Machine Learning for Data Mining
Week 6: Clustering

Christof Monz

Overview

- K-Means clustering
- Agglomerative clustering

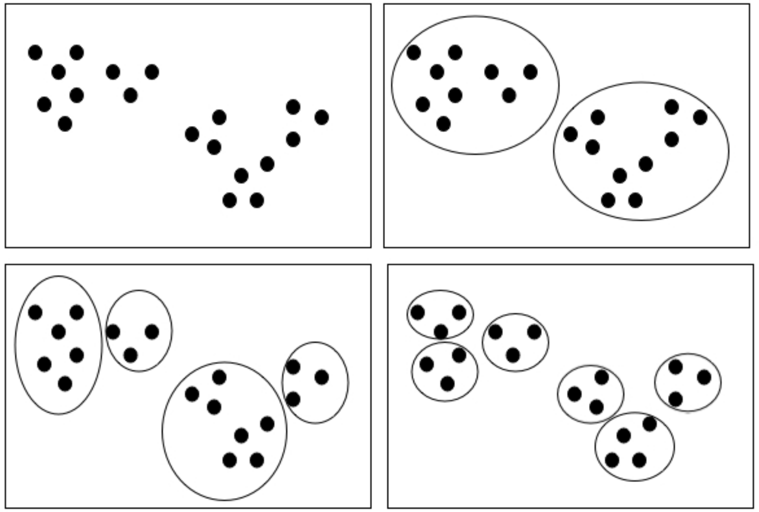
Clustering

- In classification we assume a pre-defined set of classes
- Each test example is assigned one of the classes
- If we don't know what the classes are, can we still organize our data into cluster that share many characteristics?

Clustering

- Partition unlabeled examples into disjoint subsets of clusters, such that:
  - Examples within a cluster are very similar
  - Examples across different clusters are very different
- Discover new categories in an unsupervised manner as there are no sample category labels provided (since we don't know what the classes are)

Clusters



Clustering

- There are two basic types of clustering: partitional and hierarchical
- Partitional clustering divides the data into non-overlapping subsets (clusters) without any cluster-internal structure
- Hierarchical clustering are organized as trees where each node is the cluster consisting of the clusters of its daughter nodes

K-Means

- K-Means is a partitional clustering approach
- K is the number of desired clusters (this has to be pre-defined by the user)
- K-Means uses a distance measure between points where distance is defined as above
- K-Means uses *centroids* (or prototypes) as defined above

K-Means Algorithm

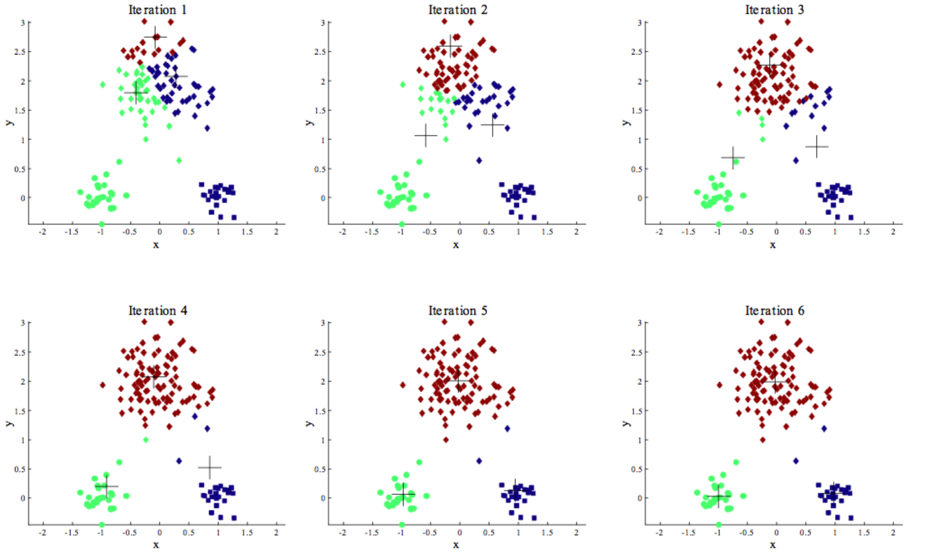
Select K random points from the data as initial seed centroids, $\{x_1, \ldots, x_K\} = C$
Until convergence or other stopping criterion:

- For each point x assign it to the closest centroid c, such that $c = \text{argmin}_{c' \in C} d(x, c')$
- Update each cluster $c = \{x \mid c = \text{argmin}_{c' \in C} d(x, c')\}$
- Re-compute the centroid of each cluster

K-Means: Error Estimation

- In order to compare two different runs of the K-Means algorithm we have to be able estimate its quality
- There is no ground truth (we don't know the clusters/labels beforehand)
- Instead the **sum of squared errors (SSE)** is used:
  $$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in c_i} d(x, c_i)^2$$

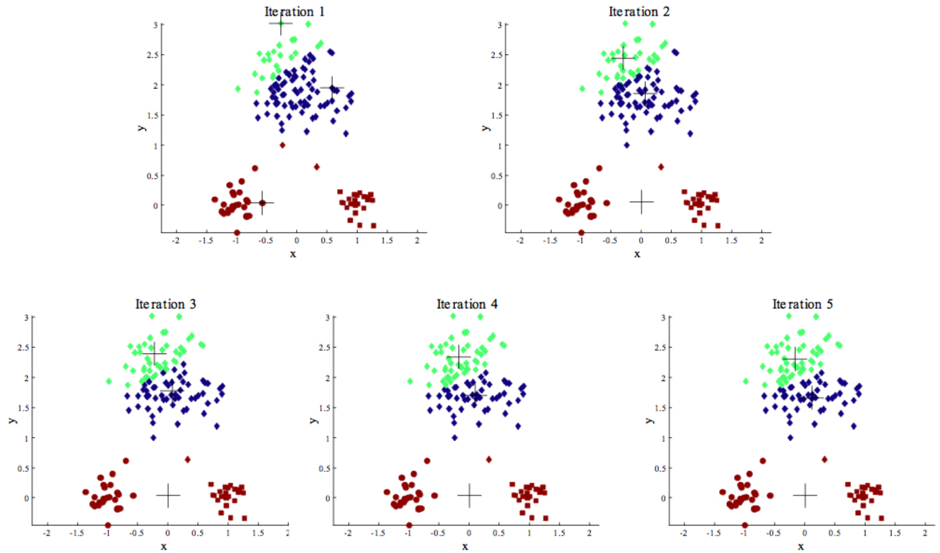
K-Means: Example



Initialization

- As mentioned above, the K initial seed clusters are selected randomly
- 'Wrong' initialization can lead to poor clusters
  - This can be partially addressed by choosing several initializations
  - Apply K-Means to all of them and keep the one with the lowest SSE

K-Means: Initialization Example



Outliers

- K-means is susceptible to outliers
- Outliers inflate the SSE
- Simple approach is to remove outliers beforehand
- If the data is skewed, outliers can be the interesting cases and removal is inappropriate

Empty Clusters

- K-Means can result in empty clusters
- Empty-cluster centroids need to be re-defined
- Two centroid redefinition strategies:
  - Take the point that is farthest away from any current centroid (from a non-empty cluster)
  - Randomly choice a point from the cluster with the highest SSE
- Repeat this for all empty clusters

Incremental Updates

- In basic K-Means the centroids are re-computed after all points have been assigned to a cluster
- Alternatively, centroids can be re-computed incrementally after each assignment
  - Two updates: if a point is assigned to a different cluster (re-compute the centroids of the old and the new cluster of that point)
  - Zero updates: if a point stays in the same cluster
- Pro: no empty clusters
- Con: Depends on the order in which points are processed (some randomization required)

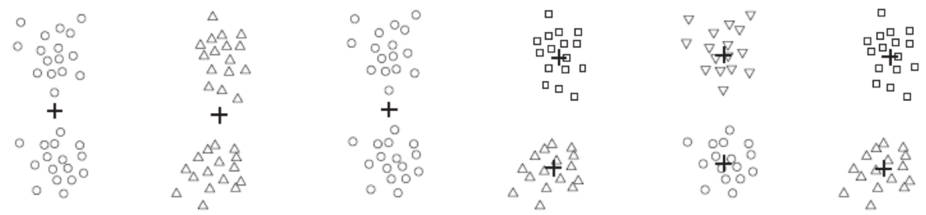
Bisecting K-Means

- Extension of the basic K-Means algorithm
- Basic idea: Initially split the data into two cluster, then further split one of the clusters, and so on, until there are K clusters
- Side-product: results in hierarchical clusters

Bisecting K-Means Algorithm

- Initialization: Set of clusters contains one cluster with all points
- Repeat until list of clusters contains K clusters
  Remove cluster from list
  For number of trials do:
    Bisect cluster with basic K-Means
  Select bisection with lowest total SSE
  Add both clusters to list of cluster

Bisecting K-Means Example

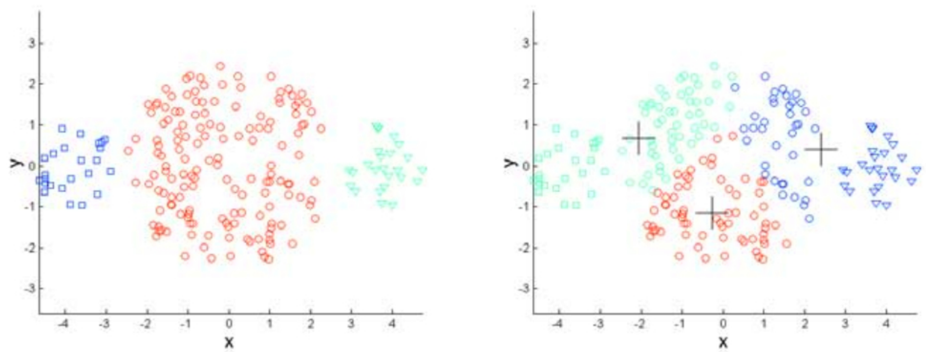


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

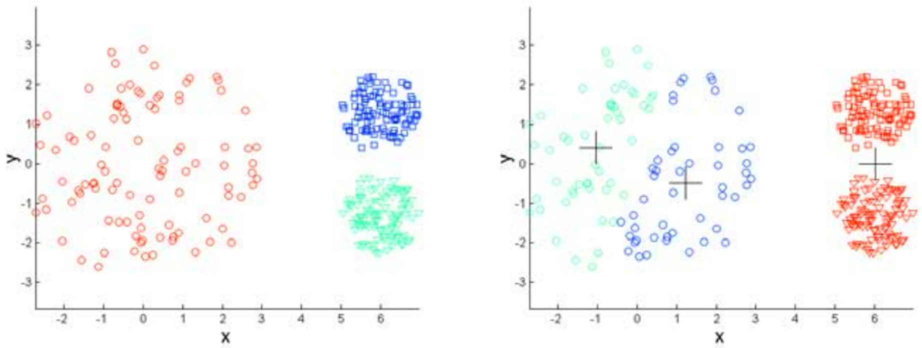
Bisecting K-Means

- Which cluster should be selected for bisection?
  - Cluster with largest SSE
  - Largest cluster (in terms of number of points)
- The 'trials' in the bisecting K-Means algorithm try different seed initializations (see basic K-Means)

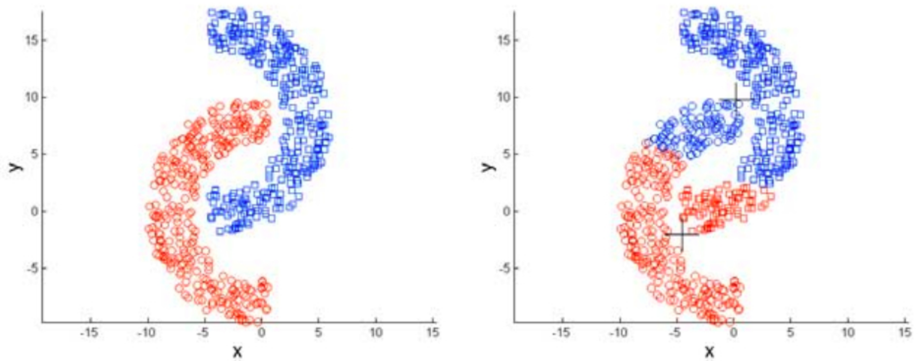
Limitations: Differing Sizes



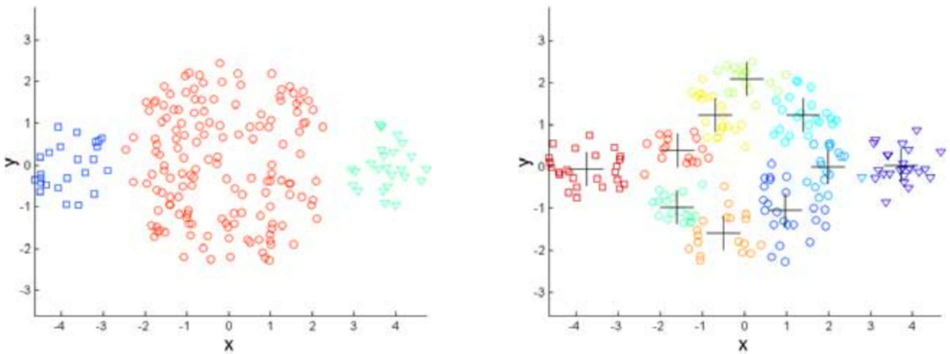
Limitations: Differing Densities



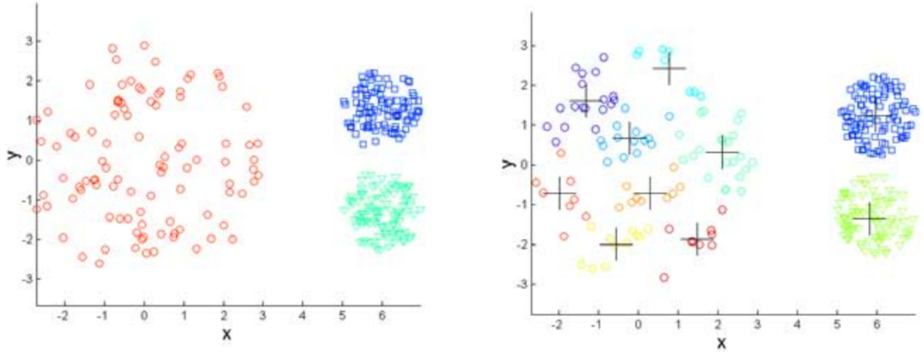
Limitations: Non-Globular Shapes



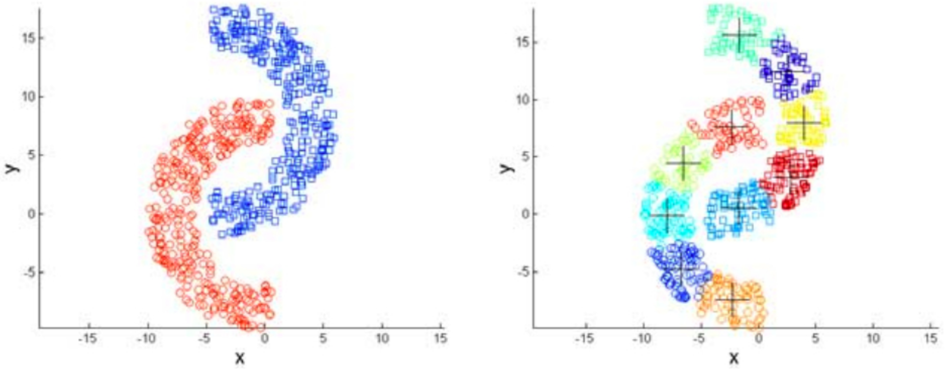
Larger K: Differing Sizes



Larger K: Differing Densities



Larger K: Non-Globular Shapes



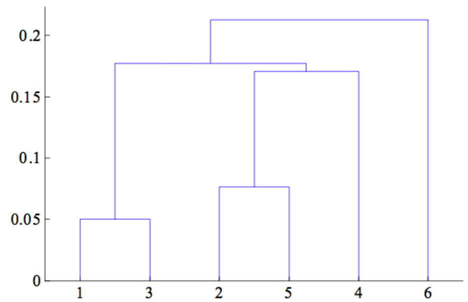
K-Means: Pros and Cons

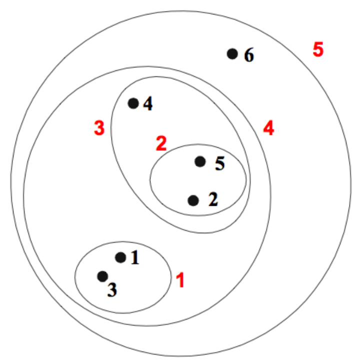
- K-Means is a simple clustering algorithm
- Runs efficiently
- Susceptible to initialization (but bisecting K-Means overcomes this to some extent)
- Limitations with respect to different size and densities and non-globular shapes
- Difficult to predict the best value for K in advance

Agglomerative Clustering

- Basic idea: start with individual points as cluster and then iteratively merge the closest two clusters into one larger cluster
- Results in a hierarchical clustering structure
- Requires a definition proximity between clusters

Agglomerative Clustering

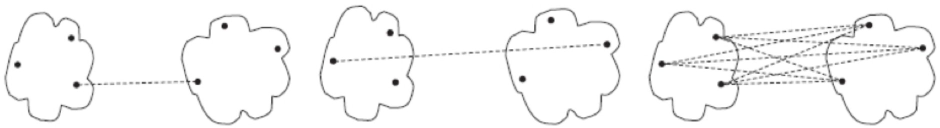




Agglomerative Clustering Algorithm

- Initialization: Form a cluster for each point
- Compute proximity matrix for all points
- Repeat until only one cluster remains
  - Merge the closest two clusters
  - Update proximity matrix by computing the distances between the new cluster and all other clusters

Proximity Between Clusters



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

Proximity Between Clusters

- MIN, MAX, and average use the proximities between individual from different clusters
- Alternatively, one can compute the centroids of each cluster and compute the distances between those
- Ward's method also use centroids but merges the pair of clusters that results in the lowest SSE wrt the new centroid
- Which proximity measure fares best depends on the shape/distribution of the points

Hierarchical Clustering Review

- Hierarchical clustering does lead to clusters with internal structure that can correspond to an actual taxonomy
- Does not try optimize any global criterion (such as SSE)
- All merging decision are made locally
- Once clusters are merged, they can not be separated again (unlike K-Means)

Recap

- Clustering vs classification
- K-Means
  - centroid
  - Sum of squared errors (SSE)
  - Initialization
  - Empty clusters
  - Incremental updates
  - Bisecting K-Means
- Agglomerative hierarchical clustering
  - Local merging of clusters
  - MIN, MAX, Average, and Ward's proximity measures