Unsupervised Learning: clustering

Used by B. Ombuki-Berman in COSC 4P80: neural networks course

Overview

- What is clustering?
- Hierarchical algorithms
- Partitioning algorithms
- Choosing an algorithm
- Evaluating clustering

Supervised vs. Unsupervised learning

- **Supervised learning**
  - the desired output must be available for each input vector used in the training.
- **Unsupervised learning**
  - no desired output; must rely on input data to learn similarities
  - Discover interesting features; separate sources that have been mixed together
  - tries to find structures in the input data space.
  - Can only work if there is something in the data to be discovered.
    - If the data is uniform or unstructured, or contains the wrong type of structure, the system may produce meaningless results
    - must test an unsupervised network to show if it makes sense after it has been trained
      - i.e., test, if the obtained structure is representative of the data.
      - this leaves a lot of responsibility to the user

Unsupervised Learning

- Given: data set *D*
  - Vectors of attribute values (*x1*, *x2*, …, *xn*)
  - No distinction between input attributes and output attributes (class label)
- Return: descriptor *y* of each *x*
  - **Clustering**: *grouping points* (*x*) into inherent regions of mutual similarity
  - **Vector quantization**: *discretizing continuous space* with best labels
  - **Dimensionality reduction**: *projecting many attributes* down to a few
  - **Feature extraction**: *constructing (few) new attributes* from (many) old ones

What is Clustering?

- Grouping of similar objects to produce a classification.

- Objects in cluster should be similar, but actual clusters should be different, otherwise belong to the same cluster.

What can be clustered?

- Images (e.g., astronomical data)
- Patterns (e.g., robot vision data)
- Words, documents etc
- Shopping items…
- …

Applications of Clustering I

- Data mining (DNA-analysis, Marketing studies, insurance studies,…)
- Text mining (text type clustering)
- Information retrieval (document clustering
- …

Applications of Clustering II

- Biological community formation
  - Groups of cells
  - Higher organisms
- Social networking
  - Cliques
  - Facebook-style friend grouping
- 3P98 convex hull multi-peels

Clustering

- We assume that the data was generated from a number of different classes. The aim is to cluster data from the same class together.
  - **How do we decide the number of classes?**
    - Why not put each data point into a separate class?
    - What is the payoff for clustering things together?
  - **What if the classes are hierarchical?**
  - **What if each data vector can be classified in many different ways?**
    - A one-out-of-N classification is not nearly as informative as a feature vector.

Measurement of Proximity/Similarity

- Central to clustering
  - How close or far apart are individuals from each other

- Topic scheduled as a separate talk (normally seminar topic)

Clustering algorithms

- **Hierarchical Clustering**
  - Tree structure
  - Determine clusters as you go
- **Partition Clustering**
  - Clusters determined a priori
  - Problem becomes data membership

Hierarchical Algorithms

- Data not partitioned into a particular number of clusters or classes in one step
  - Data partitioned into a series of partitions
  - How do we decide the number of classes?
    - Single cluster? N clusters each with single individual?
- Two types of methods
  - Agglomerative Algorithms
    - Series of fusion of n individuals into groups
  - Divisive Algorithms
    - Separate n individuals successfully into finer groupings
- Optimal number of clusters?
  - When to stop?

Hierarchical Algorithms

- Key step
  - decision on which sets to join (or split)
- Drawback
  - once the fusion (join) /split decision done, cannot be undone
  - cannot repair what was done in previous steps
- Decision criterion
  - many variants

Example: Hierarchical Tree structure

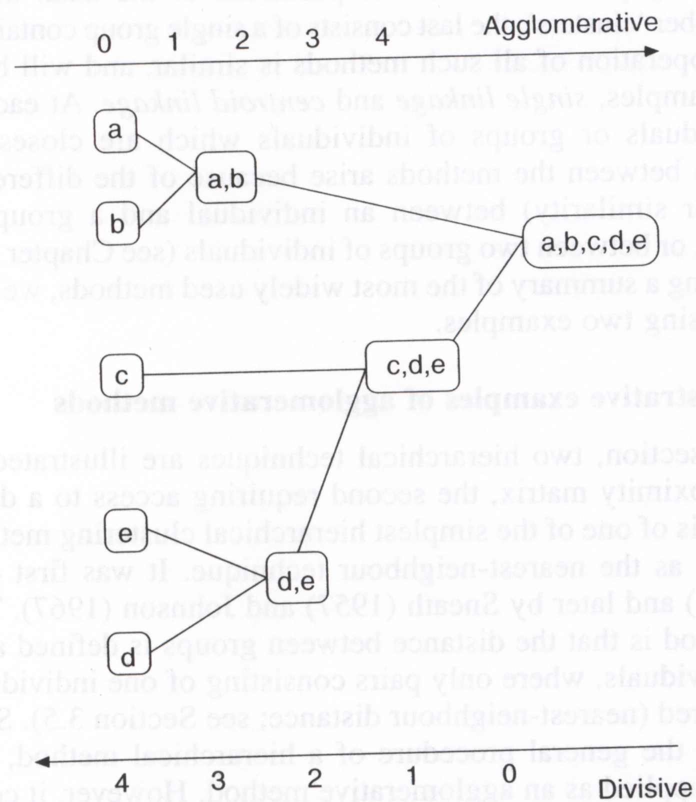


Figure 4.1 Example of a hierarchical tree structure. (Taken from *Finding Groups in Data*, 1990, Kaufman and Rousseeuw, with permission of the publisher, John Wiley & Sons.)

Main Steps: Agglomerative Algorithms

- Step 1: Place each object in its own cluster
- Step 2: Choose (via a linkage criterion), two clusters and merge them
- Step 3: If there is only one cluster left, stop. Else, go to Step 2

Problem: When to stop? After how many clusters?

Linkage criterions

- Single linkage
  - nearest Neighbor
    - selects cluster with shortest distance between the closest object in one cluster and the closest object in the other cluster
- Complete linkage
  - furthest neighbor
    - Selects the clusters with the shortest distance between the furthest object in the other cluster
- Average linkage
  - Selects the clusters with the shortest average distance between all objects in one cluster and all objects in one cluster
- Centroid linkage
  - Selects the clusters with the shortest distance between the centroid of one cluster and the centroid of another cluster

Comparisons

- Single linkage
  - Makes unbalanced clusters
  - long drawn-out clusters
  - does not take account of cluster structure

- Complete linkage
  - makes compact clusters
  - does not take account of cluster structure

- Average linkage
  - Between single and complete
  - Tends to join clusters with small variances
  - Takes account of cluster structure

Divisive Hierarchical Algorithms

- Start with one large cluster and successfully split the clusters
  - First consider the divisions of entire set into two non-empty sub-clusters
  - successively divide the sub-clusters of partition

- Complexity: 2 ^ (n-1) – 1 = O(2^n)
  - Compared to agglomerative methods which are

(n(n-1))/2 = O(n^2)

Divisive Hierarchical Algorithms

- Tricks exists on how to get around the expensive first steps.
- heuristic to find “good” partitions.
  - E.g., using a single attribute to make the division
    - monothetic method
      - Problem: multivariate data
  - E.g., Using all variables at each split
    - polythetic

Partitioning Algorithms/Optimization Algorithms

- Divide a set of objects into a given number of smaller clusters
- Mostly a 2-step process
  - Choose a set of representative objects*
  - Assign each remaining object to its nearest representative

Partition Clustering

- Partitioning data = data point membership
- K-means
- QT Clust
- Fuzzy c-means

K-means algorithm

- Widely used partitioning algorithm
  - Initially assign k-cluster centers to k randomly chosen instances
  - Repeat until converged
    - Assign each point to its closest representative
    - Recalculate positions of the centers
- Note: The representatives need not be actual

  data points.

K-means clustering

- Aim
  - Divide data points into K clusters such that some metric relative to the centroids of the clusters is minimized.
- example metrics to the centroids that can be minimized include:
  - maximum distance to its centroid for any point.
  - sum of the average distance to the centroids over all clusters.
  - sum of the variance over all clusters.
  - total distance between all points and their centroids.

K-means clustering

- **Step 1.** Choose **initial** *K* seed points (i.e., group centroids) into the space represented the data points being clustered using some method
- **Step 2.** Assign each data point to the group that has the closest centroid.
- **Step 3.** Re-compute the *K* cluster centers, when all points have been assigned
- **Step 4.** Iterate the procedure until it either converges or the count exceeds some threshold.

K-means

- What is it trying to optimize?
- Is it guaranteed to terminate?
- Is it guaranteed to find an optimal clustering?
- How should we start it?
- How to automatically choose the initial K, i.e., number of centers?

Optimality

- Not guaranteed!
- Choose starting points carefully
  - Use data points, far apart if possible.
- Perform many runs of K-Means
  - each from a different random starting point.
- Other tricks floating around

Choosing initial K

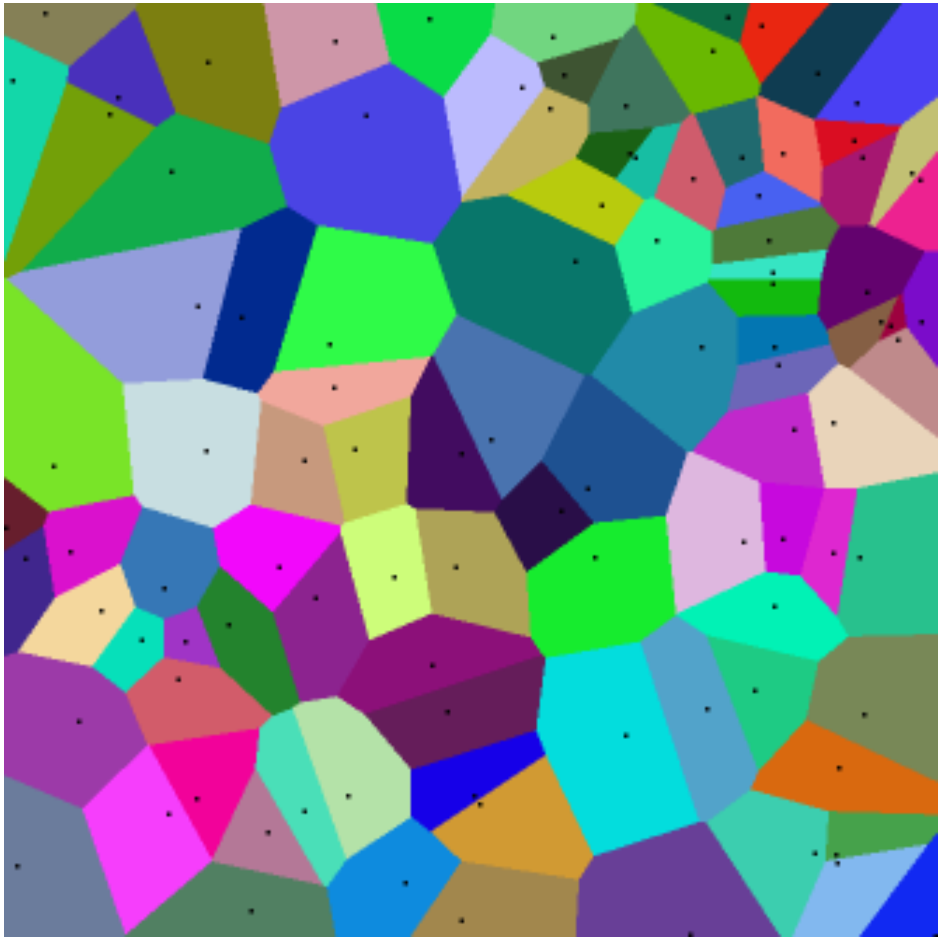
- Randomly chosen or can be just the first *K* data points in the data set.
- Can be imported from some other algorithm
- They need not even be actual data points.

K-means Clustering

- Lloyd’s Algorithm
  - Find Voronoi diagram
  - Calculate centroid for each voronoi cell
  - Center each point in its voronoi cell

K-means Clustering Voronoi Diagrams

- Voronoi Diagram
  - Set of hyperplanes
  - Each point to surrounding points
  - Forms set of boundary “lines”



K-means

- advantage
  - Simple to implement
  - Fast, so can be used on very large data sets
  - Can re-assign a data item to another cluster if initially assigned to a non-suitable cluster.

- disadvantage
  - outputs are influenced by the initial choice of seed points.
  - Cannot utilize meta-information
  - Randomness

K-means Variants

- Reading assignment + Seminar topic
- Other Clustering Methods: many exist

Validation

- A cluster algorithm will always produce clusters;
  - How do we know that the clusters are not artifacts of the algorithm?
- when to stop.
  - How many clusters are there?
  - If a hierarchical tree is produced, what levels of the tree are significant?

Validation

- Have we found meaningful clusters or just groups of data points that are not related?
- **External Criteria**
  - Use your clustering technique on artificial data & compare the results from the real data
  - Perform significance tests on external variables
- **Internal Criteria**
  - Rely totally on information from the data and clusters we have
    - Hierarchical methods can be validated by information from their tree structure
    - Correlation measures

Validation

- **Relative criteria**
  - Involves comparing the results obtained from your clustering algorithm with results for the same data with same algorithm but different parameters
  - Chances are an algorithm is correct the more algorithms agree on one clustering solution

Difficulties with Clustering

- Tends to fall in to local maximums
- Hard to find the “right” number of clusters
- Difficult to validate

Summary of clustering

- How many clusters *should* there be?
- How do we visualize large, multidimensional datasets?
- Objects to the cluster
- Which variables to use?
- How do we handle multidimensional datasets?
- How do we handle noise?
- Which clustering algorithm should we use?
- Which proximity measure should we use?