Clustering

The k-Means Algorithm

Cary Goltermann

Galvanize

2017

Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing k

Supervised

- Have a target / label that we model.
- Models look like functions that take in data and create prediction.
- Have an error metric that we can use to compare models.

Supervised vs. Unsupervised Learning

Supervised

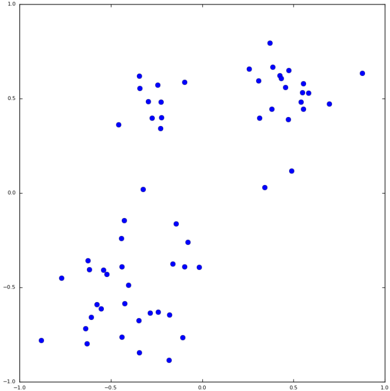
- Have a target / label that we model.
- Models look like functions that take in data and create prediction.
- Have an error metric that we can use to compare models.

Unsupervised

- No labels $\rightarrow$ no target!
- No stark error metric to compare models with.
- It's easy to be wrong, but it's hard to prove you're right.
- Trying to uncover/ **discover hidden structure** in our data.

Overview

What Is a Cluster?



- How many clusters do you see?
- What makes something a cluster?
- What makes something not a cluster?

Overview

Supervised vs. Unsupervised Learning

Clustering
- Intuition
- Definition

k-Means Algorithm
- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing k

Defining "Cluster"

- A partition of the dataset - not necessarily crisp.
- A strong internal similarity - small intra/within cluster distance.
- A strong external dissimilarity - large extra cluster distance.

Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing *k*

k-Means

The algorithm in all it's glory:

1. Initialize centroids.
2. While stopping condition not met:
   1. Find closest centroid to each point.
   2. Move centroids to the average of all the points closest to them.

k-Means

The algorithm in all it's glory:

1. Initialize centroids.
2. While stopping condition not met:
    1. Find closest centroid to each point.
    2. Move centroids to the average of all the points closest to them.

This training algorithm may look pretty simple...

k-Means

The algorithm in all it's glory:

1. Initialize centroids.
2. While stopping condition not met:
    1. Find closest centroid to each point.
    2. Move centroids to the average of all the points closest to them.

This training algorithm may look pretty simple... and that's because it is.

Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing *k*

Centroid Initialization

- The simplest way to do this is to randomly choose k points from your data and make their locations your initial centroid locations.

Centroid Initialization

- The simplest way to do this is to randomly choose k points from your data and make their locations your initial centroid locations.
- Another straightforward method is to randomly assign each data point a number 1-k, and start the initialize the k^{th} centroid to the average of the points with the k^{th} label (in each dimension).

k-Means++

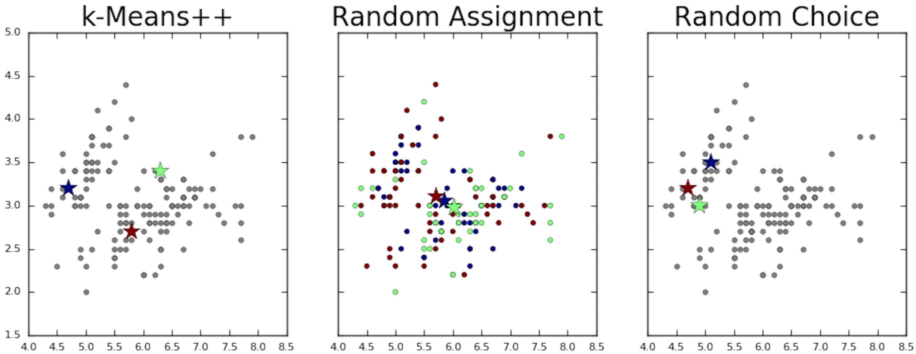
A more advanced centroid initialization method, known as k-Means++, chooses well spread initial centroids.

→ sklearn: init='k-means++', set as default.

k-Means++ follows the procedure:

1. Choose the first centroid to be the location of a data point chosen at random.
2. For each remaining centroid, choose the location of a data point with probability proportional to its squared distance from the point's closest existing centroid (points further from existing centroids have higher probability of being chosen as the next centroid).

Initialization - Visual Comparison



More even spread to start with.

All start close to the center.

Who the eff knows... could be anything!

Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing *k*

Stopping Criteria

We can update...

- for a pre-specified number of iterations.
  $\rightarrow$ sklearn: *max_iter*=1000.

Stopping Criteria

We can update...

- for a pre-specified number of iterations.
  $\rightarrow$ sklearn: *max_iter*=1000.
- until the centroids don't change at all - may take a ton of iterations.

Stopping Criteria

We can update...

- for a pre-specified number of iterations.
  $\rightarrow$ sklearn: *max_iter*=1000.
- until the centroids don't change at all - may take a ton of iterations.
- until the centroids don't move very much - takes fewer iterations.
  $\rightarrow$ sklearn: *tol*=0.0001, for tolerance of "how much".

Overview

Supervised vs. Unsupervised Learning

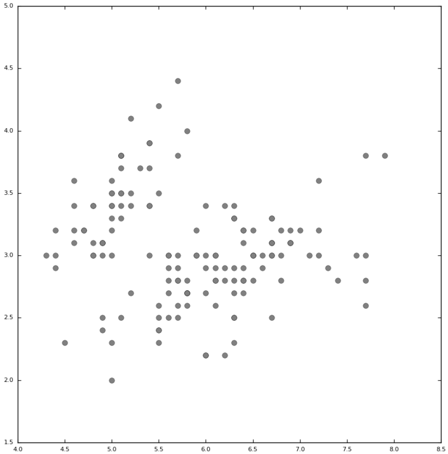
Clustering

- Intuition
- Definition

k-Means Algorithm

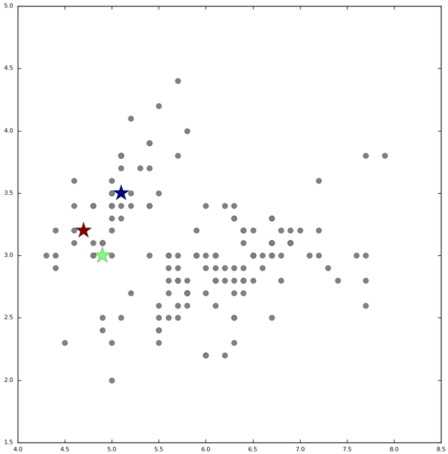
- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing k

Step-by-step Execution: DATA!!



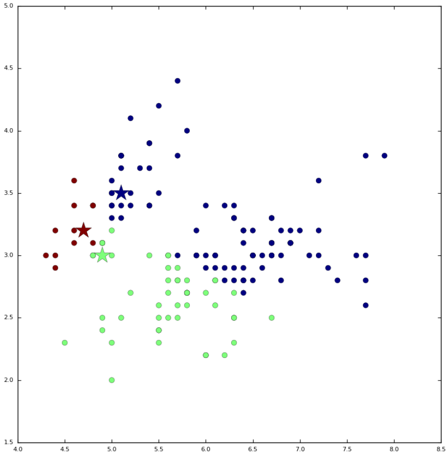
Step-by-step Execution: Initialize

1. Initialize centroids. ⟸
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location



Step-by-step Execution: Iteration 1 - Step 1

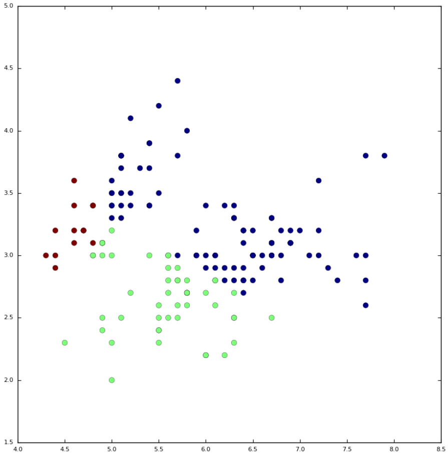
1. Initialize centroids.
2. While not stopping condition:
   1. Assign points to centroid ⟸
   2. Move centroids to new average location



Step-by-step Execution: Iteration 1 - Prep Step 2

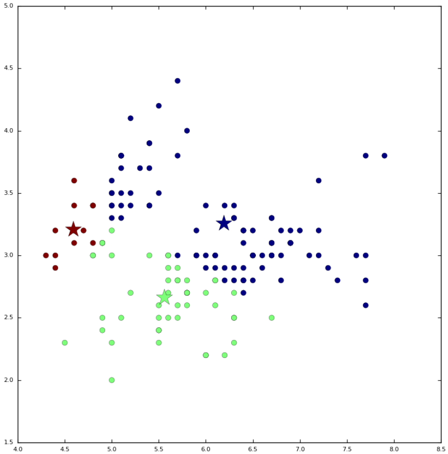
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location

$\Longleftarrow$



Step-by-step Execution: Iteration 1 - Step 2

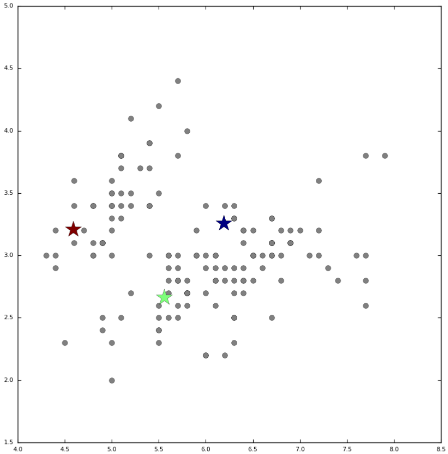
1. Initialize centroids.
2. While not stopping condition:
   1. Assign points to centroid
   2. Move centroids to new average location $\Longleftarrow$



Step-by-step Execution: Iteration 2 - Prep Step 1

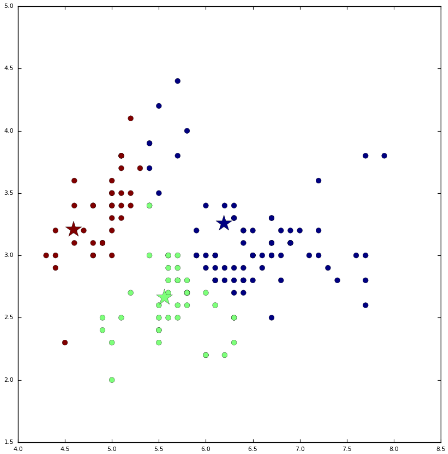
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location

$\Longleftarrow$



Step-by-step Execution: Iteration 2 - Step 1

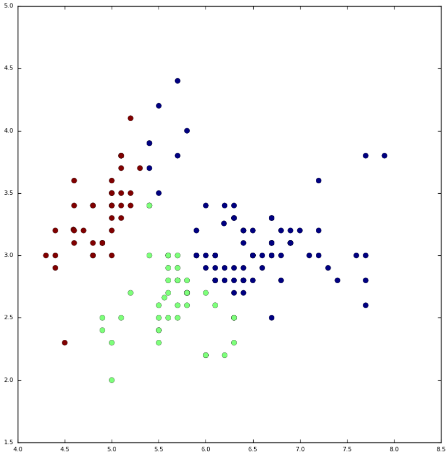
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid $\Longleftarrow$
    2. Move centroids to new average location



Step-by-step Execution: Iteration 2 - Prep Step 2

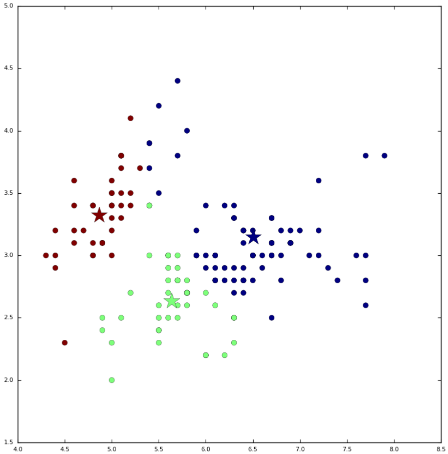
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location

$\Longleftarrow$



Step-by-step Execution: Iteration 2 - Step 2

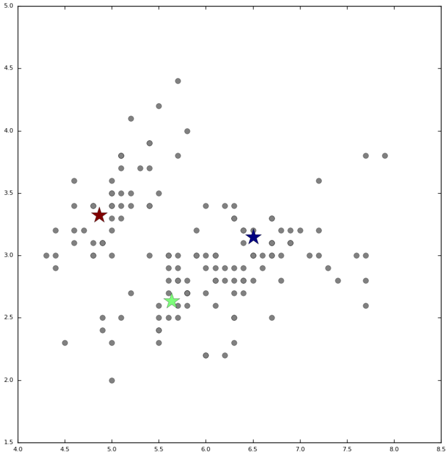
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location ⟸



Step-by-step Execution: Iteration 3 - Prep Step 1

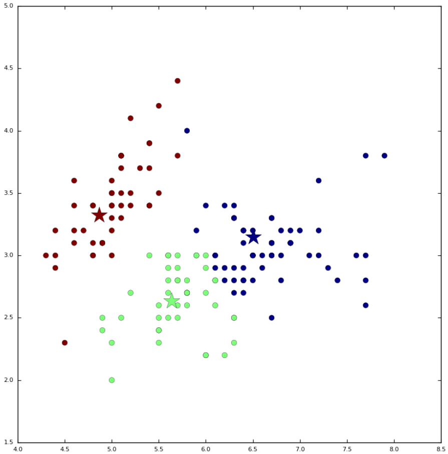
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location

$\Longleftarrow$



Step-by-step Execution: Iteration 3 - Step 1

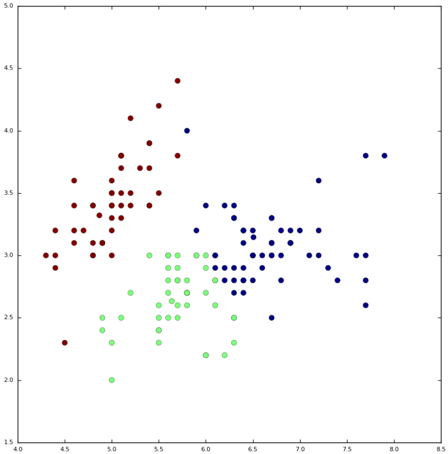
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid ⟸
    2. Move centroids to new average location



Step-by-step Execution: Iteration 3 - Prep Step 2

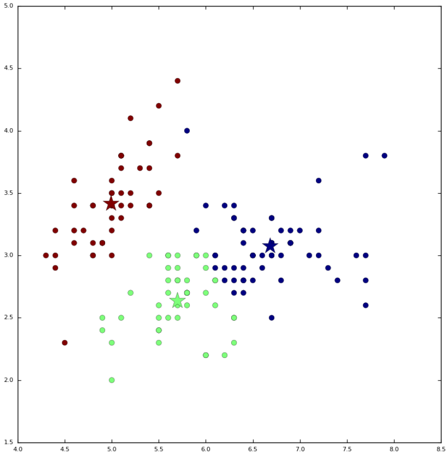
1. Initialize centroids.
2. While not stopping condition:
    1. Assign points to centroid
    2. Move centroids to new average location

⟸



Step-by-step Execution: Iteration 3 - Step 2

1. Initialize centroids.
2. While not stopping condition:
   1. Assign points to centroid
   2. Move centroids to new average location $\Longleftarrow$



Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing k

Evaluating *k*-Means

- How can we quantify how "good" our clustering is?

Evaluating k-Means

- How can we quantify how "good" our clustering is?
- A good measure should quantify how similar things are in a cluster.

Evaluating k-Means

- How can we quantify how "good" our clustering is?
- A good measure should quantify how similar things are in a cluster.
- The metric that we will use is called intra-cluster or within cluster variance:

$$WCV = \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Overview

Supervised vs. Unsupervised Learning

Clustering

- Intuition
- Definition

k-Means Algorithm

- Pseudocode
- Centroid Initialization
- Stopping Criteria
- Step-through
- Evaluation
- Problems
- Choosing *k*

Problems

- Centroids that are "discovered" will likely be different depending on initialization.

Problems

- Centroids that are "discovered" will likely be different depending on initialization.

  $\longrightarrow$ Run algorithm more than once and choose the run that yields the smallest intra-cluster variance.

Problems

- Centroids that are "discovered" will likely be different depending on initialization.
  $\longrightarrow$ Run algorithm more than once and choose the run that yields the smallest intra-cluster variance.

- k-Means is highly dependent on distance as a metric.

Problems

- Centroids that are “discovered” will likely be different depending on initialization.
  $\longrightarrow$ Run algorithm more than once and choose the run that yields the smallest intra-cluster variance.

- *k*-Means is highly dependent on distance as a metric.
  $\longrightarrow$ Normalize features before clustering.

Problems

- Centroids that are "discovered" will likely be different depending on initialization.
  $\longrightarrow$ Run algorithm more than once and choose the run that yields the smallest intra-cluster variance.

- k-Means is highly dependent on distance as a metric.
  $\longrightarrow$ Normalize features before clustering.
  $\longrightarrow$ Have to think about the curse of dimensionality.

Overview

Choosing k

Unsupervised

Choosing k is HARD!!! It usually takes some work and you're never quite sure if you're "right".

Choosing k

Unsupervised

Choosing k is HARD!!! It usually takes some work and you're never quite sure if you're "right".

There are a number of ways you can go about choosing k:

- Domain knowledge
- Elbow method
- Silhouette score
- GAP Statistic

Elbow Method

- Looks at the total amount of within-cluster sum of squares (WCSS) across all the clusters for different values of k.

$$WCSS = \sum_{k=1}^{K} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

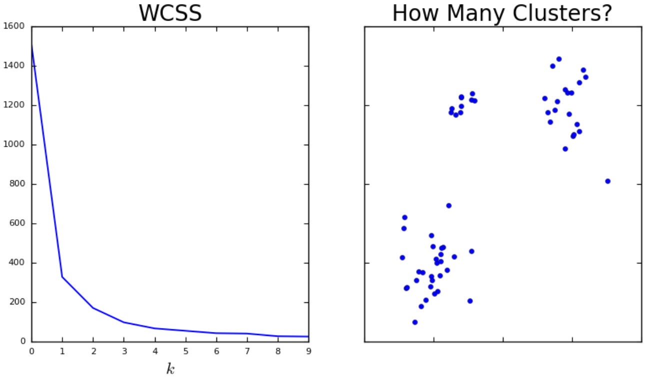
Elbow Method

- Looks at the total amount of within-cluster sum of squares (WCSS) across all the clusters for different values of k.

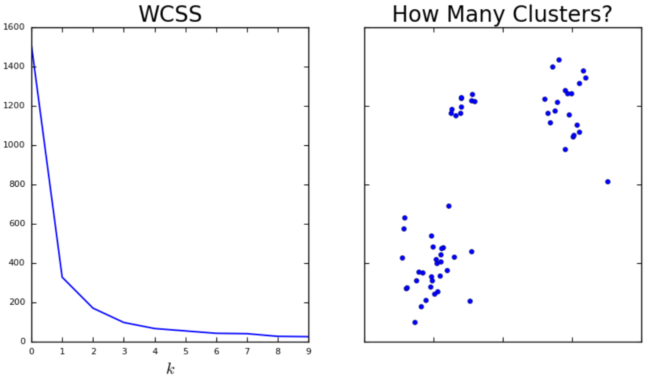
$$WCSS = \sum_{k=1}^{K} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

- Chooses the k such that adding one more cluster doesn't decrease the WCSS by much more. Leads us to look for an elbow in the k vs. $WCSS$ plot.

Elbow Method

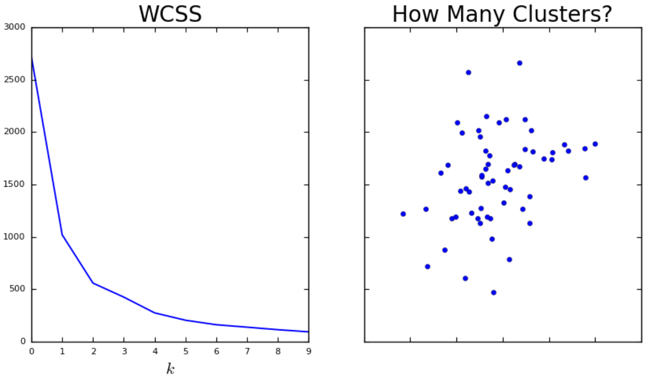


Elbow Method

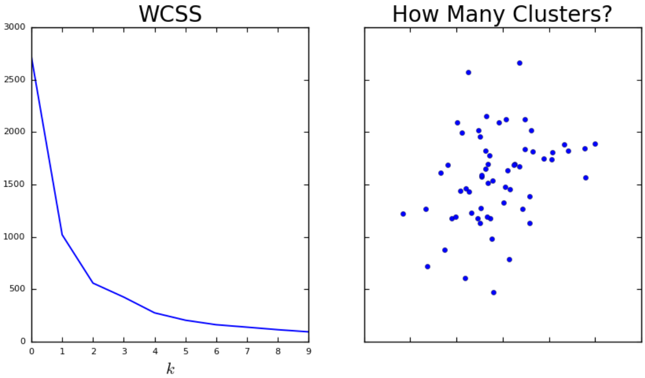


Question: Do you think the elbow will always be so obvious?

Elbow Method - Not Always So Clear



Elbow Method - Not Always So Clear



Question: How is this related to the curse of dimensionality?