advanced cluster analysis

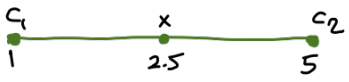
prototype-based=

hard (crisp) clustering= soft clustering with weight either 0 or 1

soft (fuzzy) clustering= allows point to belong to more than one cluster with different weights

↳ in K-means SSE error calculating with multiplication of the weights

↳ to minime SSE first fix centers then fix weights, recompute SSE



$SSE = w_{x1}^p (2.5-1)^2 + w_{x2}^p (5-2.5)^2 = 2.25\, w_{x1} + 6.25\, w_{x2}$

↳ min when $w_{x2}=0 \Rightarrow 2.25$

- summation of weights of a point must be add up to 1, fix all points weights accordingly

p= fuzzier (p>1) update centroids= $\Sigma wx / \Sigma w$

expectation-maximization algorithm= probability clustering, high complexity

↳ probability of a point belongs to clusters, uses gaussian distribution

↳ similar to K-means, but each step instead of computing the dist, compute probability

↳ weights are probabilities calcuted using bayes rule

self-organizing maps (SOM)= centroid based, fixed number of clusters

↳ like K-means, but centroids are updated based on their spatial proximity to the closest centroid

↳ can be viewed as dimensionality reduction

↳ high complexity and no guarantee to converge, no objective function

density-based=

grid-based clustering= assign objects to predefined grid cells, eliminate cells below treshold, form clusters from contiguous/adjacent cells

subspace clustering= considers subspaces of attributes or data

↳ clique= grid-based

↳ monotone property= if set of points cannot form a density based cluster, their all possible supersets cannot form either (like apriori algorithm)

↳ algorithm starts from one dimension to k-dim

↳ at each step eliminate cells that have points fewer than treshold

↳ complexity is exponential

graph based=

- edges are weights, single linkage hierarchical can be viewed as graph based

chameleon algorithm= dataset → proximity matrix (using KNN) → sparse graph → partition the graph

↳ sparsification= keeps the connections to the nearest while breaking the connections to furthers

↳ relative interconnectivity= interconnetivitiy of two clusters normalized by interconnectivity of the clusters. $RI = EC(C_j, C_i) \,/\, \frac{1}{2}(EC(C_j) + EC(C_i))$

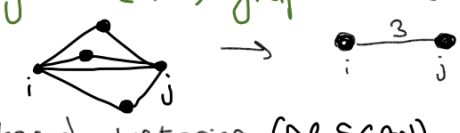
↳ relative closeness= absolute closeness of two clusters normilized by the internal closeness of the clusters RC

↳ merge clusters if RI and RC similar to original cluster

spectral clustering= partition the graph into components such that the nodes in a component are strongly connected while, weakly connected to the other componets

↳ can detect clusters with different sizes and shapes, but sensitive to outliers

shared nearest neighbor (SNN) graph= the weight of an edge is # shared nearest neighbors between vertices



- two points are similar, if they are similar to the same other points
- also uses density based clustering (DBSCAN)

jarvis-patrick clustering= uses SNN, first finds KNNs of points, then puts the points into same clusters if they have edges with the same more than some treshold vertices