Parameterless clustering by dynamic tree-cutting

Simon Lehmann Knudsen
simkn15@student.sdu.dk
ECTS: 10
04/09-2017 - 31/01-2018
5th semester in Computer Science

31/01-2018

Contents

1 Where to introduce the dynamic cutting tree approach?(Currently brief description in section hierarchical clustering)

2 Abstract

2.1 Background

2.2 Results

2.3 Conclusion

3 Introduction

Clustering, or cluster analysis, is a way of grouping a set of objects such that a cluster of objects is more similar to each other than those of another cluster. Clustering can help with the description of patterns of similarities and differences in a data set. There are many different tools to do cluster analysis, which all intend to find an optimal clustering depending on a set of criteria. Given a set of criteria and parameters, we can analyze the data and discover the clusters. The resulting clusters can all be either feasible or infeasible. In some circumstances, even overlapping clusters can provide acceptable solutions. When there are no clear separation of clusters, it can be hard to determine if the solution is acceptable or not.
Figure 1 shows somewhat similar datasets where the sepration of the clusters are decreasing. `g2-2-10` and `[g2-2-30]` has a clear separation. In `g2-2-50` we can still see a little difference in the density around the center, showing a small separation. There are no longer a separation in `g2-2-70` and it is difficult to say how many clusters the optimal solution has. Clustering requires a understanding of the dataset, and depending on the data, the solution could, e.g., have 10 clusters to be optimal, like the clustering in figure 2.

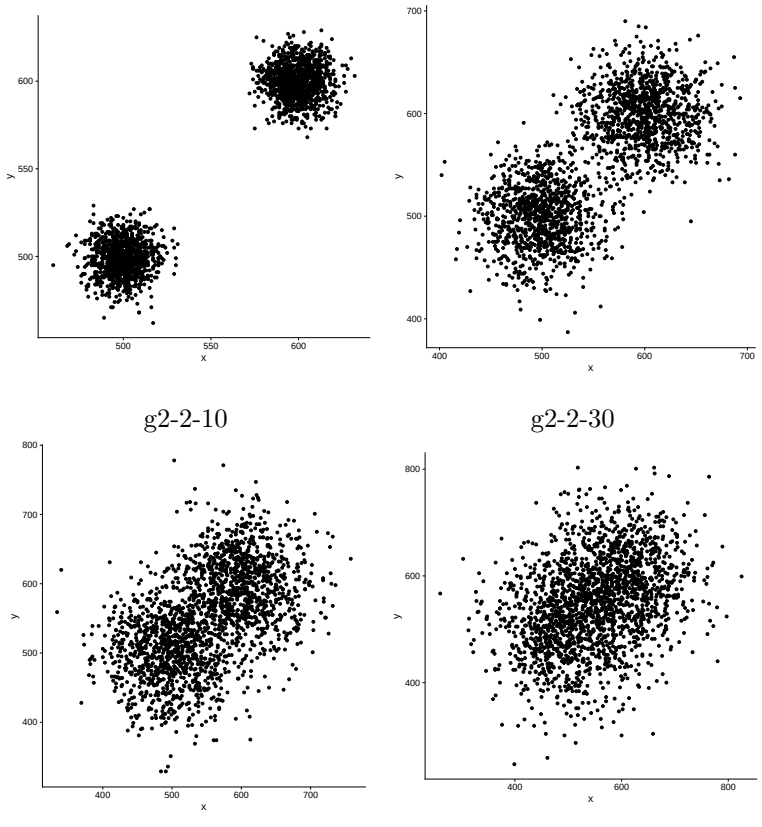

Figure 1: URL: https://cs.joensuu.fi/sipu/datasets/

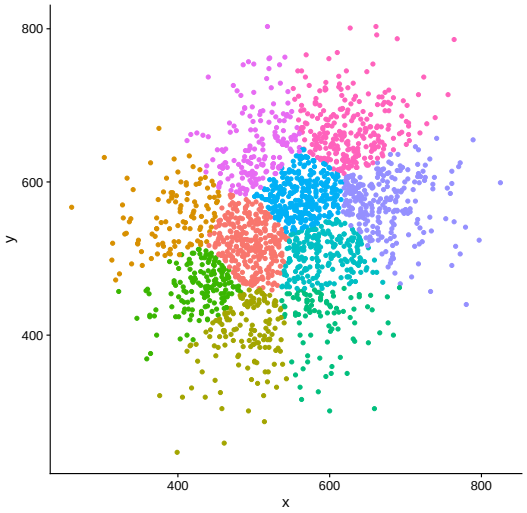

Figure 2: g2-2-70 clustering with 10 clusters

The basic data for a cluster analysis starts with a $n \cdot p$ multivariate data matrix, `X`, which describes each object to be clustered. The entry x_{ij} gives the value of the `j`th variable on object `i`:

$$\mathtt{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & \dots & x_{np} \end{bmatrix}$$

Figure 3: Multivariate data matrix

Many clustering techniques begins by converting `X` into a symmetric $n \cdot n$ matrix. Both rows and columns represents the objects in the data set. The resulting matrix could be of similarities, dissimilarities or distances between all objects.

The formal definition of a cluster can be difficult to give, and it is not clear how a cluster is recognized when displayed in the plane. Looking at figure 4 we have a lot of different types of data sets, which does not have much in common in terms of shapes. Looking at the "two" clusters to the right in the `Aggregation` set, intuitively these are in fact two clusters, but in some situations it would be more relevant to see them as a single cluster, since they are linked together by a few objects. Looking at the `path-based2:spiral` set, the clusters are linked together by the neighbors of an object. Dependent on what we are looking for it could be that the objects closest to the center should be in the same cluster. In terms of the `Zahn's Compound` set it can be hard to determine the outliers. The objects in the lower left corner would intuitively be two clusters. Or, the objects in the middle as a cluster and the ring of objects as outliers. With the

two circular clusters(upper left corner) and the square cluster(right side), the clusters could be the dense area around the centers, and outliers around them. Also, the outliers could be members of the clusters.

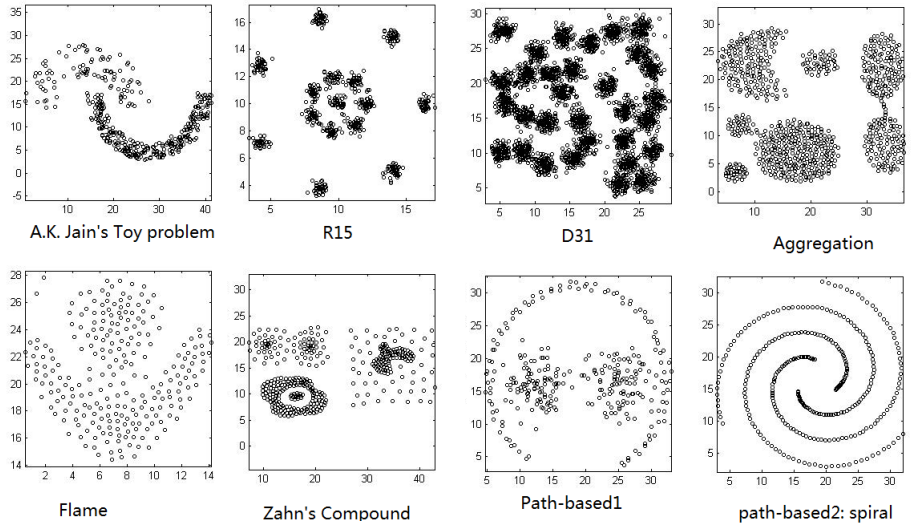

Figure 4: Shape Data sets from URL: https://cs.joensuu.fi/sipu/datasets/

Assessing the quality of a clustering can be a challenging task. Clustering is Unsupervised Learning where it is unknown a priori what the best clustering looks like. Many validation techniques requires a `gold standard` in order to measure the quality. A `gold standard` to a given dataset is a solution developed by experts. Quality measures that requires a `gold standard` are Rand Measure, F-measure, Jaccard Index and Dice Index to name a few. One that does not need a gold standard is Sum of Squares which calculates the sum of distances of all objects in a cluster to its centroid. A low value indicates a good clustering, and a high value indicates bad clustering. One problem with Sum of Squares is that a clustering with all objects as it own cluster(singleton clusters) would have a Sum of Squares of 0. Thus, theoretically imply a good clustering, but rarely, if not none, would this ever give any new information.

There are many different algorithms for cluster analysis and which one to use is highly dependent on the dataset. Some of the more known algorithms are `k-means`, `hierarchical clustering`, `Shared Nearest Neighbor`, `DBSCAN`. `k-means` is the de-facto standard algorithm for clustering, as it generally performs well. All algorithms takes a set of parameters in order to do the clustering. `k-means` has one, where `DBSCAN` has two. Parameter `k` in `k-means` is the number of clusters it should return. It can be difficult to determine a single parameter, and increasing the number of parameters exponentially increases the difficulty of setting the parameters. Although, calculating the `F-ratio` can give a good indication of an optimal `k` for `k-means`. In this project we chose `Transitivity Clustering(TC)`, based on the `Weighted Transitivity Graph Projection Problem`. F-measure will be used as quality measure. TC and F-measure will be discussed later.

4 Background

4.1 Transitivity Clustering

Before going into any details about Transitivity Clustering(TC) we need some basic graph-theoretic definitions.

Definitions from 'Extension and Robustness of Transitivity Clustering for Protein...'

Definition 1 (`Undirected simple graph`)**.** An `undirected simple graph` $G = (V, E)$ consists of a set of nodes V and a set of edges $E \subseteq \binom{V}{2}$, where $\binom{V}{2}$ denotes the set of two-element subsets of V. The edges are undirected and contains no self-loops or multiple edges between two nodes. uv is an unordered par $\{u, v\} \in \binom{V}{2}$.

Definition 2 (`Transitive graph`)**.** An undirected simple graph $G = (V, E)$ is called `transitive`

$$\text{if for all triples } uvw \in \binom{V}{3}, uv \in E \text{ and } vw \in E \text{ implies } uw \in E.$$

Definition 3 (`Weighted Transitive Graph Projection Problem(WTGPP)`)**.** Given a set of objects `V`, a threshold $t \in \mathbb{R}$, and a pairwise similarity function sim: $\binom{V}{2} \to \mathbb{R}$, the graph G is defined as

$$G = (V, E);\ E = \left\{ uv \in \binom{V}{2} : \text{sim}(uv) > t \right\} \tag{1}$$

The WTGPP is the determination of a transitive graph $G' = (V, E')$ such that there exist no other transitive graph $G'' = (V, E'')$ with $\text{cost}(G \to G'') < \text{cost}(G \to G')$. The modification costs are defined as

$$\text{cost}(G \to G') := \underbrace{\sum_{uv \in E \setminus E'} |\text{sim}(uv) - t|}_{\text{deletion cost}} + \underbrace{\sum_{uv \in E' \setminus E} |\text{sim}(uv) - t|}_{\text{addition cost}} \tag{2}$$

Transitivity Clustering takes one parameter, t, which is the threshold for similarities. Following is the steps in Transitivity Clustering:
Reference: Comprehensive cluster analysis with Transitivity Clustering

1. Model the given pairwise similarity, from the similarity matrix, as a similarity graph, G. The nodes corresponds to the objects, with weighted edges as the similarity values.

2. Transform the similarity graph, G, into another graph, G', by subtracting the threshold from the edge weights. Subsequently removing those edges with weights below zero, which is the deletion cost for equation 2.

3. Transform G' into a transitive graph, G'', with minimal cost. Thus, in this step we add all edges such that the graph is transitive, which is the addition cost for equation 2.

The resulting transitive graph, G'', is the clustering solution.

4.2 Hierarchical Clustering

Definition 4 (`Hierarchical Clustering(HC)`)**. Reference from Unsupervised Learning slides: Hierarchical Clustering**
Builds a nested structural partition $C = \{C_1, \ldots, C_k\}$ of V such that $\cup_{i=1}^{k} C_i = V$ and $C_i \neq \emptyset \ \forall i \in \{1, \ldots, k\}$. $\forall$ pairs C_i, C_j where $i, j \in \{1, \ldots, k\}, i \neq j$, exactly one of the following holds

- $C_i \cap C_j = \emptyset$
- $C_i \subset C_j$
- $C_j \subset C_i$

There are two forms of hierarchical Clustering, `agglomerative` and `divisive`. Agglomerate starts with n clusters, where n is the number of objects in the dataset. Joining clusters until one cluster is remaining, where all n objects are members. Divisive is the opposite. Starting with one cluster with n objects. Splitting the clusters until all clusters are singletons. Thus, all n objects represents are cluster. Agglomerative is the most used, as divisive usually is a more expensive procedure. One important feature of hierarchical clustering is that a join or split of clusters are irrevocable, thus cannot be undone. Figure 5 shows an overview of agglomerative vs. divisive.

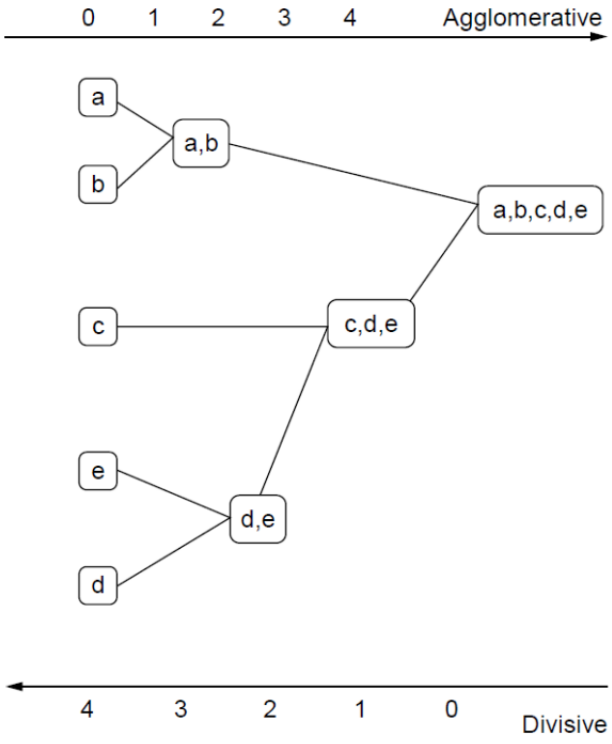

Figure 5: Agglomerative vs. Divisive

The steps of joins or splits is often showed as a `dendrogram`, which is viewed as a tree structure. The root of the tree is the cluster containing all n objects.

Moving down the tree the nodes represents the clusters which was split from its parent. At the bottom of the tree we have the leafs, where the number of leafs represents all singleton clusters(the n objects). The tree can be cut a given height resulting in a clustering solution. Figure 6 shows a `dendrogram` of the tree structure of a HC. The horizontal axis shows all the objects in the dataset. The vertical axis shows the distances between objects and/or clusters. Objects 1 and 2 are joined to a cluster at height 2. These are joined with the remaining objects at height 4, resulting in one cluster holding all objects.

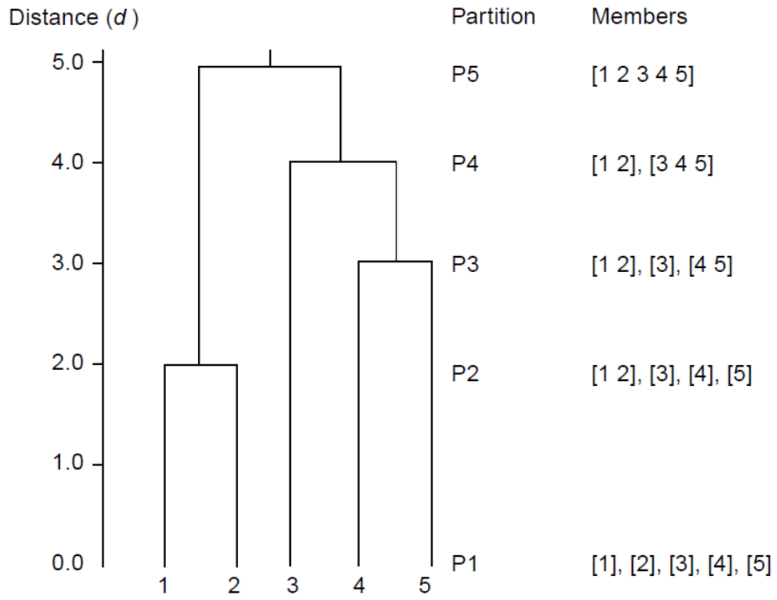

Figure 6: Dendrogram of a Hierarchical Clustering

In a typical hierarchical Clustering the size and number of clusters are given by a threshold, much like the one used in `TC`. Meaning that one iteration can possibly affect all clusters, which either increases or decreases the overall quality. In Figure 6 the tree could be cut between height 3-7.5 to obtain a solution containing two clusters.

Might not be the right section explaining about cutting the tree to gain optimal solution(Dynamic tree cut)

In Figure 7 we could cut the tree between height 3-7.5 and we would get a solution containing two clusters. One issue in terms of quality is that, the optimal clustering solution could have the clusters 5, 2, 3, 6, 1, 7 and 4, 0. Making a horizontal cut in the tree, it would be impossible to obtain this solution. However, if it was possible to make a dynamic tree cut, the optimal solution could in fact be obtained. The solution can be made by cutting the tree in three different heights. First cut is between 2.5 and 7.5 to obtain the cluster 5, 1. Cutting between height 2.0 and 3.0, obtaining cluster 3, 6, 1. The last cut is between 2.5 and 3, giving the last clusters 7 and 4, 0.

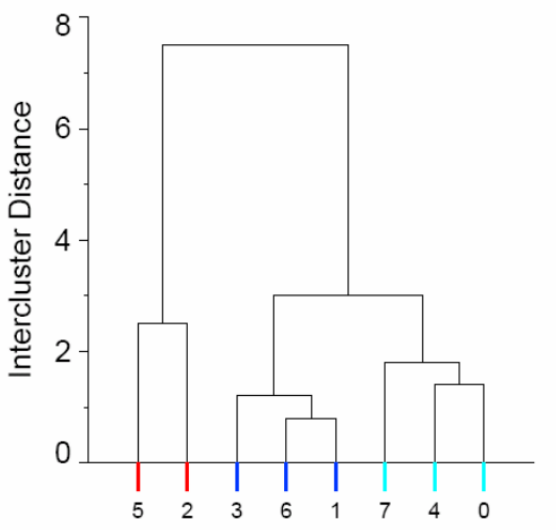

Figure 7

4.3 Cluster Validation

4.4 Multidimensional Scaling

4.5 GAP Statistics

5 Method

5.1 Assessing best clustering

5.2 Randomization approach 1

5.3 Randomization approach 2

5.4 Randomization approach 3

5.5 Randomization approach 4

5.6 Results for randomizations

5.7 Short about the implementation / tools

6 Conclusion

7 References