CS M146: Introduction to Machine Learning
Clustering

Aditya Grover

The instructor gratefully acknowledges Sriram Sankaraman (UCLA) for some of the materials and organization used in these slides, and many others who made their course materials freely available online.

 https://aditya-grover.github.io/  @adityagrover_

Unsupervised Learning

- Supervised learning used labeled data pairs (**x**, y) to learn a function $f : X \rightarrow Y$
  - But, what if we don't have labels?

- No labels = **unsupervised learning**
- Only some points are labeled = **semi-supervised learning**
  - Labels may be expensive to obtain, so we only get a few

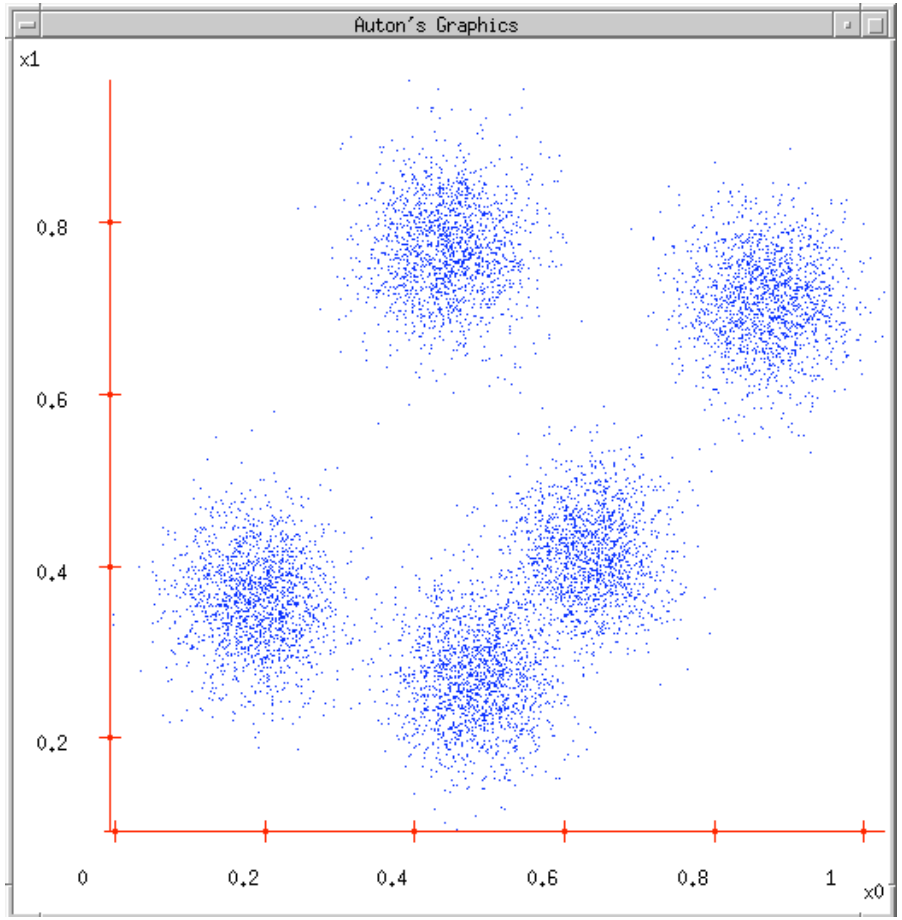
- **Clustering** is the unsupervised grouping of data points. It can be used for **knowledge discovery**.

k-means Clustering

Some material adapted from slides by Andrew Moore, CMU.

Visit http://www.autonlab.org/tutorials/ for
Andrew's repository of Data Mining tutorials.

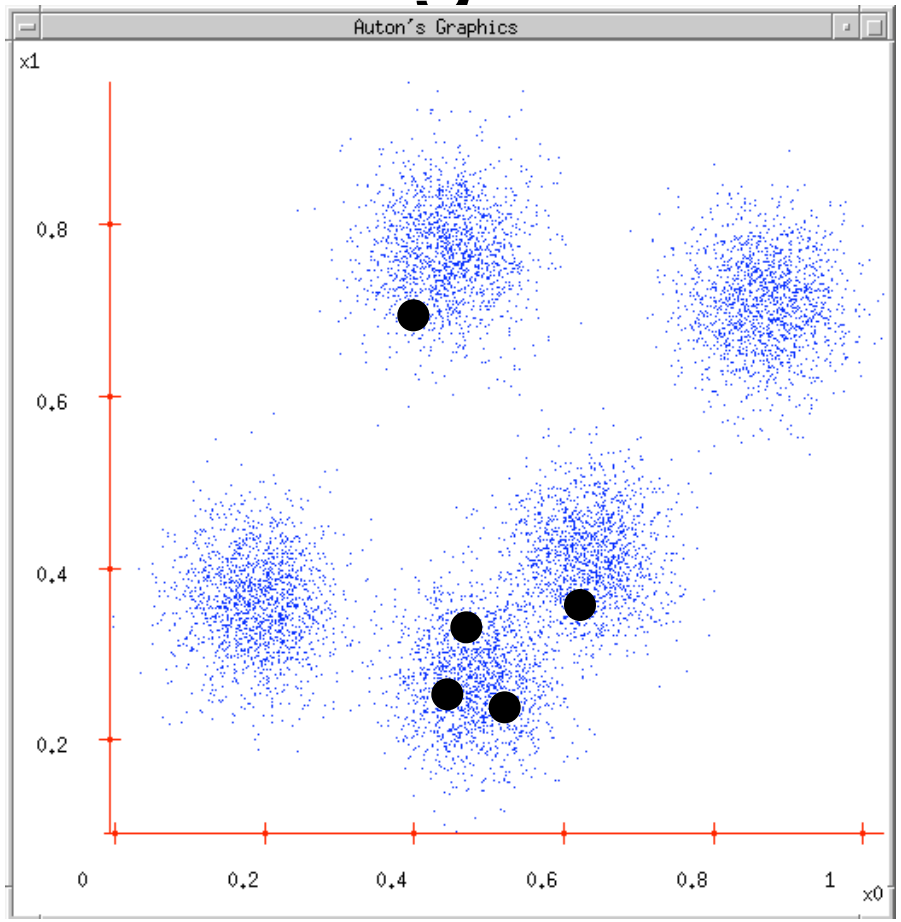
Clustering Data



k-means Clustering

k-means(k, X)

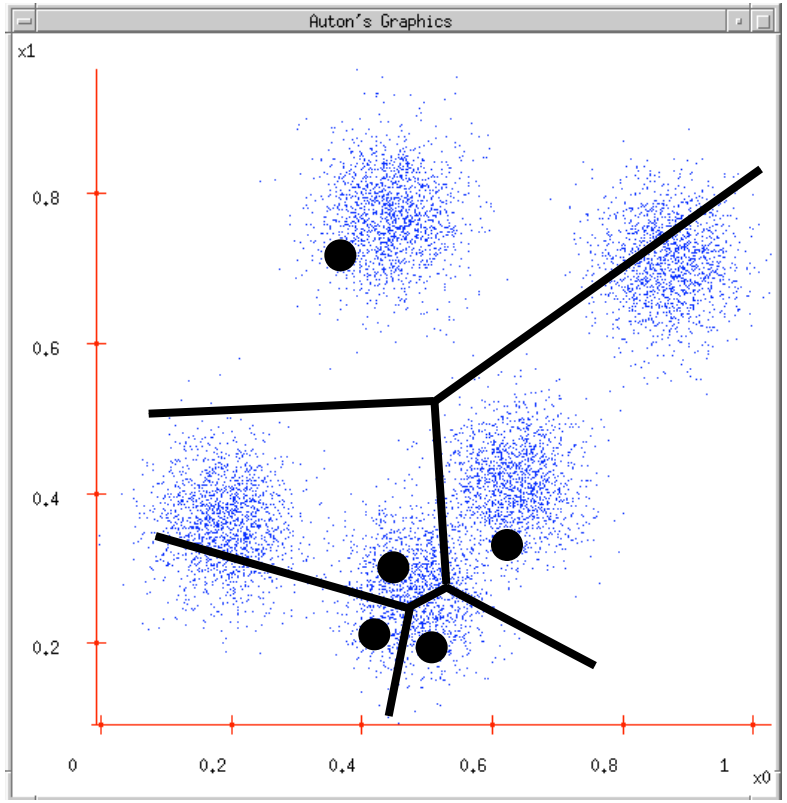
- Randomly choose k cluster center locations (centroids)
- Loop until convergence
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster



k-means Clustering

k-means(k, X)

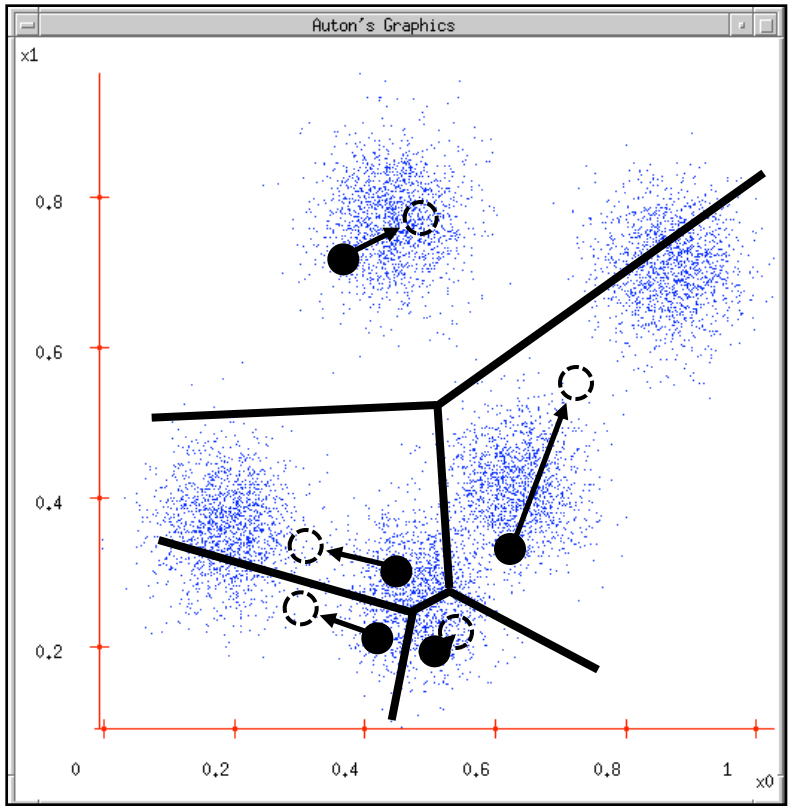
- Randomly choose k cluster center locations (centroids)
- Loop until convergence
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster



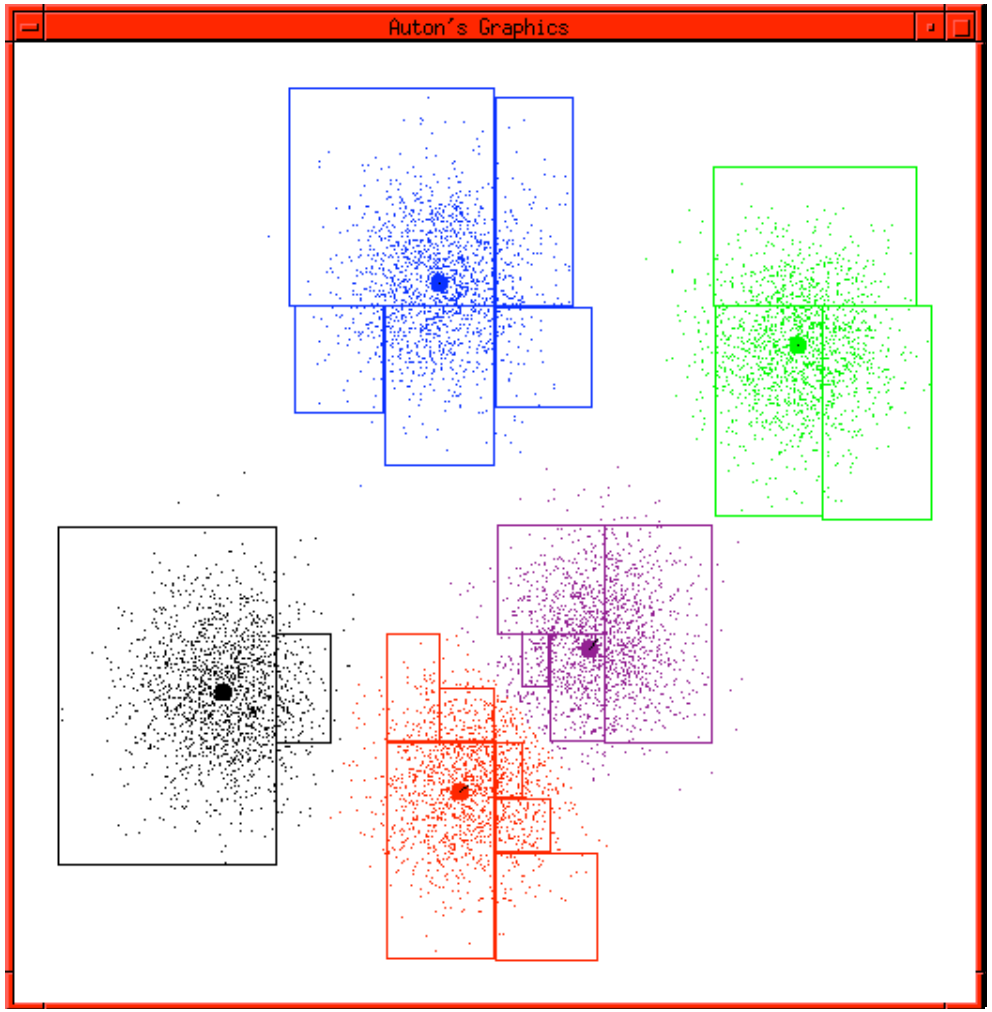
k-means Clustering

k-means(k, X)

- Randomly choose k cluster center locations (centroids)
- Loop until convergence
  - Assign each point to the cluster of the closest centroid
  - Re-estimate the cluster centroids based on the data assigned to each cluster



k-means Animation



Example generated by Andrew Moore using Dan Pelleg's super-duper fast k-means system:

Dan Pelleg and Andrew Moore. Accelerating Exact k-means Algorithms with Geometric Reasoning. Proc. Conference on Knowledge Discovery in Databases 1999.

Pseudocode

Inputs: Training set of n datapoints $\{\boldsymbol{x}^{(i)}\}$, number of clusters k

- **Step 1:** Initialize cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_k$
- **Step 2:**

    Loop until convergence: {

        For every point i, assign it to a cluster

        $$c_i := \arg\min_j \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j \right\|_2^2$$

        For every cluster j, re-estimate its centroid

        $$\boldsymbol{\mu}_j := \frac{\sum_{i=1}^{n} 1\{c_i = j\} \boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} 1\{c_i = j\}}$$

    }

Output: clusters for training points $c_{1:n}$, cluster centers for test points $\boldsymbol{\mu}_{1:k}$

k-means Objective Function

- k-means optimizes the following objective function:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|_2^2$$

  where $\boldsymbol{S} = \{S_1, \cdots, S_k\}$ is a partitioning over $X = \{x^{(1)}, \cdots, x^{(n)}\}$ such that $X = \cup_{i=1}^{k} S_i$ and $\boldsymbol{\mu}_i = \text{mean}(S_i)$

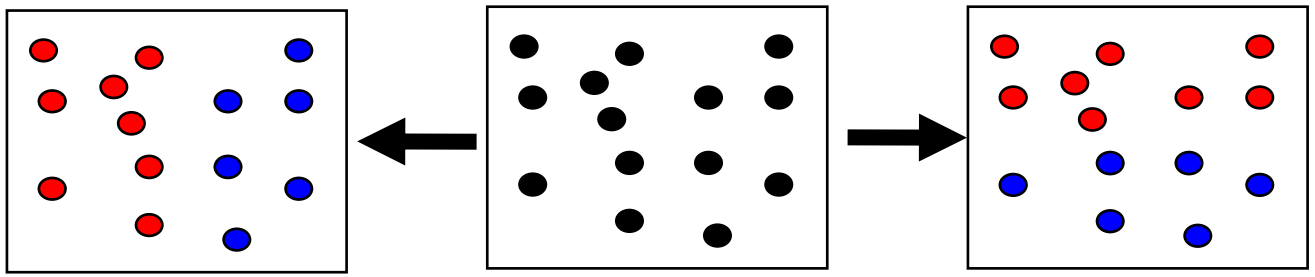
- Non-convex optimization problem. Solving it is NP-hard
- Convergence criteria for k-means: no change in objective value
- k-means is guaranteed to converge to a local optima
  - Worst case: Exponential in the number of data points
  - In practice, very fast to converge

Problems with k-means

- Very sensitive to the initial points
  - Do many runs of k-means, each with different initial centroids
- Must manually choose k
  - Validating the optimal k is not direct
    - Note that this requires a downstream performance measure
- Sensitive to outliers
  - k-median: Instead of computing mean in Step 2, compute median

Evaluating Clustering

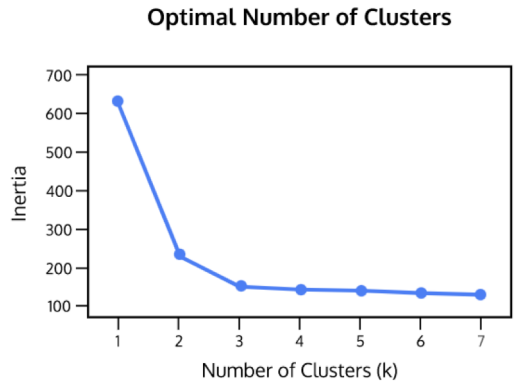
- Ambiguity in evaluating clustering. Which cluster is better?



- Performance metrics of clustering are often tied to a downstream task
  - E.g., human interpretability, classification (requires labels)

Evaluating Clustering

- Unsupervised evaluation: Low inertia **and** low k is desirable
  - **Inertia:** Sum of squared distances of datapoints to cluster centers
  - Tradeoff with number of clusters k: As k increases, inertia decreases



- Rule of thumb: choose k at which drop in inertia slows down (e.g., k=3 in the above figure)

Summary

- k-means clustering

    Hard assignment of points into k clusters

    Alternates between cluster assignment and centroid estimation

    Evaluation is tricky