Graph Clustering

Shruti Sureshan
M21CS015

Dhriti Prasanna Paul
M21CS056

Indian Institute of Technology Jodhpur

Introduction

- Graph clustering is a way of grouping data by forms of graph in a way that the elements are related by some similarity measure

- Graph clustering falls under the category of graph mining algorithms

- A graph is clustered in one of the two ways, either by clustering its nodes by their edges, or by their edges’ distances.

Introduction

Graph Clustering Algorithms:

- Minimum spanning tree based Clustering
- Hierarchical Agglomerative Clustering
- Markov Clustering Algorithm
- Spectral Clustering
- Shared Nearest Neighbor (SNN) Clustering
- Betweenness Centrality
- Maximal Clique Enumeration
- Kernel k-means clustering

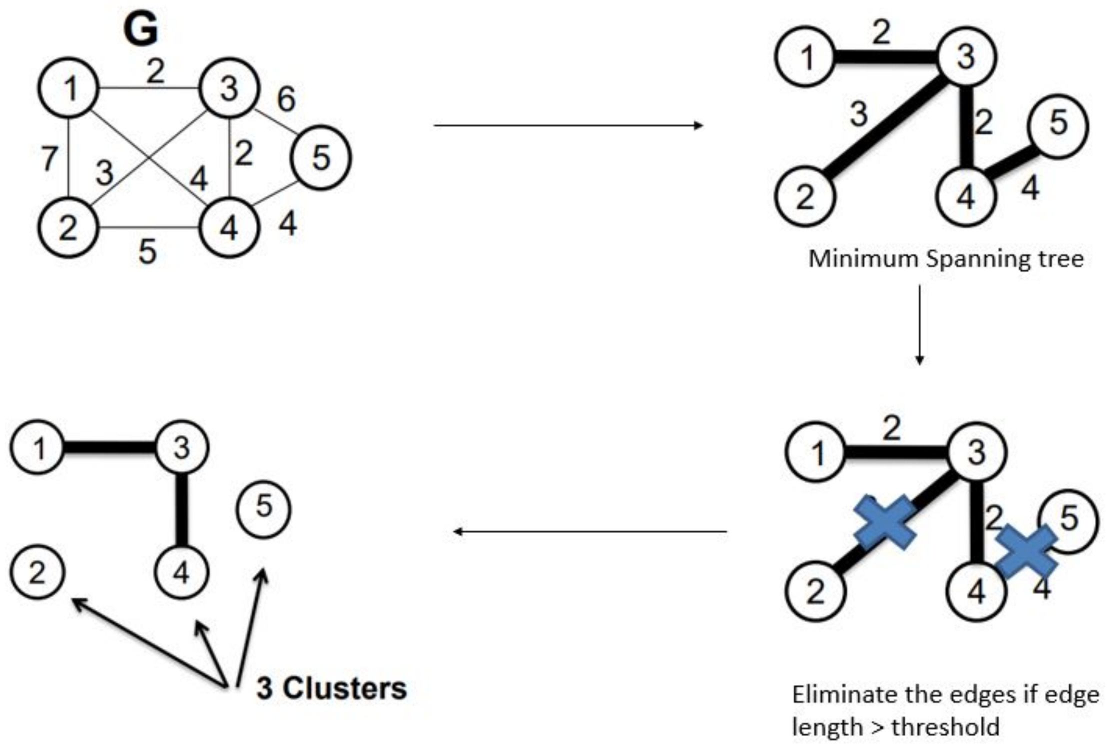
Minimum Spanning Tree based Clustering

- Apply Kruskal’s algorithm to construct a Minimum Spanning Tree
- Compute the threshold and the step size
- If edge length > threshold then eliminate it
- Compute the ratio between the distance of the intra-cluster and the distance of the inter-cluster

Minimum Spanning Tree based Clustering

- Advance the step size and thus update the computed threshold and repeat the same steps again in a loop
- When no MST edges can be removed and the threshold is highest, we stop the loop
- Thus, the lowest Intra-inter ratio to form the clusters based on the threshold to get the optimum threshold is computed.

Minimum Spanning Tree based Clustering



Minimum Spanning Tree based Clustering

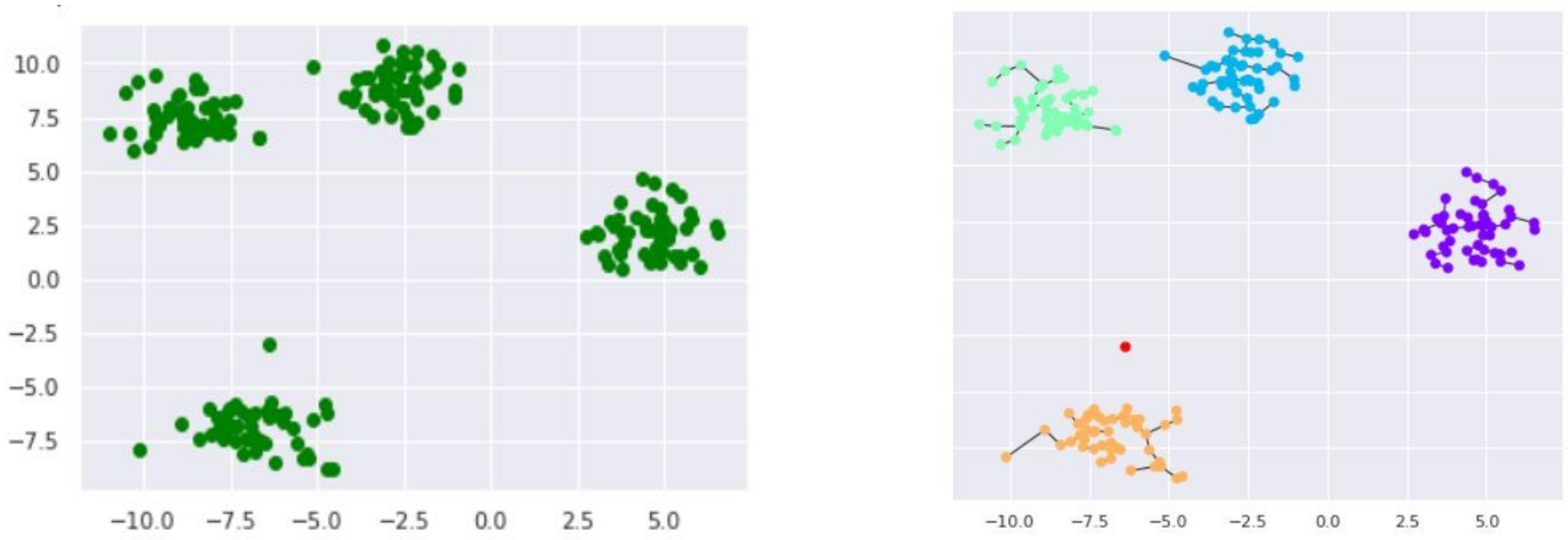
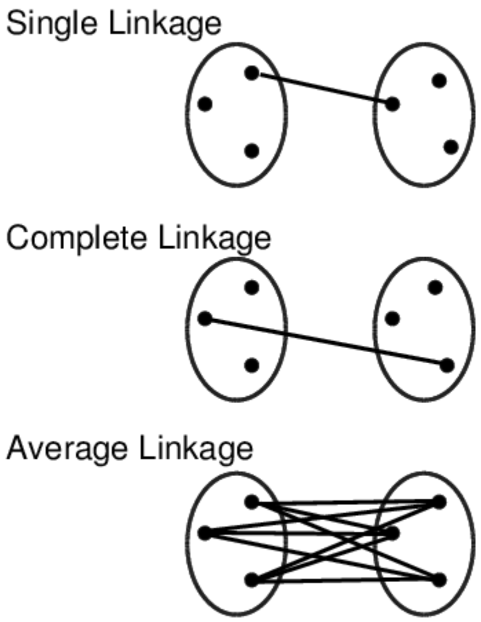


Fig: MST based Clustering on a sample data

Hierarchical Agglomerative Clustering

- Begin with small groups and then combine them to form larger clusters

- Type of linkages:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Ward Linkage



Hierarchical Agglomerative Clustering

- To get an idea about the actual path, create its dendrogram
- Dendrogram gives an idea about the similarity level among the different points
- Thus, it provides a visibility on the closeness of the data points via construction of dendrograms

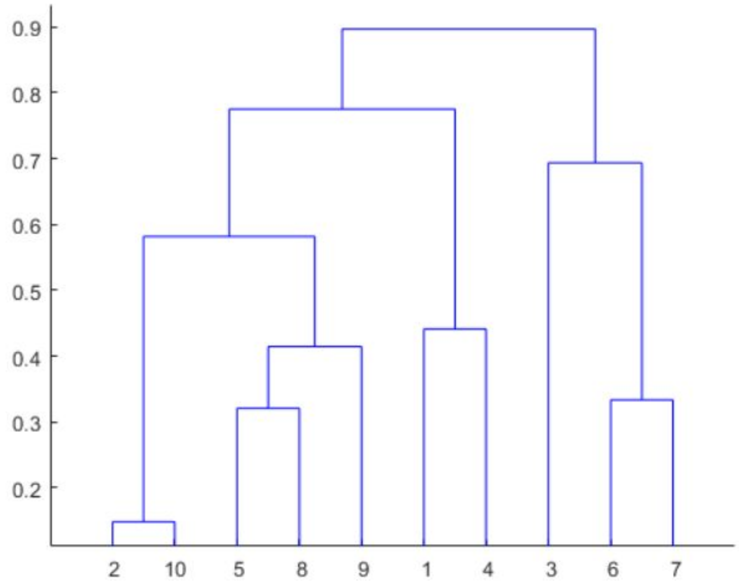


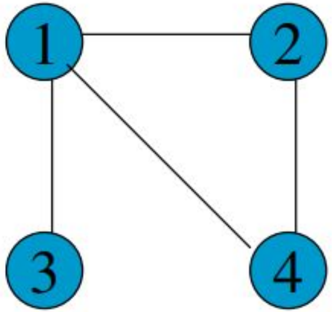
Fig: Dendrogram

Markov Clustering Algorithm

- It is based on Random walks

- In this algorithm, we follow two basic processes: Expansion and Inflation

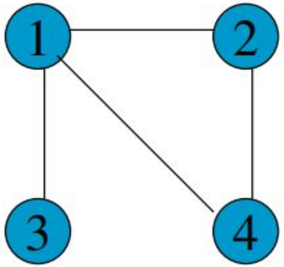
- The input to this algorithm is a power parameter e, an inflation parameter r and an undirected graph G



Power of 2
Inflation of 2

Markov Clustering Algorithm

- Compute an associated matrix depending on the associativity of the nodes



$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- Include the self loops (optional)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Markov Clustering Algorithm

- Matrix Normalization

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 \\ 1/4 & 0 & 1/2 & 0 \\ 1/4 & 1/3 & 0 & 1/3 \end{pmatrix}$$

Markov Clustering Algorithm

- Expand by including the matrix to the power of e and then using the r parameter, compute the inflation of the matrix and again repeat this step till convergence

$$
\begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 \\ 1/4 & 0 & 1/2 & 0 \\ 1/4 & 1/3 & 0 & 1/3 \end{pmatrix}^2
\overset{\substack{\text{Power of 2} \\ \text{Inflation of 2}}}{=}
\begin{pmatrix} .35 & .31 & .38 & .31 \\ .23 & .31 & .13 & .31 \\ .19 & .08 & .38 & .08 \\ .23 & .31 & .13 & .31 \end{pmatrix}^2
\overset{\substack{\text{Power of 2} \\ \text{Inflation of 2}}}{=} \cdots\cdots\cdots\cdots =
\begin{pmatrix} 1 & .33 & .50 & .33 \\ \text{--} & .33 & \text{--} & .33 \\ \text{--} & \text{--} & .50 & \text{--} \\ \text{--} & .33 & \text{--} & .33 \end{pmatrix}
$$

- The final matrix obtained can be useful to find new clusters

Markov Clustering Algorithm

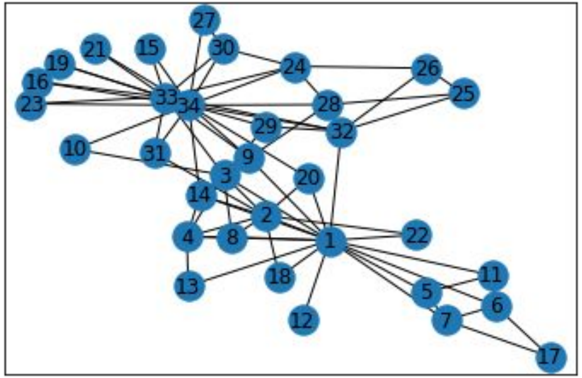


Fig: Input graph data

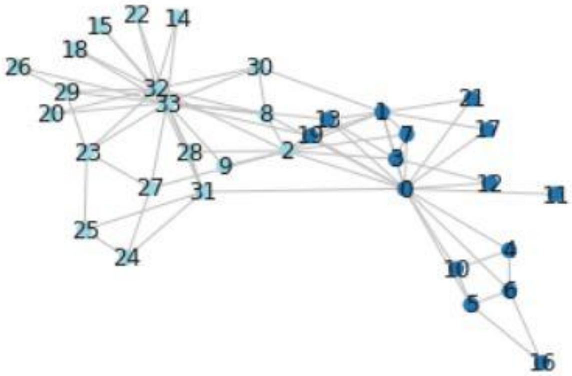


Fig: Result of Markov Clustering Algorithm

Spectral Clustering

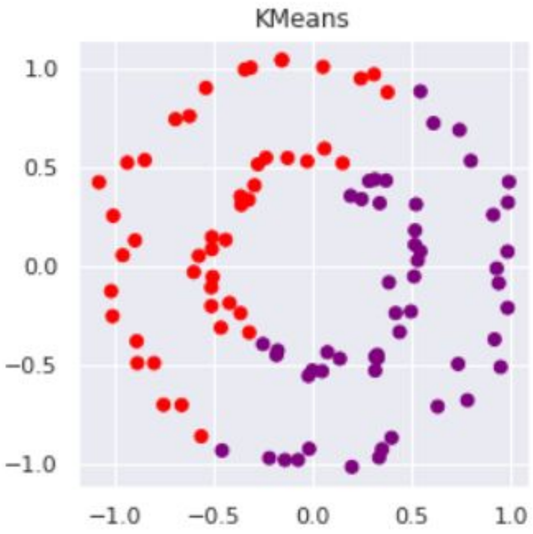
- Spectral Clustering method outperforms the k-means clustering algorithm by a wide margin

- The three primary steps in spectral clustering are:
  - Constructing a similarity network
  - Projecting data into a lower-dimensional space
  - Grouping the data

Spectral Clustering

- Form a distance matrix and then transform it into affinity matrix A.
- Compute the Laplacian matrix which is L=D-A where D is degree matrix. For this laplacian matrix compute eigenvalue and corresponding eigenvectors.
- A matrix is formed by the eigenvectors of the k greatest eigenvalues obtained in the preceding phase.
- Normalization
- In k-dimensional space, group the data points

Spectral Clustering



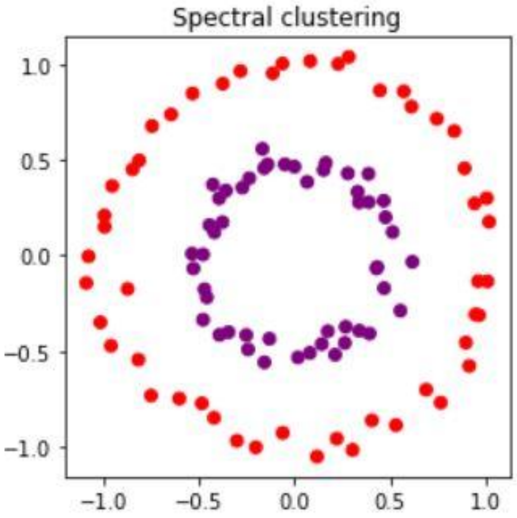


Fig: Results of K-means and Spectral Clustering

Shared Nearest Neighbor Clustering

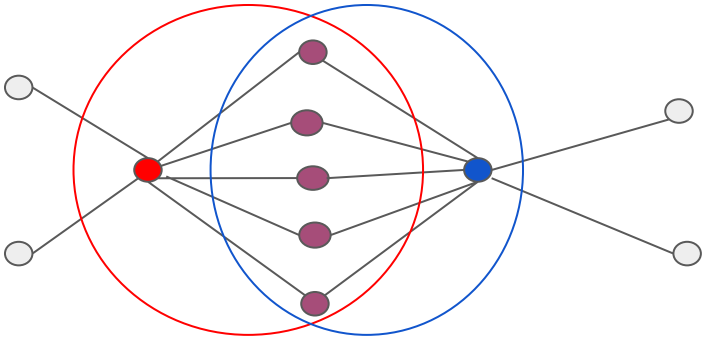
- The concept of similarity based on the sharing of near neighbors.

- If (two nodes are among the k-nearest neighbors of each other):

  similarity is equal to number of shared neighbors.

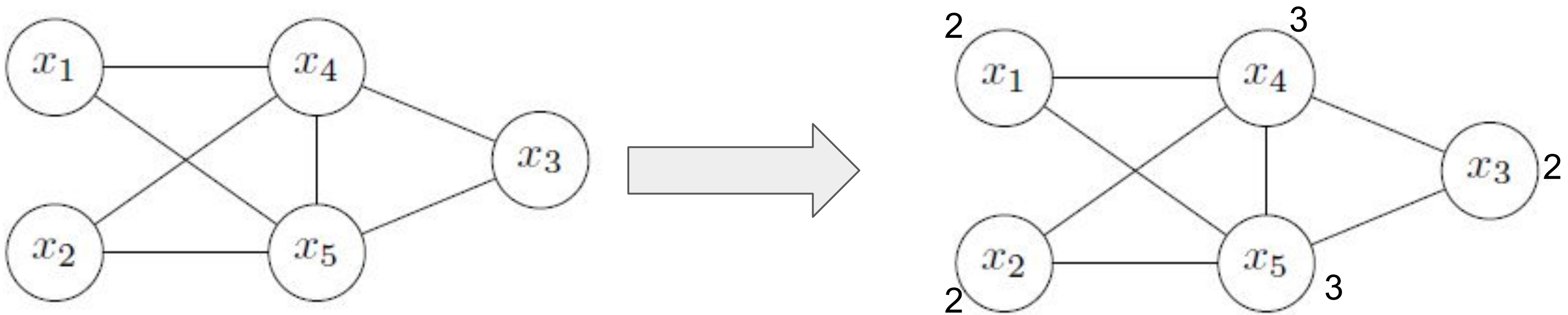
  else :

  similarity is zero.



Shared Nearest Neighbor Clustering

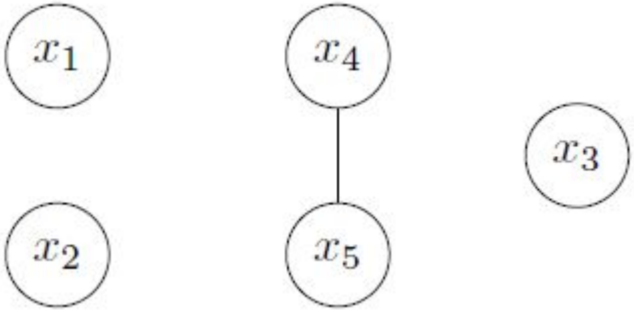
Jarvis Patrick Clustering :



Shared Nearest Neighbor Clustering

Jarvis Patrick Clustering :

If t >3 :



Clustered nodes in a graph.

Betweenness Centrality

- Represents :

  a measure to which a vertex or an edge occurs on the shortest path between all the other possible pairs of nodes in a graph.

- Central nodes in a network.

- Centrality here refers to the numbers or rankings to nodes corresponding to their network position

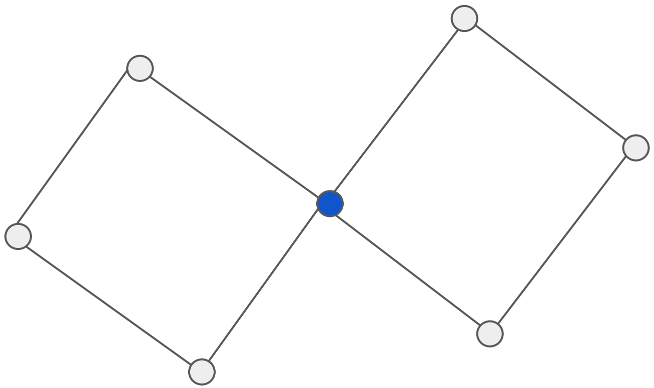
Betweenness Centrality

<u>**Vertex Betweenness Clustering:**</u>

- We calculate the betweenness of all existing vertex in the graph.

- Select the vertex with the highest betweenness.

- Vertex with highest betweenness is removed resulting in a disconnect the graph.

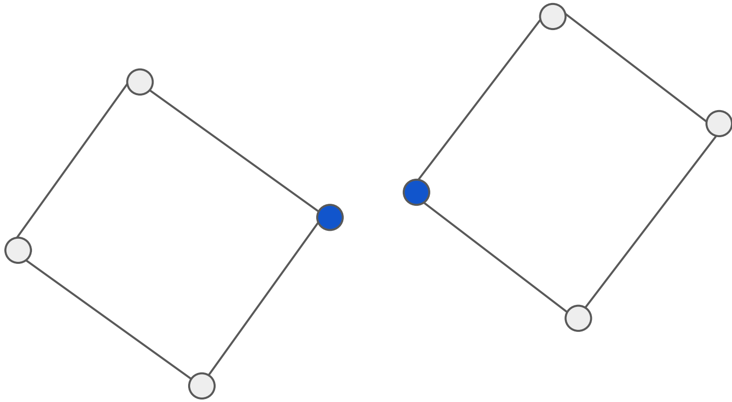


- Repeat until highest vertex betweenness is less than equal to p.

Betweenness Centrality

Vertex Betweenness Clustering:

- We calculate the betweenness of all existing vertex in the graph.
- Select the vertex with the highest betweenness.
- Vertex with highest betweenness is removed resulting in a disconnect the graph.



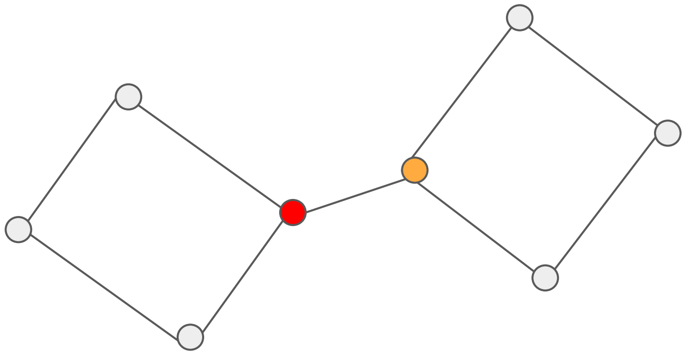
- Repeat until highest vertex betweenness is less than equal to p.

Betweenness Centrality

Edge Betweenness Clustering:

Girvan and Newman Algorithm

- We calculate the edge betweenness for each edge.
- Select the vertex with the highest betweenness.
- Edge with the highest betweenness is removed.
- We repeat it until highest edge betweenness is less than equal to q.

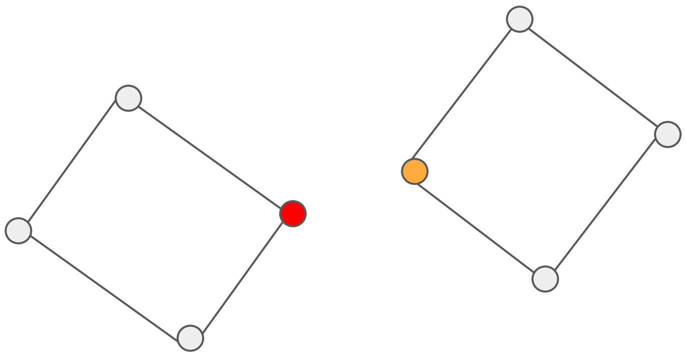


Betweenness Centrality

<u>Edge Betweenness Clustering:</u>

Girvan and Newman Algorithm

- We calculate the edge betweenness for each edge.
- Select the vertex with the highest betweenness.
- Edge with the highest betweenness is removed.
- We repeat it until highest edge betweenness is less than equal to q.



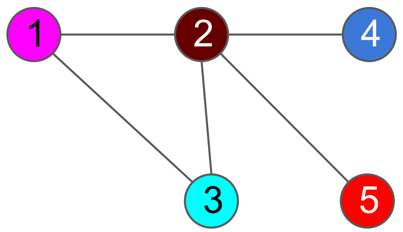
Maximal Clique Enumeration

Bron and Kerbosch Algorithm : takes three parameters, [algo(c,p,n)]

Total vertices in clique(c)

Vertices that can be added(p)

Vertices that cannot be added(n)



algo{Φ,{ 1,2,3,4,5},Φ} → algo{1,{ 2,3},Φ}

algo{{1,2},{ 3},Φ}

algo{{1,3},Φ,2}

algo{{1,2, 3},Φ,Φ}

p and n are empty now, terminate.

Kernel k-means clustering

Vector data points used here.

Select k data points from input as centroids and do the following :

1. Assign other data points to the nearest centroid.
2. Compute centroid for each cluster present.
3. Above two steps are repeated until the centroids don't change.

Within-graph kernel function is used inplace of standard distance used in k means.

Thank you