Data Clustering & Classification

(Chapter 15)

Kai Goebel, Bill Cheetham

GE Corporate Research & Development

goebel@cs.rpi.edu

cheetham@cs.rpi.edu

Outline

- k-means
- Fuzzy c-means
- Mountain Clustering
- knn
- Fuzzy knn
- Hierarchical Methods
- Adaptive Clustering

Preliminaries

- Partitioning of data into several groups s.t. similarity within group is larger than that among groups
- Clustering ≠ Classification !
- Need similarity metric
- Need to normalize data
- Supervised vs. unsupervised clustering issues
  - Unsupervised: labeling cost high (large # of data, costly experiments, data not available, …)
- Understand internal distribution
- Preprocessing for classification

k-means Clustering

- Partitions data into k (or c) groups
- Finds cluster centers
- Dissimilarity measure: e.g., Euclidean distance

$$d^2(x_m, c_i) = \sqrt{\sum_{p=1}^{q} \left(x_{m_p} - c_i\right)^2}$$

s.t. cost function is minimized

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{m=1, x_m \in G_i}^{n} d^2(x_m, c_i) = \sum_{i=1}^{c} \sum_{m=1, x_m \in G_i}^{n} \sqrt{\sum_{p=1}^{q} \left(x_{m_p} - c_i\right)^2}$$

where
c=number of clusters
n=data within cluster
q=number of dimensions

Other Dissimilarity Measures

$$d(x,y) = |x-y|^T \sum\nolimits_{xy}^{-1} |x-y|$$

$$d(x,y) = \frac{x^T y}{x^T y + y^T y - y^T x} \quad \text{(binary x, y)}$$

$$d(x,y) = \frac{x^T y}{|x| \cdot |y|}$$

$$d(x,y) = \|x-y\|^m = \left[\sum_{i=1}^{N} (x_i - y_i)^m\right]^{\frac{1}{m}}$$

k-means Algorithm

- Initialize cluster centers (randomly?)
- Check whether data are closest to cluster center
- Compute cost function (stop if below threshold)
- Update cluster centers

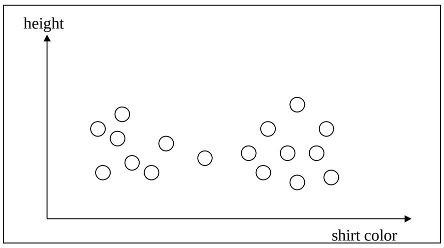
$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i}^{n} x_k \quad \text{(mean of all vectors in group i)}$$

where
$|G_i|$ is the number of elements in cluster G_i

- Go to second step

- Issues: selection of location and number of clusters

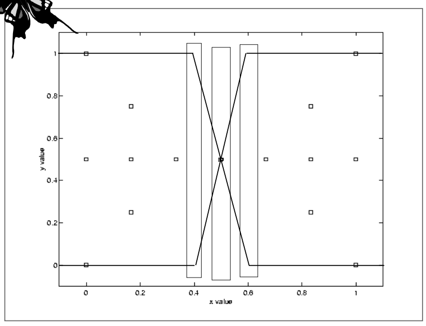
Cluster class exercise



Fuzzy C-means

- Allow points to belong partly to several clusters
- why?

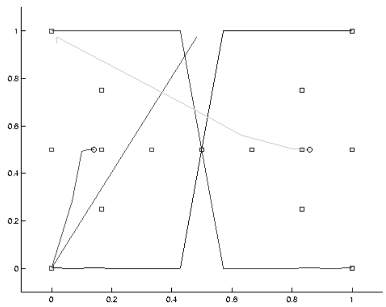
Butterfly Example



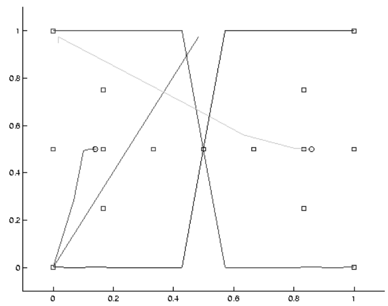
Fuzzy Clustering

- Formulate optimization problem $\min\ J(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m d^2(x_k, v_i)$
- Calculate membership values
$$u_{ik} = \frac{\left(\dfrac{1}{d^2(x_i, v_k)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c} \left(\dfrac{1}{d^2(x_j, v_k)}\right)^{\frac{1}{m-1}}}$$
- Update centroid
$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

Butterfly Simulation

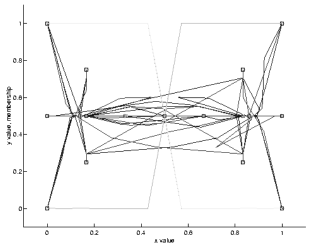


Powerpoint Butterfly Animation

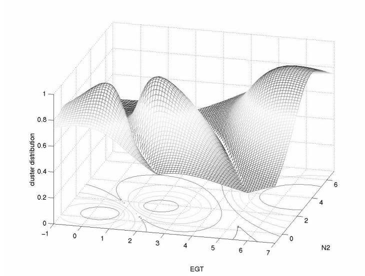


Many Simulations for Butterfly

- 50 Runs from random starting points
- centroids always settle at the same points



Fuzzy Cluster in 3-D Space



Mountain Clustering

- No need to set number of clusters a priori
- simple
- computationally expensive
- can be used to determine number of clusters for c-means

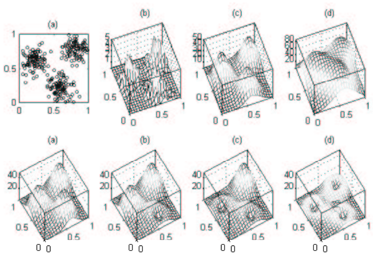
Mountain Clustering: steps

- - form a grid on the data space; intersections are candidates for cluster centers.
- - construct a mountain function representing data density
- $m(v) = \sum_{i=1}^{N} e^{-\left(\frac{\|v-x_i\|^2}{2\sigma^2}\right)}$ (each data point contributes to the height)
- -sequentially destruct the mountain function:
- make dent where highest value is

$$m_{new}(v) = m(v) - m(c_1)e^{-\left(\frac{\|v-c_i\|^2}{2\beta^2}\right)}$$

- subtracted amount inversely proportional to distance between v and c1 and height m(c1)

2D data for Mountain Clustering



- 1. mountain function with b) σ=0.02; c) 0.1; d) 0.2
- 2. destruction with β=1 b) first; c) second; d) third

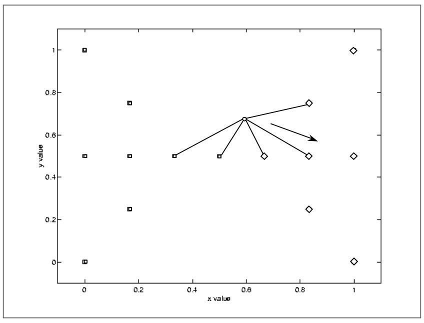
knn Algorithm

- Looks for k nearest neighbors (knn) to classify data
- Assigns class based on majority among knn
- Supervised method - needs labeled data for training

Crisp knn Algorithm

Compute distance from data point to labeled samples
If knn have not been found yet then
include data point
Else, if a labeled sample is closer to the data point than any other knn then
replace the farthest with the new one
Deal with ties
Repeat for the next labeled sample

Example



Fuzzy knn

- Assigns class membership
- Computationally simple
- Assign membership based on distance to knn and their memberships in classes

Fuzzy knn Algorithm

Compute distance from data point to labeled samples
If knn have not been found yet then
include data point
Else, if a labeled sample is closer to the data point than any other knn then
replace the farthest with the new one
Compute membership

(inverse of distances from nn and their class memberships)

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}\left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}}\right)}{\sum_{j=1}^{k}\left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}}\right)}$$

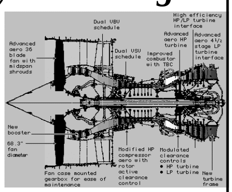
Repeat for the next labeled sample

Hierarchical Clustering

- Merge method:
- start with each x_i as a cluster
- merge the nearest pairs until #of clusters = 1

- Split method:
- start with # of clusters = 1
- split until predefined goal is reached

Classifying Turbine Anomalies with Adaptive Fuzzy Clustering



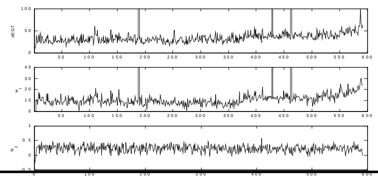
- Objective: track behavior of engines
  - measure system parameters
  - trend anal. for change detection detect changes
- Challenges:
  - large amounts of noise
  - changing operating conditions
    - **corrections with first principle models or regression models work only to some extent**
  - changes of schedules, maintenance, etc., which are not necessarily known to the analyst

Trending

- Monitor data and report observations indicative of abnormal conditions
- Issues:
  - definition of abnormality not crisp
    - step changes
    - different slope
    - combination of events
  - trade-off between false positives and false negatives
  - trade-off also with time to recognition
  - tool performance not the same for all conditions of interest
  - noise in data will influence quality of reporting

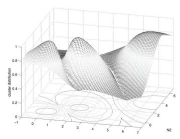
Trending of Shift Changes

- Problem:
  - detect abnormal shifts in process data
  - noise masks shifts
  - slow drifts are superimposed on data
- Solution:
  - Multivariate adaptive fuzzy clustering
  - Self-learning of "normal' drift
  - Capability to recognize abnormal changes



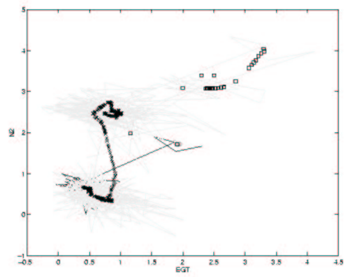
Shift Detection Tool

- Components:
  - Completeness checker and new data checker
  - Normalizer
  - Classifier (fuzzy knn)
  - Calculator of statistical propertiers (performs learning)
  - Adaptor (tracks normal wear)
  - Persistency checker (vigilance)
  - Composite alert score detector (generates alert)



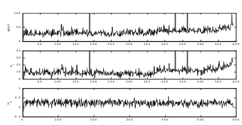
Cluster Adaptation

- tracks system changes

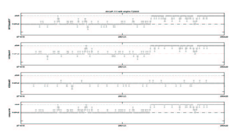


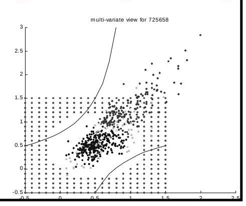
Example Data

- Raw Data Output Tool



Shift Detection





last slide