k-Means Clustering

Lecture 6

Outline

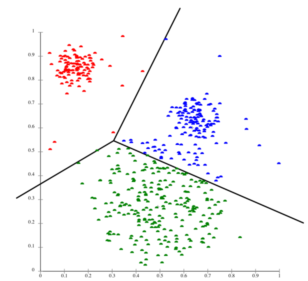
1. Learning to find instance groups without supervision
2. The k-Means algorithm
3. Issues and limitations
   – Bias vs. Variance
4. Generalizations and connections

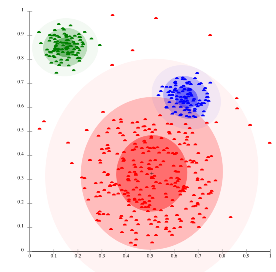
Clustering

- An **unsupervised** learning problem
- Goal: group a set of instances in such a way that objects in the same group (a **cluster**) are more similar (by some metric) to each other than to those in other clusters

Cluster Models

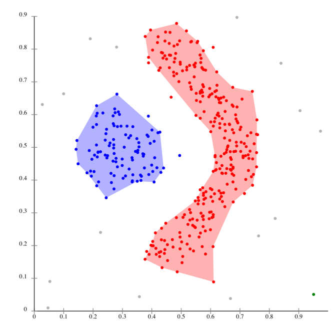
- Algorithms can be distinguished by several characteristics, including relationship between instance/cluster
  - Hard: binary relationship
  - Soft: weighted relationship

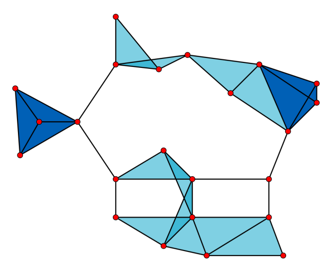




- And cluster assumptions
  - Centroid-based (e.g. *k*-Means)
  - Distribution-based
  - Density-based
  - Graph-based

  …





Cluster Validation

- Internal Validation
  - Similar to the idea of resubstitution error (i.e. use the dataset itself)
  - Dunn Index: maximize the ratio between the minimal inter-cluster distance to maximal intra-cluster distance
- External Validation
  - Similar to the idea of training/testing (i.e. require evaluation dataset + clusters/classifications)

k-Means

- Discovered by many researchers across numerous disciplines
  - You might see it referred to as a "problem" as opposed to an algorithm
- Centroid-based algorithm
  - Aims to minimize the within-cluster distances
  - Assumes instances are "spherically" oriented, variance of clusters is approximately equal
- Heuristic algorithm for NP-hard problem
  - It is computationally infeasible to find the "best" centroids for an arbitrary dataset

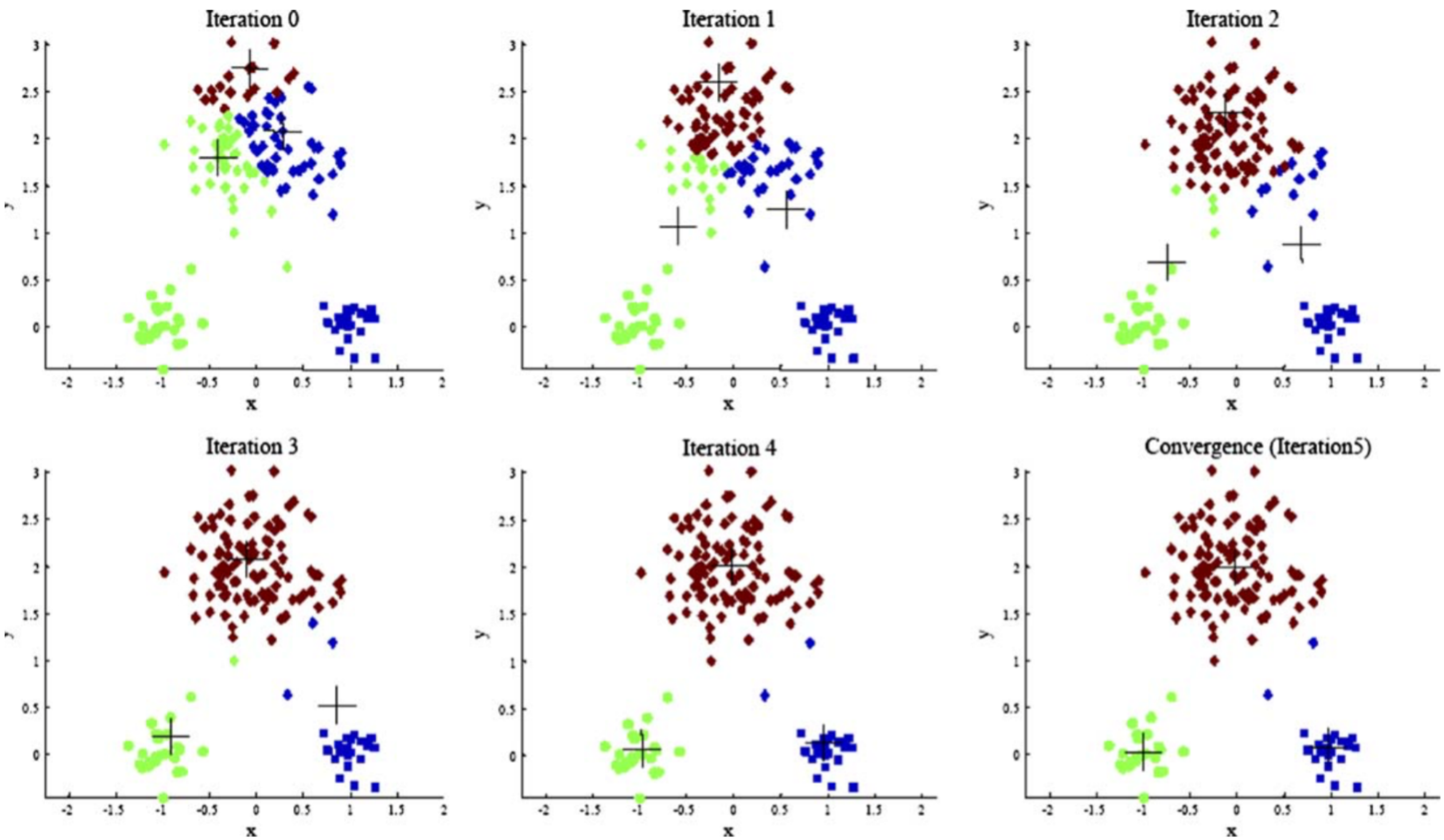
Algorithm Sketch

Inputs

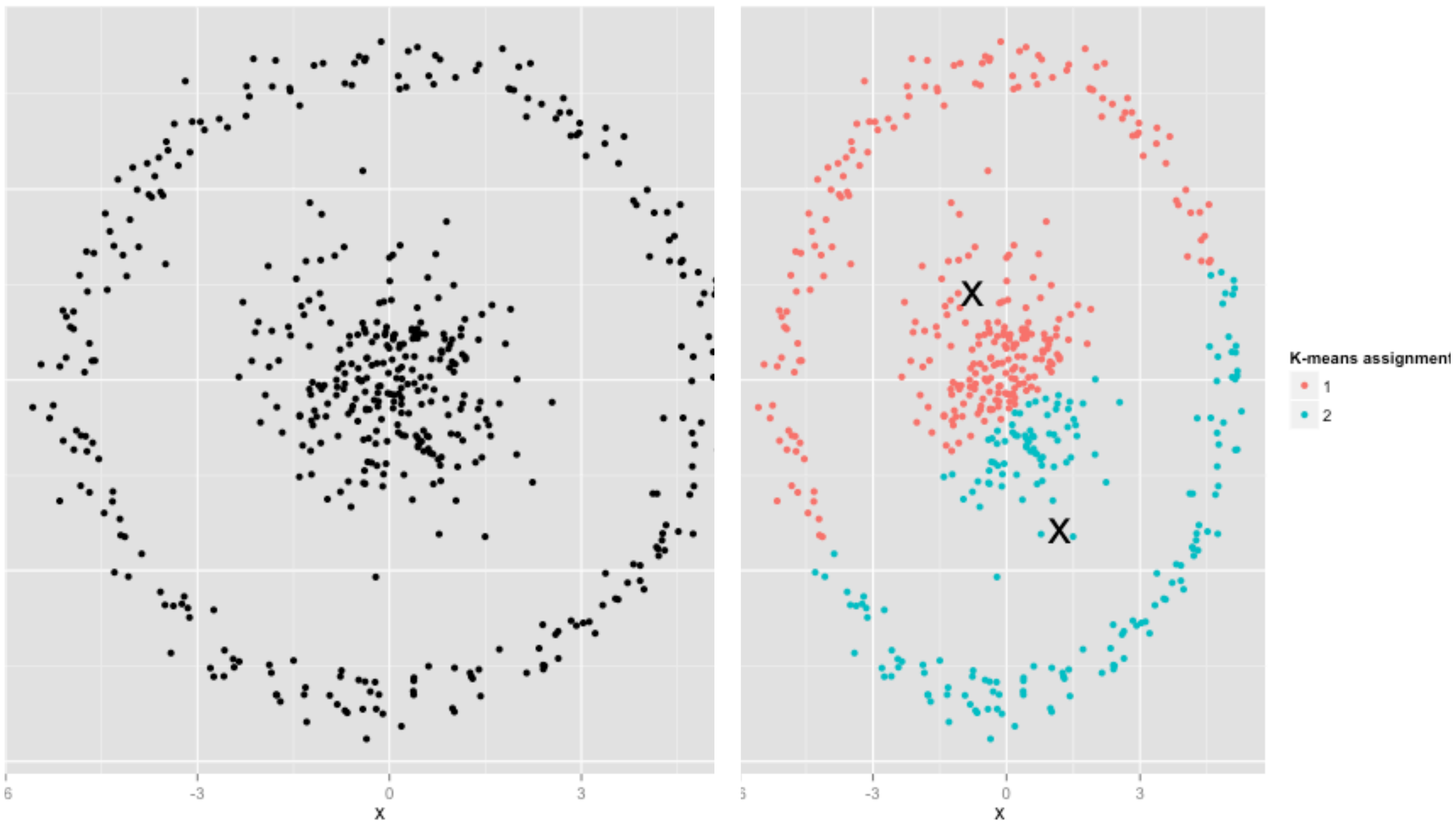
- *k* (number of centroids)
- df (distance function)

1. Initialize centroid positions
2. Repeat
   - For each instance, assign to *closest* centroid (via df)
     - If assignments haven't changed, done (*converged*)
   - For each centroid, relocate to mean of assigned instances

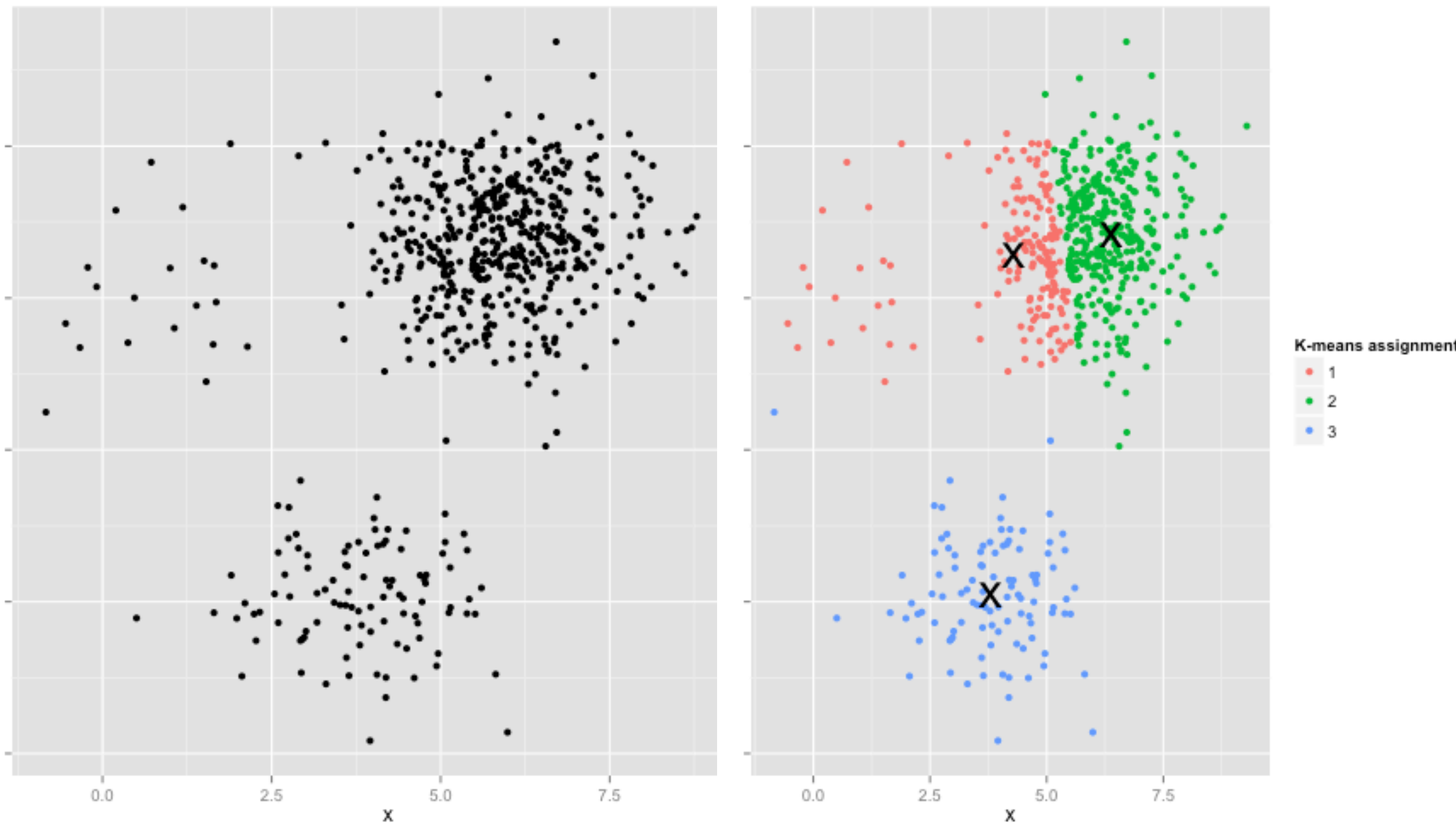
Example


Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Convergence (Iteration5)
x
y

Breaking Assumptions (1)



Breaking Assumptions (2)



Distance Function

Classically this is the Euclidean distance function, which will in effect minimize the within-cluster sum of square error (WCSSE)

$$\sum_{i=1}^{N} (\arg\min_{k} ||\boldsymbol{x_i} - \boldsymbol{c_k}||_2^2)$$

Choosing *k*

- Ideally this comes from an understanding of the data
- Can be done empirically via trying values and evaluating WCSSE/cluster quality
  - Possibly need to regularize
    - Bias vs. variance
  - See papers on 30 metrics, learning *k*
- Post-processing of clusters can also help
  - Splitting large clusters
  - Merging clusters

Initializing Centroids

k-Means is very sensitive to initial positioning, and so repeated trials may be required

Common methods

- **Forgy**: set the positions of the k clusters to k randomly chosen instances
- **Random partition**: assign a cluster randomly to each instance and compute means

Computational Complexity

- NP-hard in general to optimally solve the objective function
- *k*-Means is $\mathcal{O}(nkdi)$
  - n = # instances
  - k = # clusters
  - d = # dimensions
  - i = # iterations till convergence
    - If structure exists, small; typically good ~ 12

Variations

- *k*-mediods: rather than a mean, chooses best instance for next centroid location
- Nearest centroid classifier: run k-means on dataset, then 1-NN on clusters

Checkup

- ML task(s)?
  - Classification: binary/multi-class?
- Feature type(s)?
- Implicit/explicit?
- Parametric?
- Online?

Summary: *k*-Means Clustering

- Practicality
  - Easy, generally applicable
    - Suboptimal results if data does not satisfy assumptions
  - Very popular
- Efficiency
  - Considered linear in size of the dataset
- Performance
  - Heuristic, may need post-processing