Clustering for Graphs

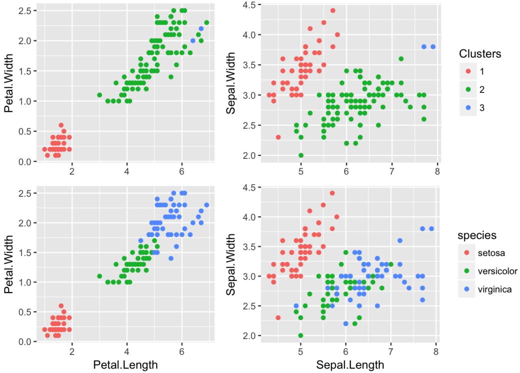
Laurenz Tomandl

January 8, 2024

TU WIEN
ALGORITHMS AND COMPLEXITY GROUP

What is Clustering

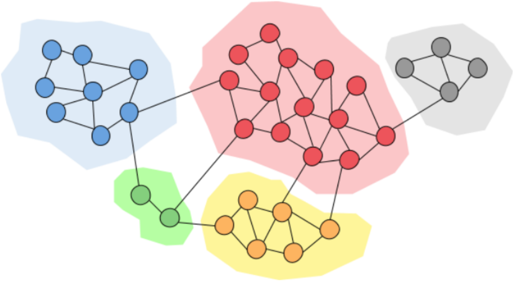
- Find points or Elements in Data/Graph that are similar to each other



rpubs.com/PunkFood_Disme/iris_clustering

What is Clustering

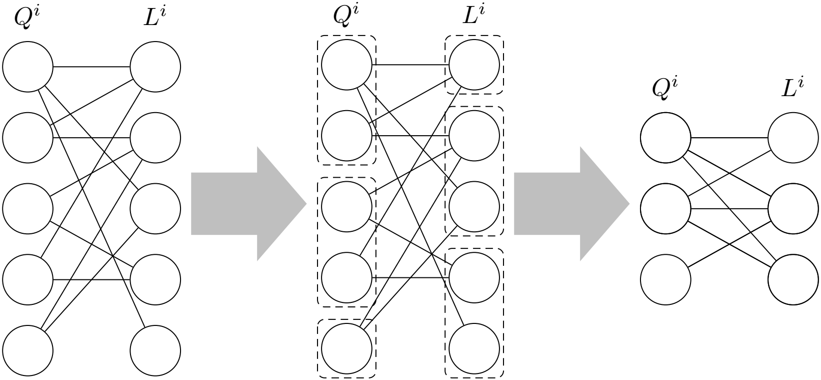
- Find points or Elements in Data/Graph that are similar to each other



towardsdatascience.com/community-detection-algorithms

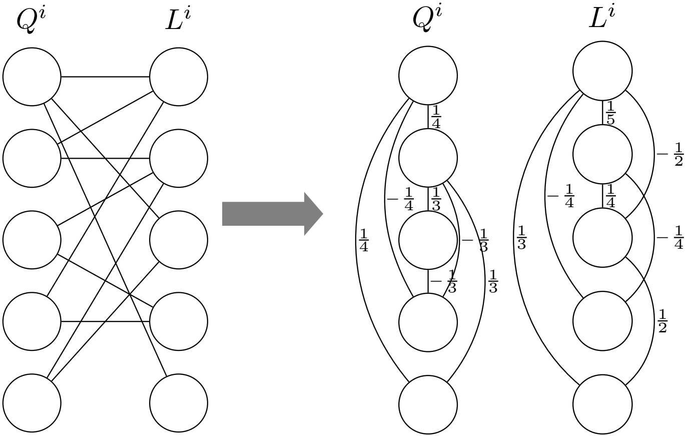
Clustering for Combinatorial Optimization Problems

- Transform the problem into a similar smaller problem



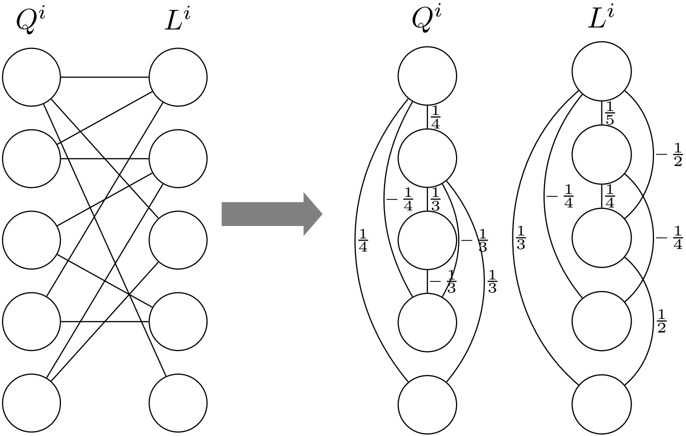
Clustering for Combinatorial Optimization Problems

- Problem defined on a bipartite Graph
- Transformed into problem on two separate Graphs
- weights represent the expected error when merging



Graph properties

- not-metric
- negative weights, may include negative circles: Solution weight transformation
- Ideas are e^x or $x < 0 \rightarrow x = 0$



Graph Partitioning Algorithms

- k-means (for metric graphs)
- agglomerative hierachical clustering
- PAM (Partitioning Around Medoids)
- CLARA (Clustering LARge Applications)
- CLARANS (Clustering Large Applications based on RANdomized Search)
- MST Clustering
- Spectral Clustering
- Graph Auto Encoders (GAE)

Clarification: many of these algorithms require a complete distance matrix with runtime $O(n^2 \log(n))$ which is a bottleneck

k-means

- Mostly used for arbitrary data not graph data
- k-means requires distance calculation from nodes to arbitrary points

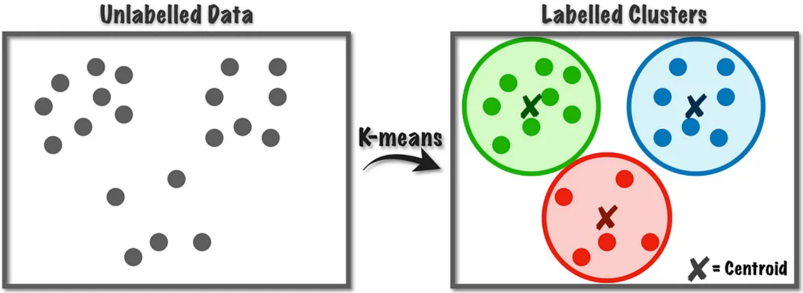


Figure: towardsdatascience.com/k-means-a-complete-introduction

agglomerative hierachical clustering

- Iteratively cluster node pairs
- Clustered nodes have new similarity to neighbors
- Different linkage variants
- Works on non metric graphs because we can use the length of a path between two nodes as distance

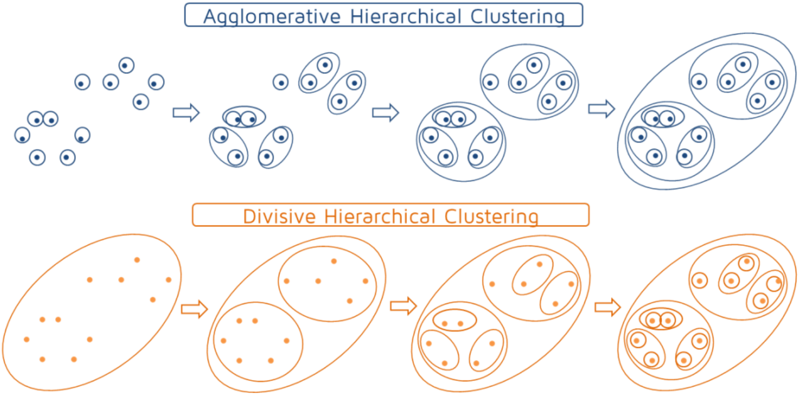


Figure: https://quantdare.com/hierarchical-clustering/

agglomerative hierachical clustering

- Single Linkage: Take the minimum distance to the cluster
- Complete Linkage: Take the maximum distance to the cluster
- Average Linkage: Take the mean distance to all nodes
- Other: e.g. Ward Linkage

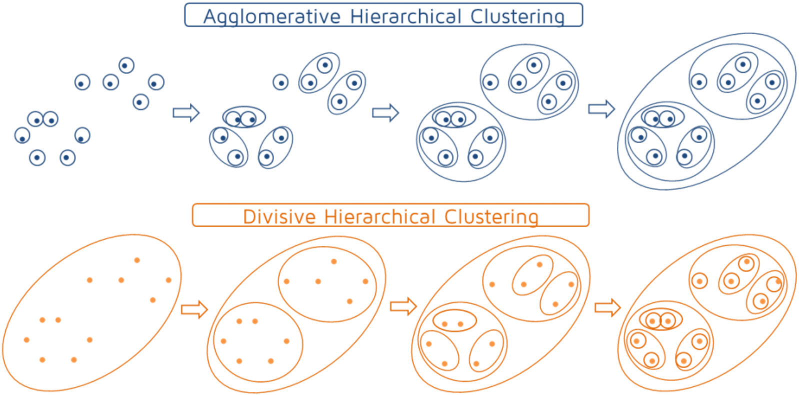


Figure: https://quantdare.com/hierarchical-clustering/

PAM (Partitioning around Medoids)

- Two Phases: Build and Swap
- Build Phase: Greedily Select Cluster Center such that distance to all other nodes is minimal
- Swap Phase: Use local search. Swap one medoid with one non medoid which improves the sum of distances the most

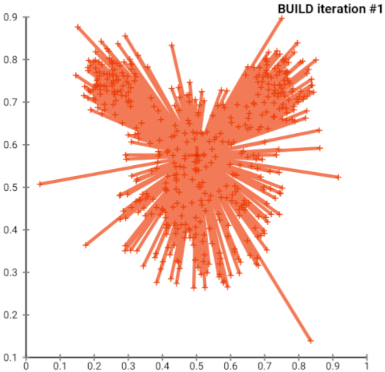


Figure: wikipedia.org/K-medoids

PAM (Partitioning around Medoids)

- node assigned to cluster around medoid with shortest distance
- Disadvantage over agglomerative clustering: No control over cluster sizes.
- FastPAM runtime: $O(|V|^2)$

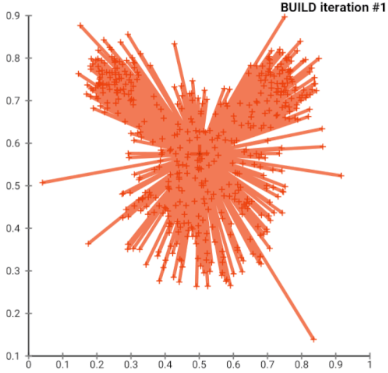


Figure: wikipedia.org/K-medoids

CLARA (Cluster LARge Applications)

- Create *n* samples of size *m* of the original graph
- apply PAM to those *n* samples
- Compare the found medoids in those *n* samples on the whole graph and choose the best.

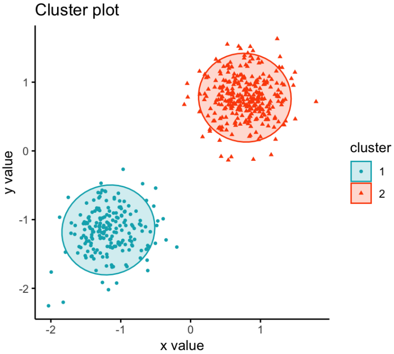


Figure: datanovia.com/clara-in-r-clustering-large-applications/

CLARANS (Clustering Large Applications based on RANdomized Search)

- Based on local search with first improvement
- Randomly choose k medoids
- check n random neighbors if they yield an improvement
- After n steps compare found optimum to global optimum
- Repeat m times

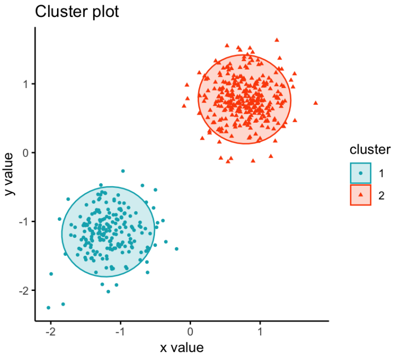


Figure: datanovia.com/clara-in-r-clustering-large-applications/

MST-Clustering

- Calculate an MST of the graph
- Iteratively add edges of MST until k disjoint clusters are created

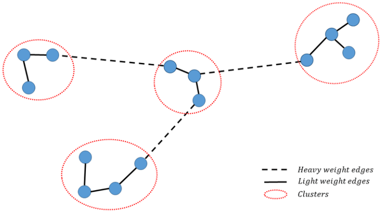


Figure: Minimum spanning tree release under differential privacy constraints

Spectral Clustering

- Based on Eigenvector analysis of Graph Laplacian
- Often combined with k-means analysis
- Doesn't scale well for many clusters
- Requires strictly positive weights

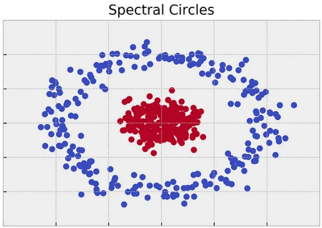


Figure: https://towardsdatascience.com/spectral-clustering

Auto Encoders

- AutoEncoders are Neural Network Structures that learn a low level Representation of the input
- Encoder encodes a low level representation of the input. Decoder reconstructs the original input from the low level representation.
- Idea is to transform high level structure into a low level representation, then cluster the low level representation using e.g. k-means, then decode the structure

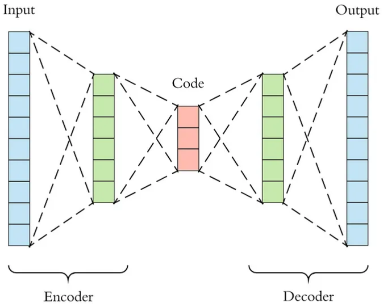


Figure: towardsdatascience.com/generating-images-with-autoencoders

Graph Auto Encoders

- Similar GAE can learn a low level graph representation of an input graph
- How to cluster nodes in the low level representation?

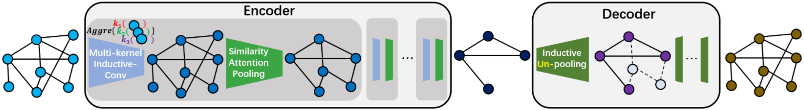


Figure: Graph Autoencoder for Graph Compression and Representation Learning

Problems that I am trying to solve

- Negative weight transformation $\rightarrow$ trial and error
- Restricting Cluster Sizes $\rightarrow$ Iterative application of an algorithm on too large Clusters