Clustering

October 16, 2012

Basics - The Idea of Clustering

- clustering is *generalizing* over a similarity measure.
- elements have a similarity
  - between one another
  - between itself and not-elemental objects like for example the hypothetical average element (*centroid*).

We are looking for a partition that best groups similar elements and separates different elements.

- maximize *intra-cluster similarity*
- minimize *inter-cluster similarity*

Basics - Measuring Clustering Performance

- intra-cluster similartiy:
  - use an *element* $\times$ *element* matrix for the new cluster
  - enter the similarity for each $\langle element, element \rangle$ pair
  - sum over all values in the matrix, devide it by the number of edges and get the overall similarity:
- inter-cluster similarity:
  - generate a hypothetical average element (*centroid*) for each new cluster
  - measure similarity between the new clusters' representatives
  - use the similarity between the most similar (*single-link* similarity function) $\langle c_1, c_2 \rangle$ pair
  - use the similarity between the most dissimilar (*complete-link* similarity function) $\langle c_1, c_2 \rangle$ pair

Basics - Measuring Clustering Performance

Instead of thinking of the inter-cluster similarity:

- measure overall similarity in the set of all new clusters
- clusters should have high similarity in comparison to the overall similarity
- maximize $p_3 = \frac{sim(A)+sim(B)}{sim(A \cup B)}$
- but this is best when each element has its own cluster! see ending conditions

Unless we are already on the bottom level, there is always a partition that satisfies the *monotonicity criterion*, that is $p \geq 1$. Another way of describing the goal of clustering is to maximize the mutual information.

- several standart similarity or distance measures
  - *euclidean distance* in vector space
  - *jaccard coefficient* for sets
  - ...
- applied on AVM's representing the elements to be clustered
- decide which features to use as attributes and how to derive values for them!
- possible to learn this from a training set of already clustered elements

$$\left\langle \begin{array}{ll} attribute_1 & v_1 \\ attribute_2 & v_2 \\ \vdots & \vdots \\ attribute_n & v_n \end{array} \right\rangle$$

Figure: An *attribute value matrix* (*AVM*)

Basics - Similarity / Distance

- relation between similarity and difference is opposed
- we can turn any similarity measure into a distance measure: $\frac{1}{1+sim}$ and the other way round: $\frac{1}{1+dist}$
- as similarity increases, distance decreases; as distance increases, similarity decreases
- similarity between two identical elements is maximal and 1. The minimum similarity is 0.

Basics - Soft- vs. Hard Clustering

- in soft clustering, an element can belong to more than one cluster
- it is even possible to assign a degree of belonging to each $\langle element, cluster \rangle$ pair
- this is not allowed in hard clustering

$$\left\langle \begin{array}{cccc} & e_1 & \ldots & e_n \\ c_1 & 0.2 & \ldots & 0.08 \\ \vdots & \vdots & \ddots & \vdots \\ c_n & 0.01 & \ldots & 0.3 \end{array} \right\rangle, \left\langle \begin{array}{ccccc} & e_1 & e_2 & \ldots & e_n \\ c_1 & 1 & 0 & \ldots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & 0 & 1 & \ldots & 1 \end{array} \right\rangle,$$

$$\left\langle \begin{array}{ccccc} & e_1 & e_2 & \ldots & e_n \\ c_1 & 1 & 0 & \ldots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & 0 & 1 & \ldots & 0 \end{array} \right\rangle \Rightarrow f = \{\langle e_1, c_1 \rangle, \langle e_2, c_n \rangle, \langle e_n, c_1 \rangle\}$$

Basics - Coherence and the MST

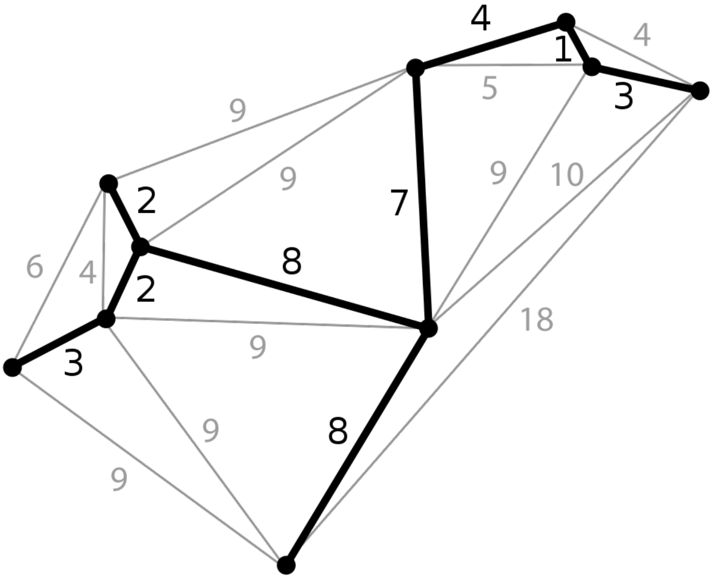
- project a *Minimal Spanning Tree* (*MST*) on set to be clustered
- MST combines all nodes with smallest possible overall edge length

The coherence of a cluster reflects the case that the most distinct element in the cluster would be a separate cluster and the inter-cluster similarity of the two clusters would be computed.

Basics - Coherence and the MST

- *single-link measure*: coherence of a cluster is the smallest similarity between two nodes in the MST
- complete-link: coherence is the smallest similarity of all $\langle element, element \rangle$ pairs in the cluster
- *group-average* measure: coherence is the average similarity of all the pair-similarities

Basics - Coherence and the MST



Basics - Ending Conditions

- hierarchical clustering needs no ending condition
- for flat clustering we need to determine a maximum number of clusters
- else it will split into separate clusters for each single element
- another possibility is to use *Minimal Description Length* (*MST*)

Basics - Reallocations

- some clustering algorithms perform reallocations during runtime
- a cluster is not clearly assigned to one cluster after some iteration
- it might be reassigned to another cluster lateron

Basics - Medoid / Centroid

- the centroid in vector space is a imaginary element that is not in the set of elements
- it is projected into the vector space by taking the average of all values for all attributest
- the medoid is the element in the set that is closest to the centroid
- some algorithms use the medoid instead of the centroid as the center of a cluster

$$e_1 = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}, e_2 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, e_3 = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix};$$

$$e_1 + e_2 + e_3 = \begin{pmatrix} 7 \\ 6 \\ 6 \end{pmatrix} \Rightarrow centroid = \begin{pmatrix} \frac{7}{3} \\ \frac{6}{3} \\ \frac{6}{3} \end{pmatrix} = \begin{pmatrix} 2,\overline{3} \\ 2 \\ 2 \end{pmatrix}$$

Basics - Clustering vs. Classification

- clustering uses a similarity measure to compare elements with elements
- derives a structural ordering by itself (*unsupervised learning*)
- classification uses a similarity measure to compare elements with already existing patterns
- patterns are defined in advance for specific groups (*supervised learning*)

Hierarchical Clustering

We can think of the hierarchical clustering process in two ways:

1. iteratively *separating* clusters top-down starting with an initial Hyper-Cluster (*Divisive Clustering*)
2. iteratively *grouping* bottom-up from initial 1-elemental clusters (*Agglomerative Clustering*).

Hierarchical Clustering

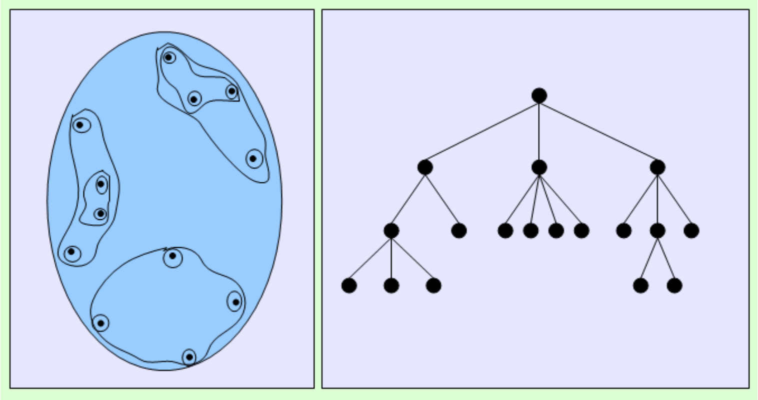


Figure: A hierarchical clustering

Hierarchical Clustering - Top Down

- a cluster is iteratively devided into sub-clusters
- similarity between the evolving clusters is minimized
- similarity between the elements within each of the clusters is maximized
- best partition is the one with the best inter-intra-similarity ratio
- maximize $p_1 = \frac{intra-sim}{inter-sim}$ or minimize $p_2 = \frac{inter-sim}{intra-sim}$, with p_1, p_2 partition quality measures

Hierarchical Clustering - Top Down

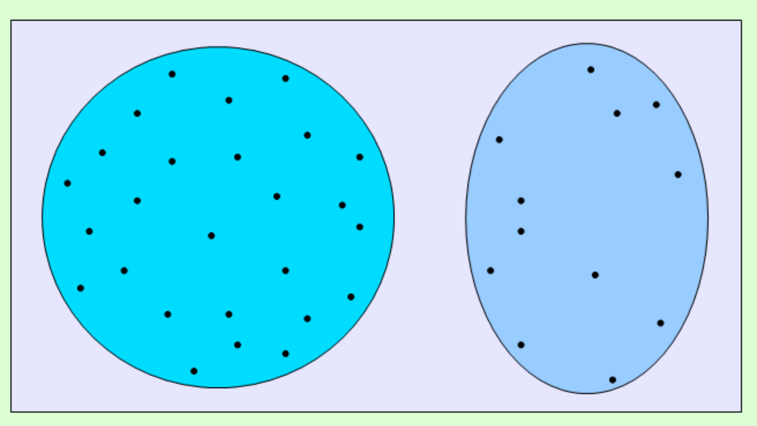


Figure: $p_3 = \frac{sim(A)+sim(B)}{sim(A \cup B)}$

Hierarchical Clustering - Bottom Up

- in Agglomerative Clustering we start with seed clusters
- in each agglomeration step we add one external element or cluster to each cluster
- maximize similarity between each of the combined pairs
- if a cluster c is to be merged with another cluster:
  - choose the cluster that will lead to the greatest intra-cluster similarity after merging
  - of all the possible $\langle c, c' \rangle$ pairs, we are looking for the one that maximizes $sim(c \cup c')$

Flat Clustering

- a *Flat Clustering* does not result in a hierarchy
- most popular algorithms for flat clustering are:
  - *k-means*
    depends heavily on the notion of the centroid or medoid
  - *Expectation Maximization* (*EM*)
    uses *statistics* (!) to calculate the cluster model that maximizes the likelihood of the data

Flat Clustering - K-means

- start with k seed "clusterpoints"
- can be set randomly or automatically or manually
- results will vary depending on where the initial "clusterpoints" were placed
- repeat until an ending condition is reached:
    1. Assign each element to the closest clusterpoint
    2. Move the clusterpoint into the actual center of the cluster

Flat Clustering - K-means

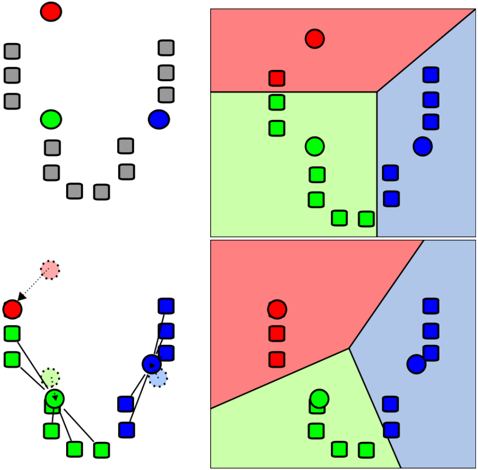


Figure: An illustration of the k-means algorithm