UNIVERSITY of
HOUSTON
DEPARTMENT OF COMPUTER SCIENCE

Clustering

Rakesh Verma

Clustering

Clustering is a family of methods used for finding similarity groups called **clusters** in data.

- Data is not always given class labels.
  - Classes are often not even known in these cases.

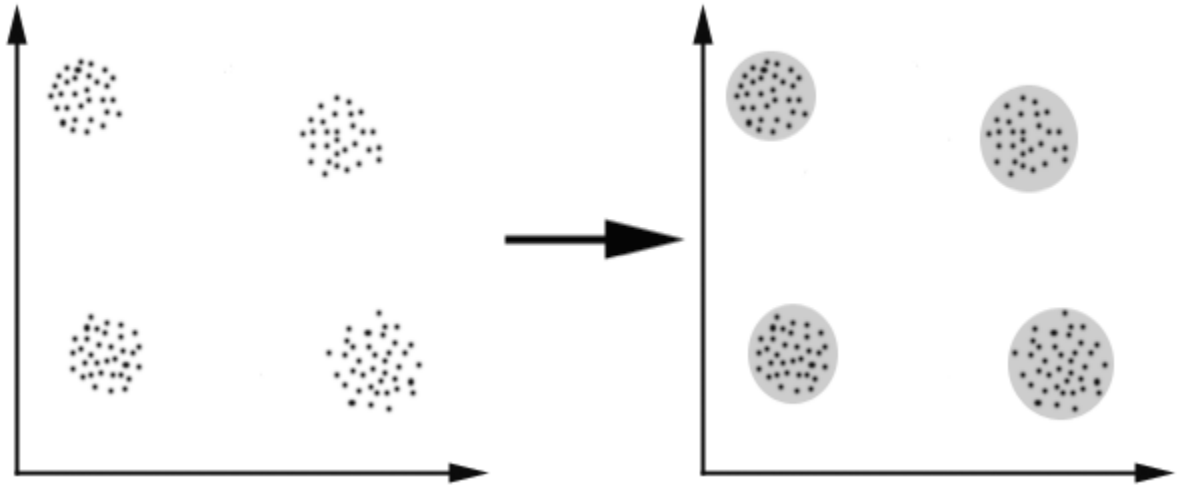
Clustering is unsupervised learning

- There is no canonical clustering for a dataset
- Supervision is not possible

Clustering

The data below has four natural clusters.

In practice, finding these clusters algorithmically is hard, yet easy for human eyes.



Why Clustering?

- The cooking staff at the Machine Learning Conference can only cook so many different meals.
  - They try to cluster all their attendees into groups based on what they're likely to eat.
  - Once they're done, they notice four clusters of common preferences
    - Chicken, Beef, Vegetarian, Gluten-free
- A security provider company wants to detect obfuscated malwares, but doesn't have a clue what they might be.
  - They cluster their suspicious applications based on their runtime behavior
  - They find a few distinct clusters
    - Adware, Bot, Ransomware, Rootkit, Spyware, etc.

Clustering

Clustering uses a similarity measure to identify "close" instances

Types of clustering

- Partitional
- Hierarchical

Cluster Quality Measures

- Maximizing inter-cluster distance, the space between clusters
- Minimizing intra-cluster distance, the space between members of the same cluster

Quality of the clustering is dependent on method, similarity measure, and application

Partitioning Algorithms

A partitioning method takes a dataset *D* and breaks it up into *k* clusters.

- *k* as a quantity is usually set by the experimenter
- Determining cluster count *k* is a difficult problem
  - Given the cluster count, determining the clusters is simplified

Methods

- Brute force
  - Try every single possible clustering.
- k-means
  - Clusters are defined by nearest centroids.
  - Calculate *k* centroids iteratively until they no longer change.

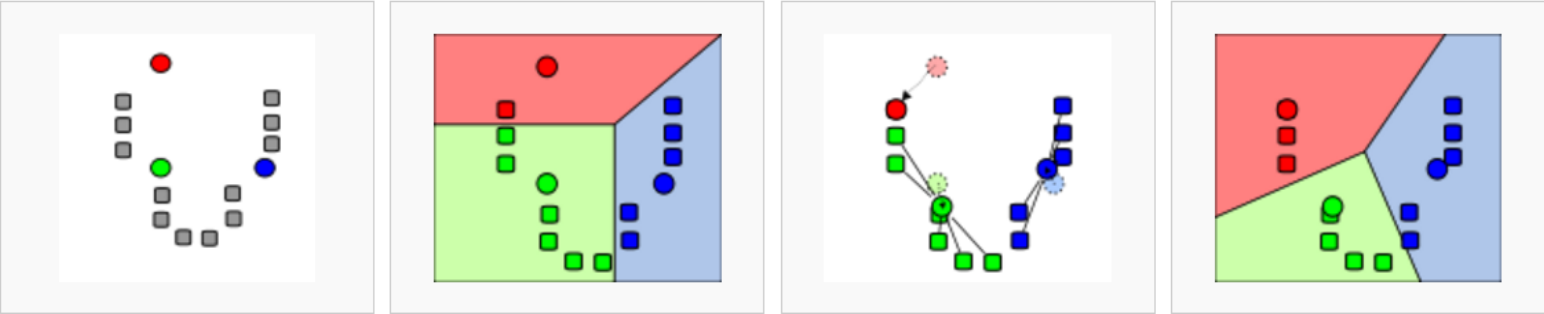
k-Means

Given *k*,

Take a random k-partition of *D*

Calculate centroids (mean points) of each cluster of the partition

Assign all instances in D to the partition whose centroid they're closest to



k-Means Summary

- Time complexity
  - O(*t*k*n*) where *t* is the number of iterations, *k* is the number of clusters, and *n* is the size of *D*
- Depends on dataset forming a measurable space
  - k-modes can be used instead for categorical data
  - k-medoids (an occurring middle point) can be used for discrete attributes with wide ranges
- *k* must be provided, not learned
  - Intuitively, one can learn the number of clusters using *k* as an upper bound.
    - During k-Means remove clusters as they become empty.
    - Results in k' clusters where $k' < k$.
- Sensitive to noise and outliers
- Convex clusters only.

k-Means Variants

Variations

- Selection of the initial k means
- Dissimilarity calculations
- Strategies to calculate cluster means

Categorical data

- Replacing means with modes for clusters
- Use new dissimilarity measures to deal with categorical objects
- Use a frequency-based method to update modes of clusters
- A mix of categorical and numerical data: k-prototype method

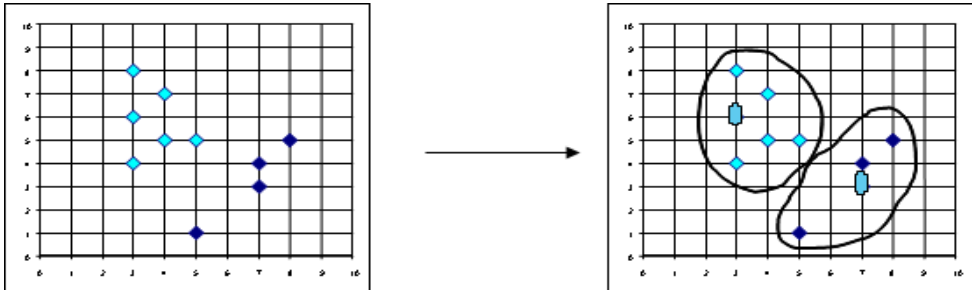
k-Means to k-Medoids

Below is a demonstration of k-Medoids'

k-Medoids enforces that the cluster must be defined by one of the points in the dataset.

k-Medoids is outlier resistant

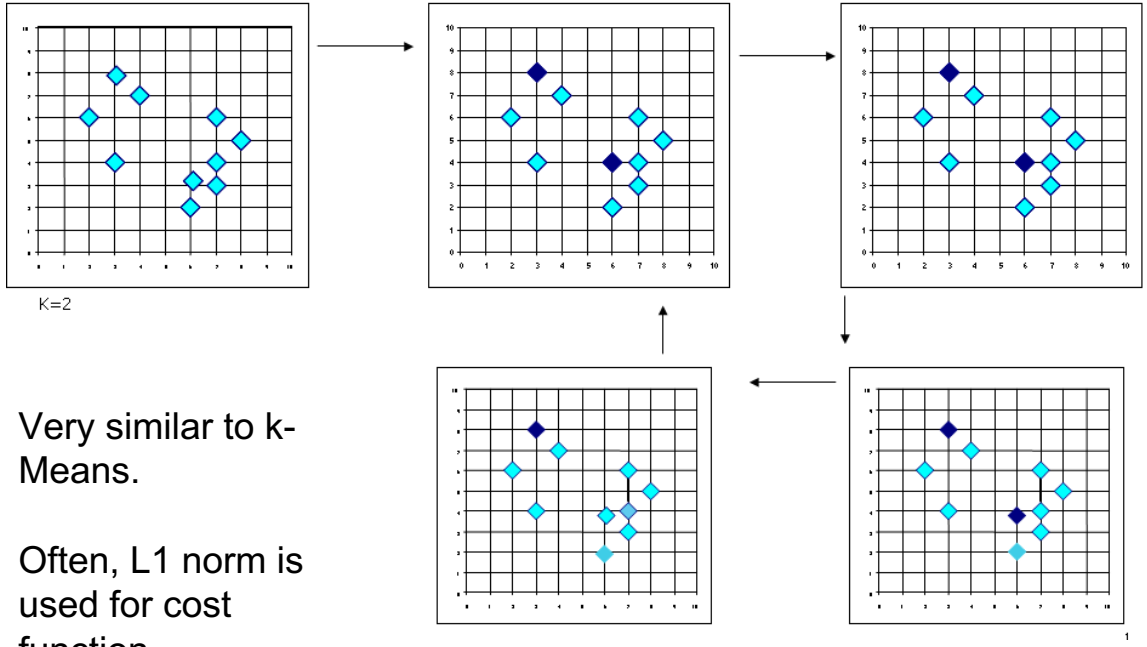
Because it is simply an order relation, the distance of outliers does not impact the result as it does with k-Means.



PAM: k-Medoids

Given *k*,

- Take a random k-partition of *D*
- Assign all instances in D to the partition whose centroid they're closest to
- Calculate better medoids for each cluster of the partition
- Repeat until no better choice of medoids exists.



Very similar to k-Means.

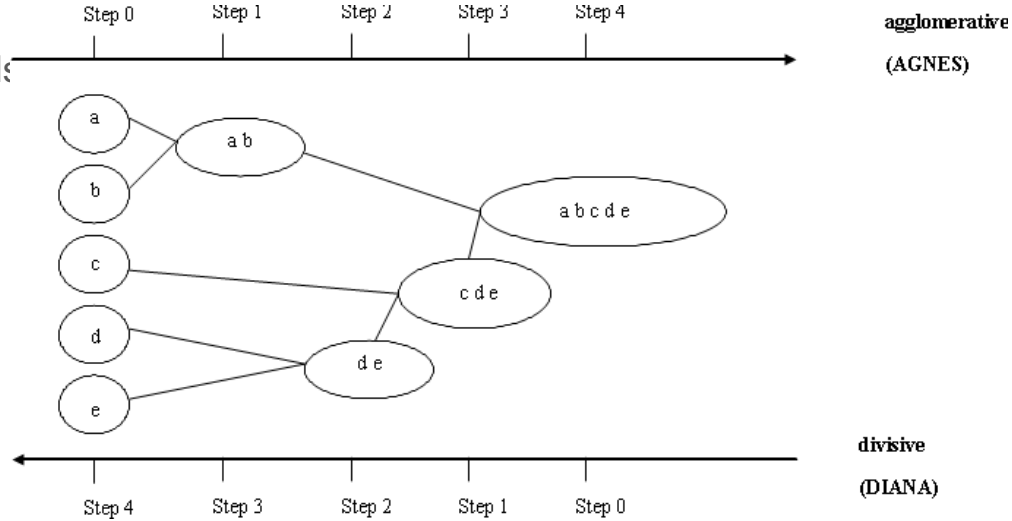
Often, L1 norm is used for cost function.

Hierarchical Clustering

Use distance matrix as clustering criteria

This method does not require the number of clusters *k* as an input

Instead, needs

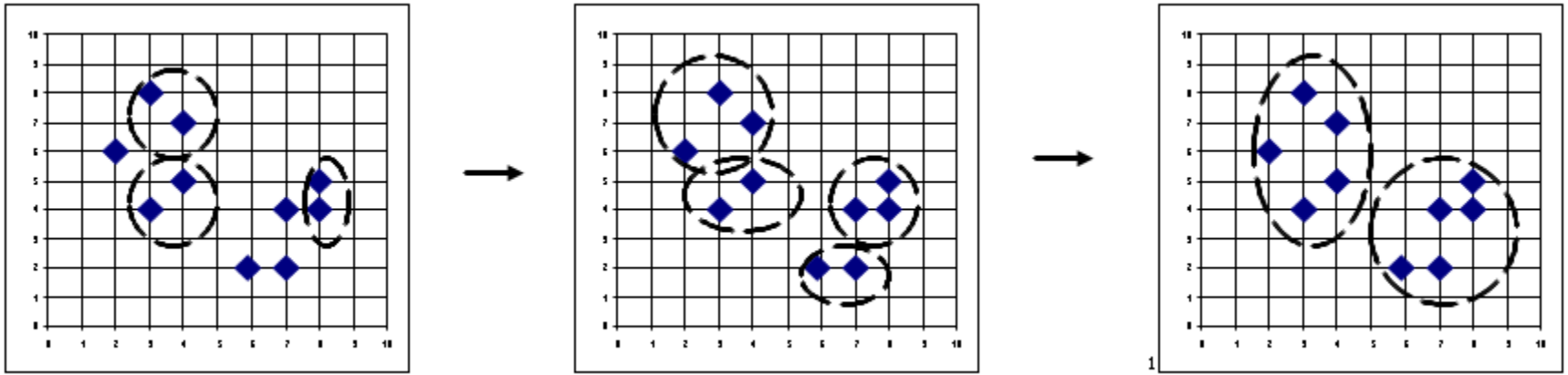


AGNES: Agglomerative Nesting

Use the single-link method and the dissimilarity matrix

Merge nodes that have the least dissimilarity

Go on in a non-descending fashion



Distances between Clusters

Single link

Smallest distance between an element in one cluster and an element in the other.

Complete link

Largest distance between an element in one cluster and an element in the other

Average

Average distance between an element in one cluster and an element in the other

Centroid

Distance between the centroids of two clusters

Medoid

Distance between the medoids of two clusters

Calculating the Centroid, Radius, and Diameter

Centroid

Middle of the cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

Radius

Square root of the mean distance to the centroid of the cluster

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

Diameter

Square root of the mean distance between all points from each other in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Hierarchical Clustering

Pros

- Provides a set of smaller clusters
- Produces an ordering for all the objects

Cons

- Hierarchical Clustering does not scale well
  - Time complexity of AT LEAST $O(n^2)$
- Mistakes and outliers are propagated forward
  - Post-result corrections are not possible without rerunning the experiment.

BIRCH: Balanced Iterative Reducing and Clustering Hierarchies

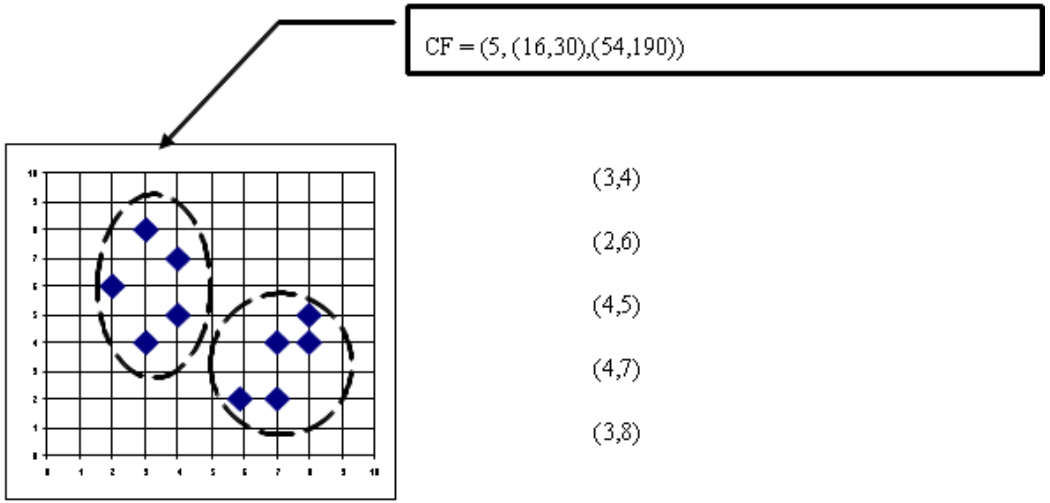
- Incrementally construct a Clustering Feature tree
  - Scan DB to build an initial in-memory CF tree
    - A clustering feature tree is a multi-level compression of the data
      - Preserves the inherent clustering structure of the data
  - Use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- Pros
  - Finds a good clustering with a single scan
  - Improves the quality with a few additional scans
- Cons
  - Only handles numeric data
  - Sensitive to the data's ordering

Clustering Feature Vector

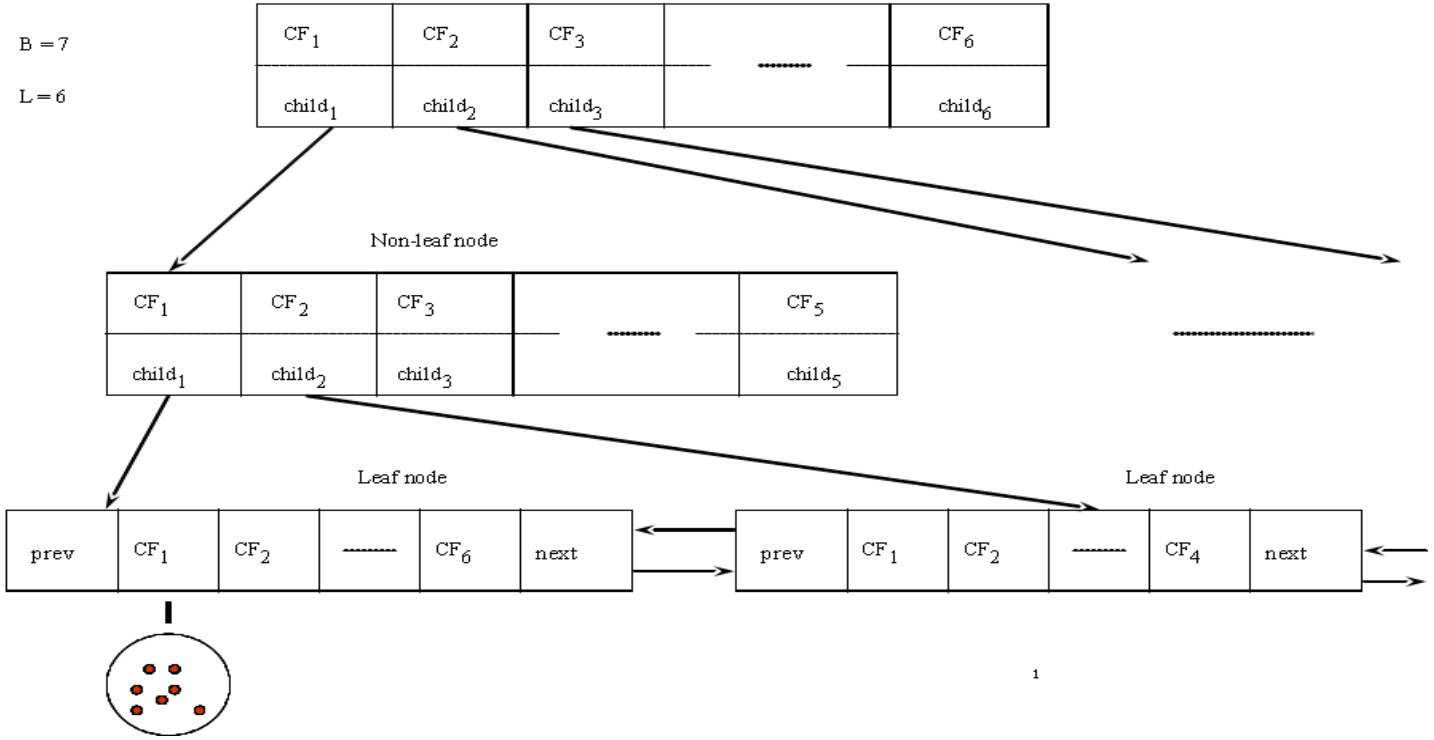
Number of data points ***N***

Linear sum of N points ***LS***

Square sum of N points ***SS***



Clustering Feature Tree



BIRCH Algorithm

- For each point in the input
  - Find closest leaf entry
  - Add point to leaf entry and update CF
  - If entry diameter > max_diameter, then split leaf, and possibly parents
- Pros
  - Algorithm is O(n)
- Cons
  - Sensitive to insertion order of data points
  - Since we fix the size of leaf nodes, clusters may be bigger or smaller than natural
  - Clusters tend to be spherical given the radius and diameter measures

Determining Cluster Count

Empirical Method

Choose k to be approximately $\sqrt{\frac{n}{2}}$

Elbow Method

Select k such that the increasing the number of clusters will not significantly improve cluster quality.

Cross validation method

Divide the dataset into m parts

Build a clustering model on m - 1 parts

Test the quality of the clustering on the m'th part.

Clustering Quality

- External
  - Supervised
  - Employs criteria not inherent to the dataset
  - Compare a clustering against prior or expert-specified knowledge using certain clustering quality measure
- Internal
  - Unsupervised
  - Criteria derived from data itself
  - Evaluate the goodness of a clustering by how compact and separated the clusters are
- Relative
  - Directly compare with different clusterings

Clustering Quality Measures Q(C, T)

For a clustering C given the ground truth T

Q is good if it satisfies the following 4 essential criteria

- Cluster homogeneity:
  - Cluster contents should all be similar to each other.
- Cluster completeness:
  - Clusters should correlate with existing categories.
- Rag bag compatibility
  - A cluster for putting undesirable instances
  - Putting a dissimilar instance into the cluster should be penalized more than putting it into a rag bag
- Small cluster preservation:
  - Small clusters are more affected by splitting than large clusters.

Common External Measures

Matching-based measures

Purity, maximum matching, F-measure

Entropy-Based Measures

Conditional entropy, normalized mutual information (NMI), variation of information

Pair-wise measures

TP, FN, FP, TN

Correlation measures

Discretized Huber static, normalized discretized Huber static