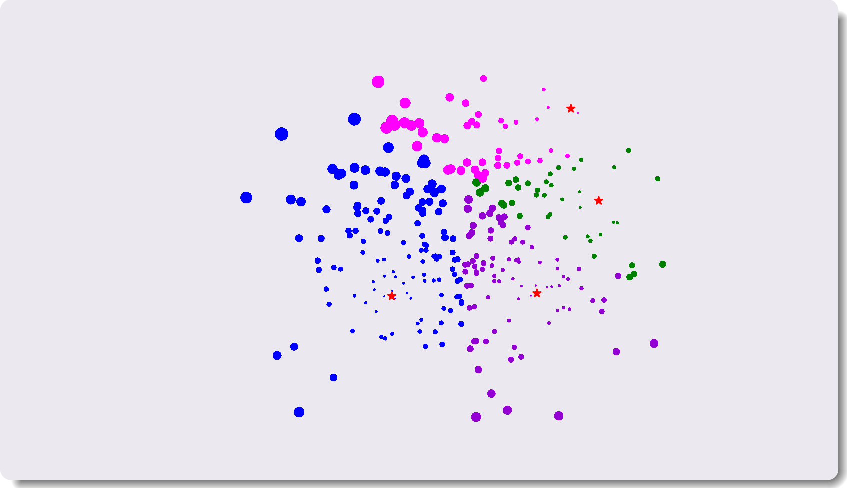
k-means++



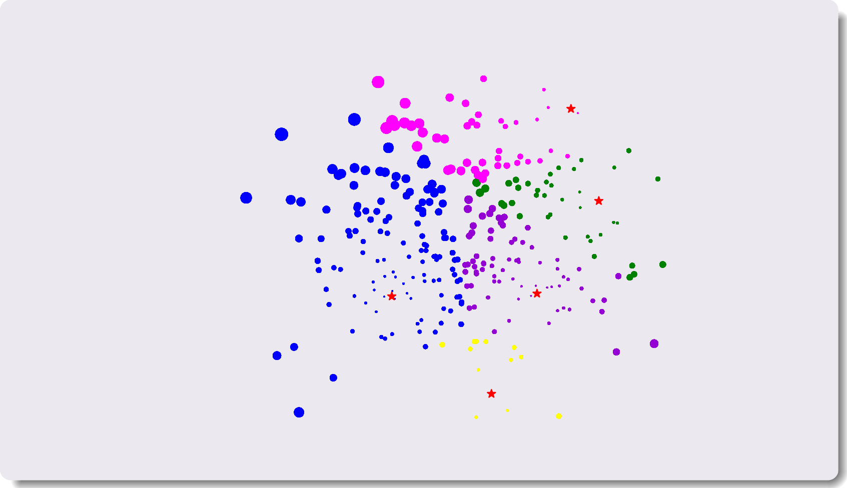
k-means++



k-means++



k-means++



Choosing k

k-means clustering assumes you know the number of desired clusters k. Knowing the value of k ahead of time can be difficult. Several methods have been developed to automatically determine k.

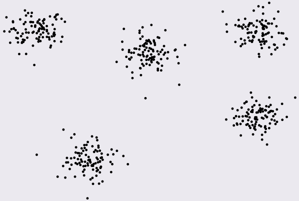
The Elbow Method

As the number of clusters increases, better models of the data are obtained, but at diminishing return. The elbow method attempts to identify the optimal number of clusters in light of this diminish return.

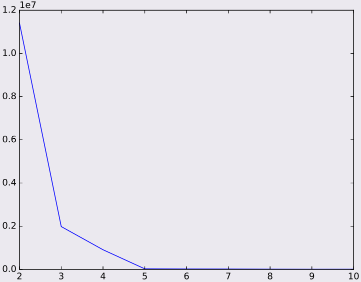
The Elbow Method

1. Run the k-means algorithm on the data for a range of k values (e.g., for $k = 1 \ldots 10$).
2. Calculate the within-cluster sums for each k.
3. Plot the within-cluster sums values for each k.
4. The graph of the within-cluster sums values will look like an arm with an elbow.

Data

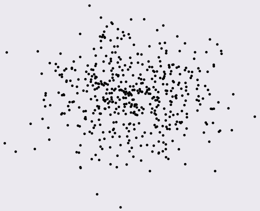


WCSS

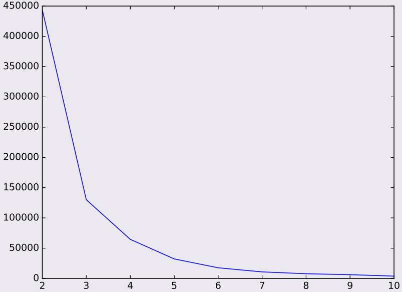


Elbow Method

Data



WCSS



Gap Statistic

The elbow can not always be unambiguously identified. In cases where there are no natural clusters within the data, the elbow may not be very pronounced. In other cases where there exists more than one natural clusters within the data, there can be multiple elbows.

Gap Statistic

The gap statistic was developed as a way to overcome the problems involved with using the elbow method. Consider the within-cluster sums of squares when there are no natural clusters within the data. This value should be noticeably different than the within-cluster sums of squares obtained when natural clusters are present in the data. The number of clusters k that result in the biggest difference should be the natural number of clusters in the data.

Gap Statistic

Recall that the within-cluster sums of squares is given by

$$\sum_{l=1}^{k} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in C_l} \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2$$

Let

$$D_l = \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in C_l} \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2$$

denote the within-cluster sums of squares for a single cluster C_l. Then, the normalized within-cluster sums of squares is given by

$$W_k = \sum_{l=1}^{k} \frac{1}{|C_k|} D_k$$

Gap Statistic

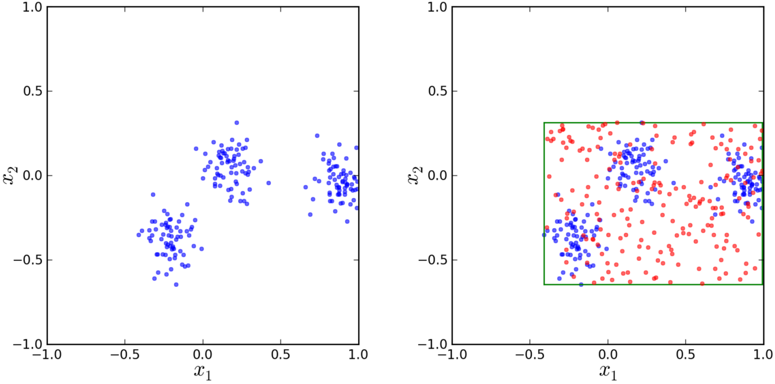
If W_k^* is the within-cluster sums of squares obtained when there are no natural clusters, then the natural number of clusters should be the k that maximizes $\text{Gap}(k) = \log W_k^* - \log W_k$. The problem is that W_k^* is not known and must be estimated.

Gap Statistic

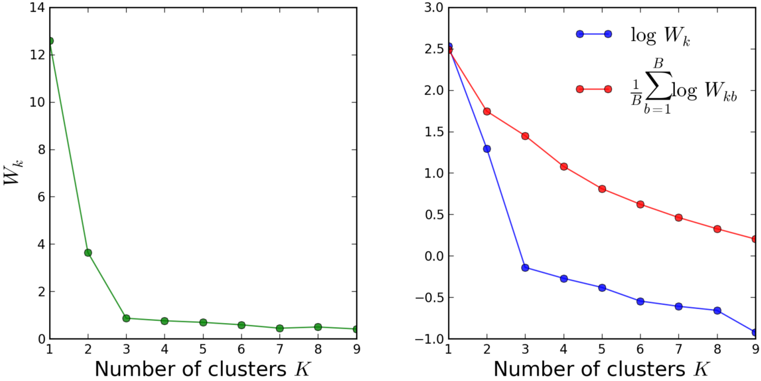
To estimate the value of W_k^*, begin by selecting uniformly random points within the bounding box of the original data and calculate W_k. Repeat this process a total of B times to obtain B different W_k values. Our estimate of W_k^* is the average of these B simulatoins.

Gap Statistic

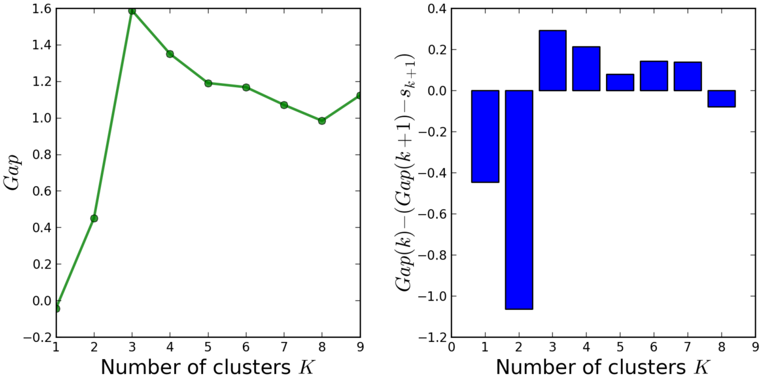
N=200, K=3



Gap Statistic



Gap Statistic



Questions?