Spectral Clustering
Spectral 2

Related Work and Background

The following contributions provide tools for efficiently identifying and improving the quality of clusters in complex networks.

Elkan (2003) improved **k-means** efficiency using the triangle inequality.

Newman and Girvan (2004) introduced **modularity (Q)** to measure and optimize cluster quality.

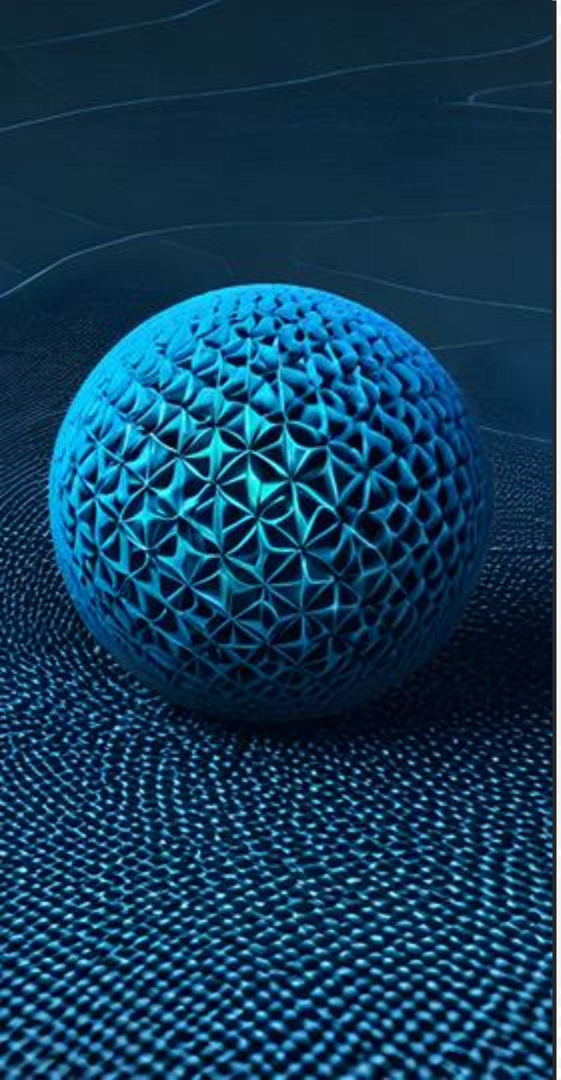
These advancements optimize and improve Spectral clustering in networks and we will discuss soon in each one

Introduction

Spectral clustering is a method that helps to group data into categories that are similar.

It does this by looking at the data in a way that finds out which pieces of data are close or similar in characteristics.

Just as you might group together friends who share similar interests



Outline of the Presentation

01. Key Concepts to Understand

02. Explanation of Algorithm Spectral 2

03. Three Examples of Spectral Clustering

04. Timing Experiments

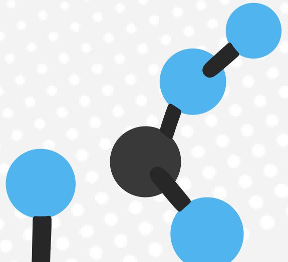
05. Discussion

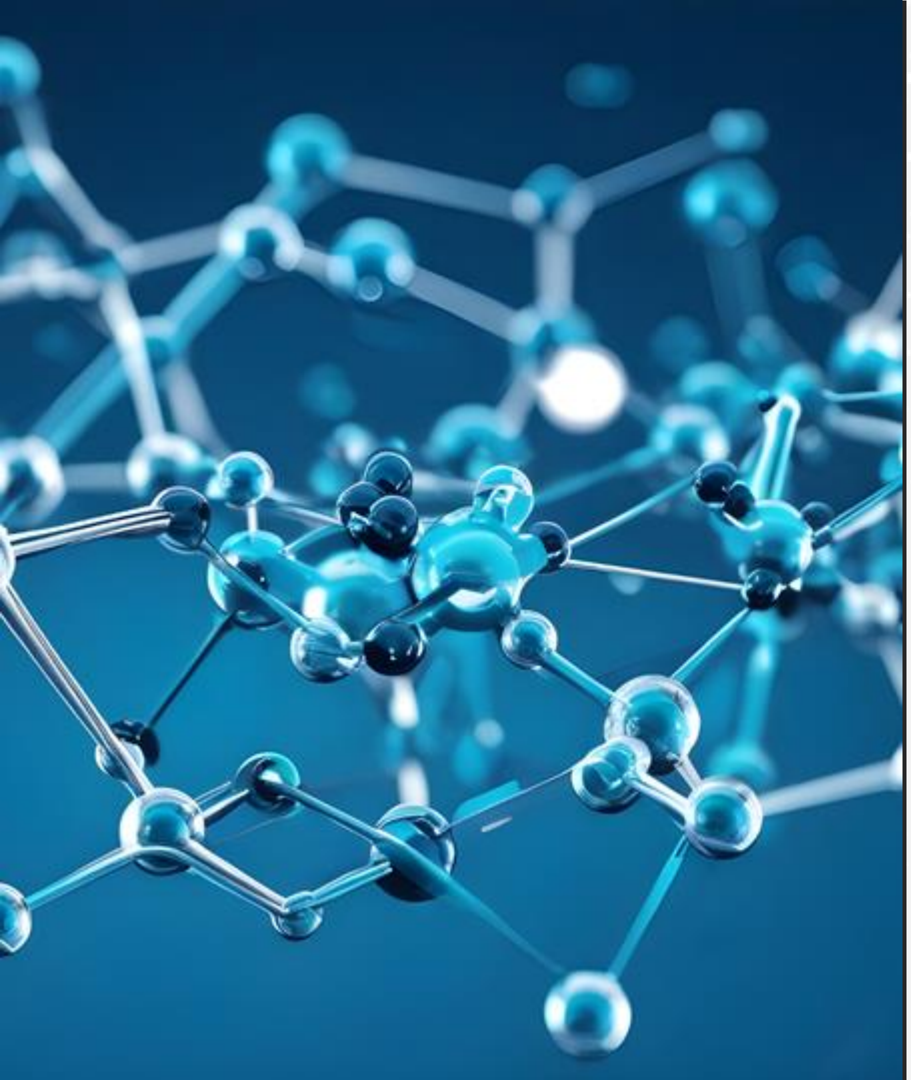
06. Conclusion

Why Use Spectral Clustering?

The goal of spectral clustering is to split data into groups where items in the same group are more alike and items in different groups are less alike.

It helps us understand the data better by organizing it into these groups for different types of usage.





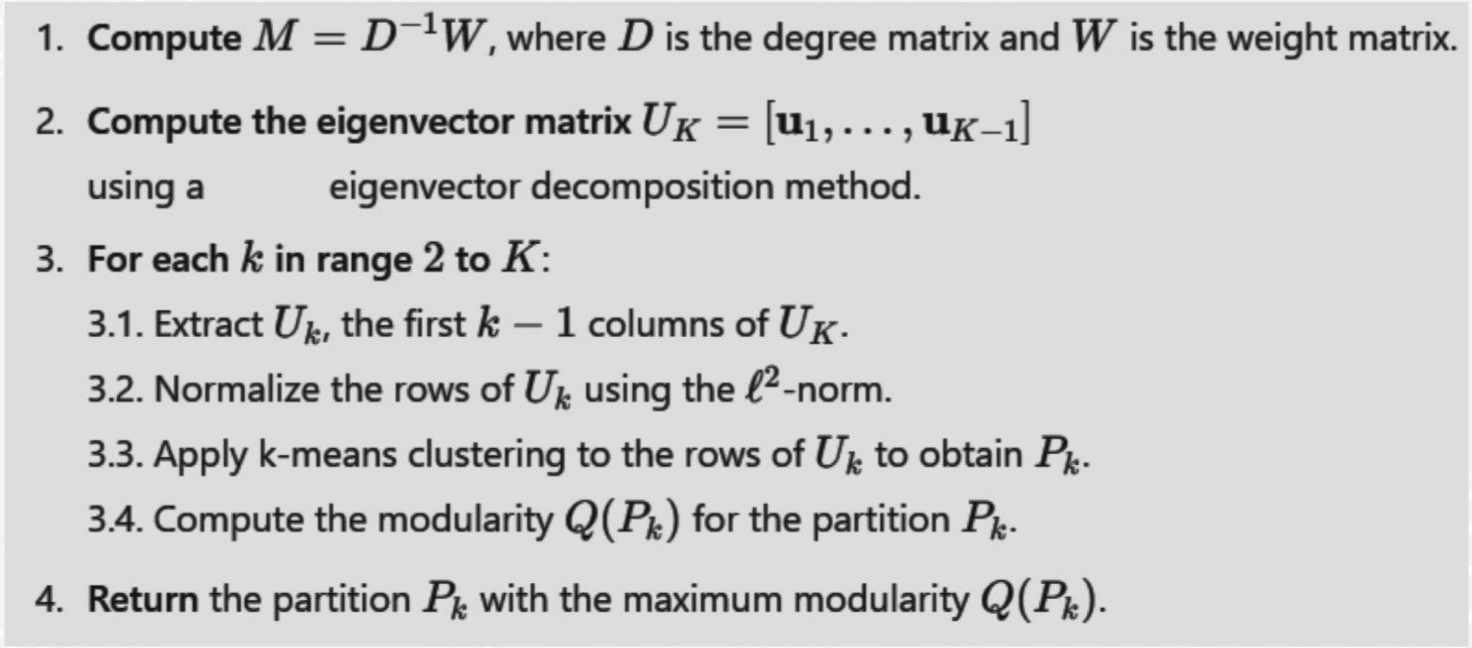
01.

Key Concepts to Understand

Let’s go back to the algorithm we talked about last week.

During this explanation, we will take the time to explore and understand some key concepts step by step.

Algorithm Spectral - 1

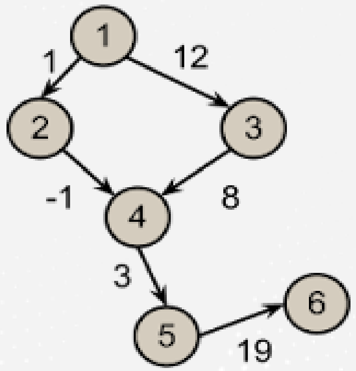


1. **Compute** $M = D^{-1}W$, where D is the degree matrix and W is the weight matrix.
2. **Compute the eigenvector matrix** $U_K = [\mathbf{u}_1, \ldots, \mathbf{u}_{K-1}]$ using a eigenvector decomposition method.
3. **For each** k **in range** 2 **to** K:
   - 3.1. Extract U_k, the first $k - 1$ columns of U_K.
   - 3.2. Normalize the rows of U_k using the ℓ^2-norm.
   - 3.3. Apply k-means clustering to the rows of U_k to obtain P_k.
   - 3.4. Compute the modularity $Q(P_k)$ for the partition P_k.
4. **Return** the partition P_k with the maximum modularity $Q(P_k)$.

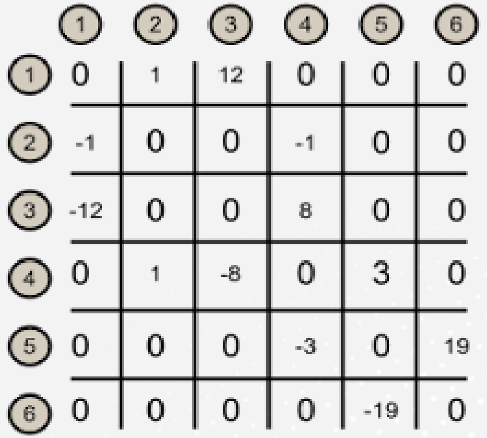
Eigenvector Matrix

W =

Weighted Directed Graph & Adjacency Matrix



Weighted Directed Graph

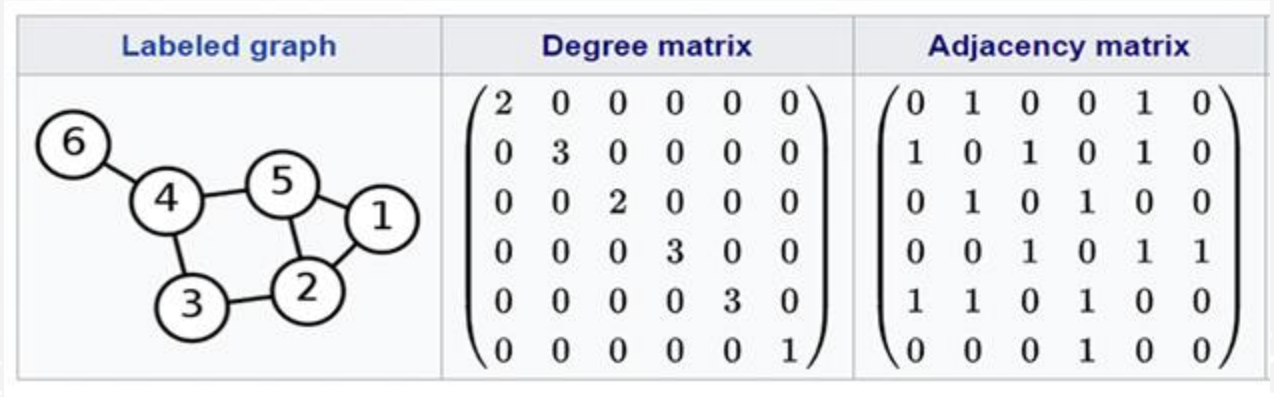


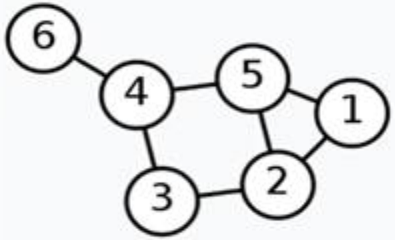
	1	2	3	4	5	6
1	0	1	12	0	0	0
2	-1	0	0	-1	0	0
3	-12	0	0	8	0	0
4	0	1	-8	0	3	0
5	0	0	0	-3	0	19
6	0	0	0	0	-19	0

Adjacency Matrix

Eigenvector Matrix

D =



Labeled graph	Degree matrix	Adjacency matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

Eigenvector Matrix

The eigenvector matrix (UK) consists of the top K eigenvectors of a given matrix.
Each column represents an eigenvector, and these are used to project data into a lower-dimensional space while preserving key structural information, commonly in tasks like spectral clustering.

$$U_K = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,K} \\ u_{2,1} & u_{2,2} & u_{2,3} & \cdots & u_{2,K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{N,1} & u_{N,2} & u_{N,3} & \cdots & u_{N,K} \end{bmatrix}$$

k-means

1. Initialization:

Choose K : Decide the number of clusters

Initialize cluster centroids: Select k random points from the dataset.

2. Assign Points to Clusters:

For each data point:

- Calculate the distance between the point and each cluster centroid
- Assign the point to the cluster with the nearest centroid.

3. Update Cluster Centroids:

Recalculate the position of each cluster's center by computing the average of the coordinates of all the points assigned to that cluster.

4. Repeat Steps 2&3:

- Reassign points to clusters based on the updated centroids.
- Recalculate the centroids based on the new assignments.

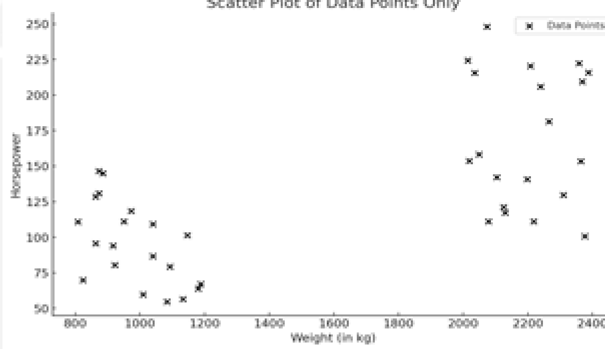
5. Stopping Criterion:

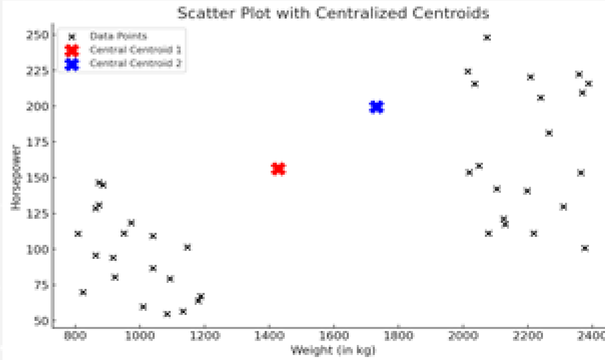
The algorithm stops when one of the following conditions is met:

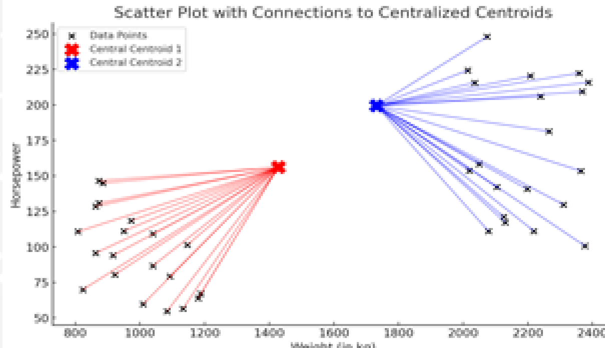
- The centroids no longer change (or change very little).
- Points remain in the same clusters between iterations.
- A maximum number of iterations is reached.

6. Final Output:

The dataset is divided into k clusters, each represented by its centroid.







k-means

algorithm visualization website:

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/?utm_source=chatgpt.com

The modularity function Q

$$Q(\mathcal{P}_k)$$

Modularity is a measure used to evaluate the quality of dividing a graph’s nodes into communities or groups.

Q modularity connects to spectral clustering because it can be improved using eigenvector calculations, which are the core of spectral methods for dividing graphs.

The modularity function Q

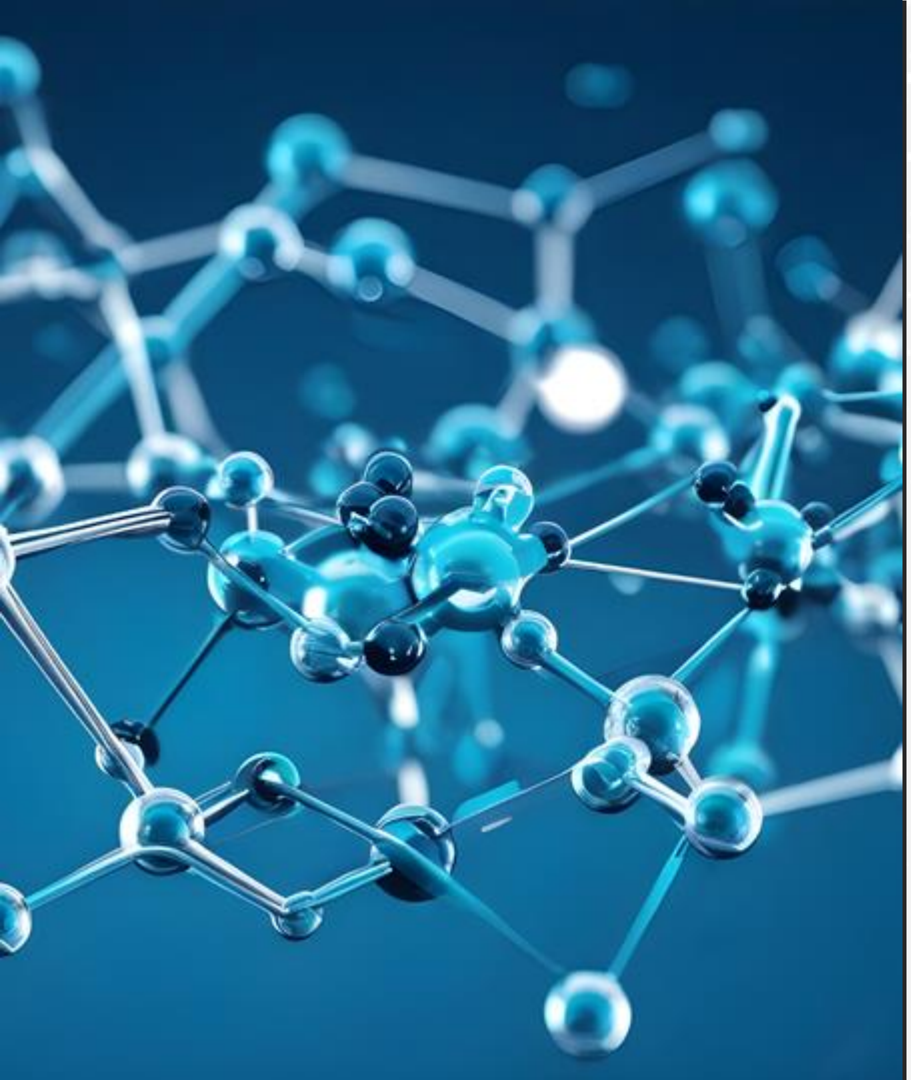
<u>It is based on two principles:</u>

1. Communities should have a high number of intra-community edges (connections between nodes within the same community).

2. They should have a low number of inter-community edges (connections between different communities).

$$-1 \leq Q \leq 1$$

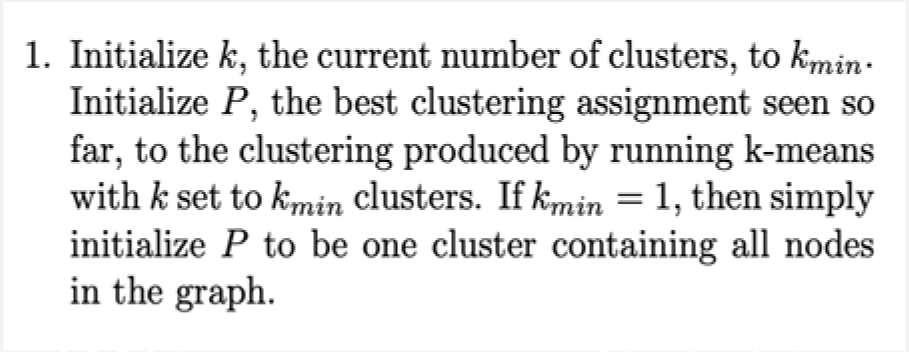
$$Q \in [0.3, 0.7]$$



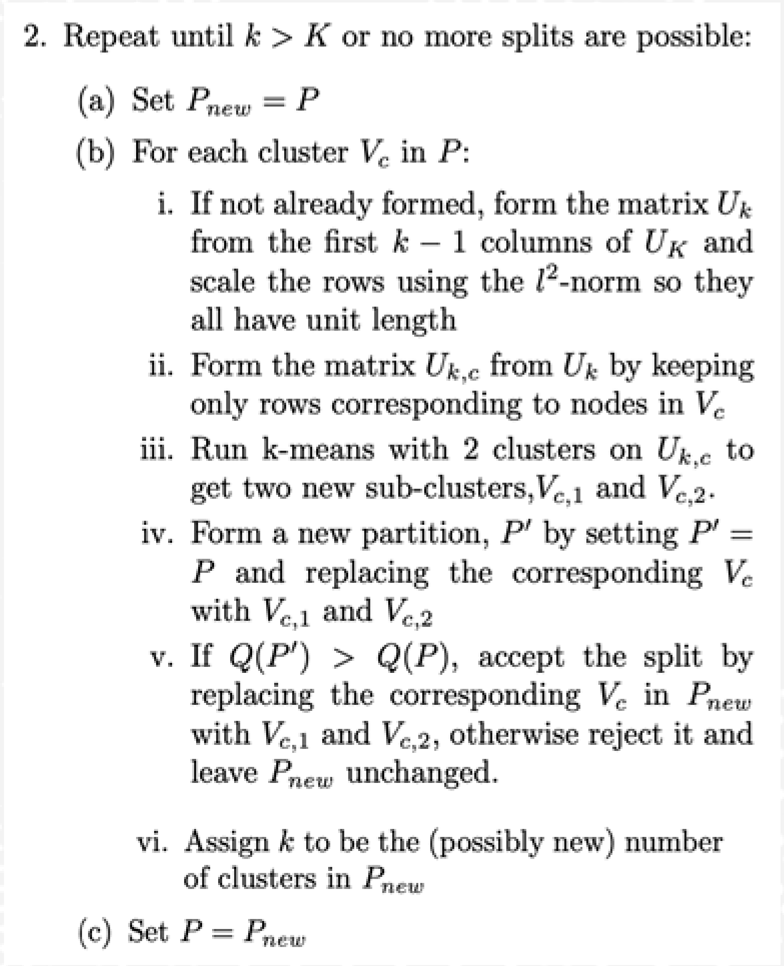
02.

Explanation of Algorithm Spectral 2

How Algorithm Spectral 2 work’s:



1. Initialize k, the current number of clusters, to k_{min}. Initialize P, the best clustering assignment seen so far, to the clustering produced by running k-means with k set to k_{min} clusters. If $k_{min} = 1$, then simply initialize P to be one cluster containing all nodes in the graph.



2. Repeat until $k > K$ or no more splits are possible:
   (a) Set $P_{new} = P$
   (b) For each cluster V_c in P:
      i. If not already formed, form the matrix U_k from the first $k-1$ columns of U_K and scale the rows using the l^2-norm so they all have unit length
      ii. Form the matrix $U_{k,c}$ from U_k by keeping only rows corresponding to nodes in V_c
      iii. Run k-means with 2 clusters on $U_{k,c}$ to get two new sub-clusters,$V_{c,1}$ and $V_{c,2}$.
      iv. Form a new partition, P' by setting $P' = P$ and replacing the corresponding V_c with $V_{c,1}$ and $V_{c,2}$
      v. If $Q(P') > Q(P)$, accept the split by replacing the corresponding V_c in P_{new} with $V_{c,1}$ and $V_{c,2}$, otherwise reject it and leave P_{new} unchanged.
      vi. Assign k to be the (possibly new) number of clusters in P_{new}
   (c) Set $P = P_{new}$

Improving Spectral-2 Clustering with k-means Optimization

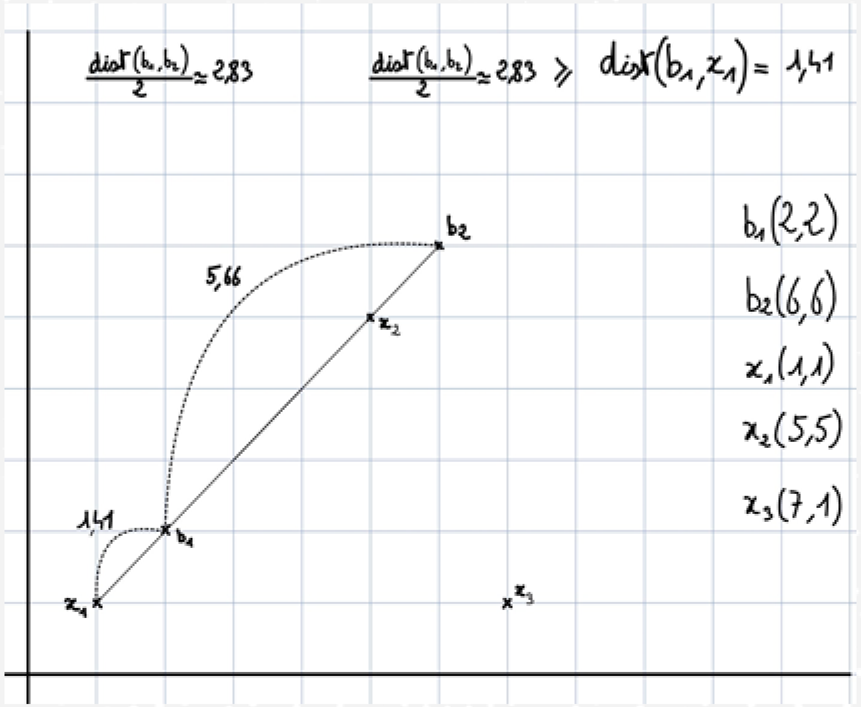
Standard k -means has a complexity of O(ndke)
n is the number of data points
d is the dimensionality of each point
K is the number of clustering
e is the number of iterations required for k-means to converge

Charles Elkan algorithm improves this complexity to O(nke) by reducing unnecessary distance computations through the use of intelligent bounds.

Using the Lemmas in the following slides

Lemma 1:

The first Lemma states that if the distance between a point *x* and a centroid *b* is shorter or equal than *half* the distance between *b* and the closest centroid *c* to *b*, then *b* is the closest centroid to *x*.



$\frac{dist(b_1,b_2)}{2} \approx 2{,}83$
$\frac{dist(b_1,b_2)}{2} \approx 2{,}83 \geqslant dist(b_1,x_1) = 1{,}41$
$b_1(2,2)$
$b_2(6,6)$
$x_1(1,1)$
$x_2(5,5)$
$x_3(7,1)$
5,66
1,41
b_1
b_2
x_1
x_2
x_3

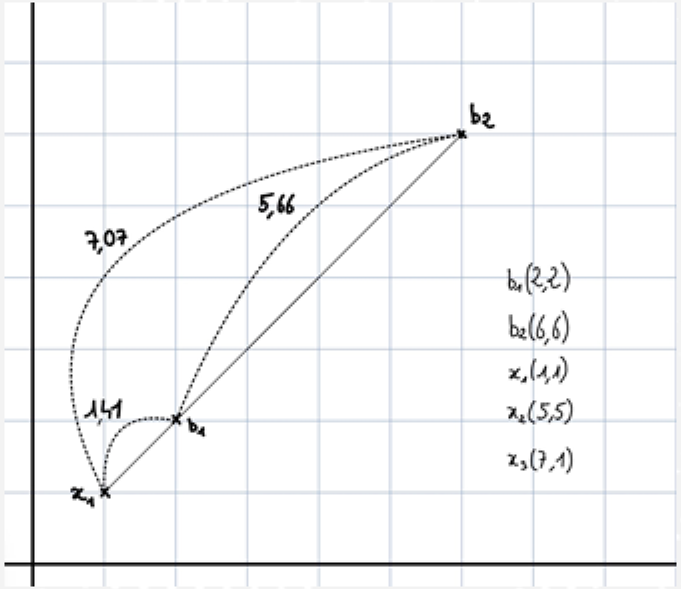
$$\text{distance}(A,B) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

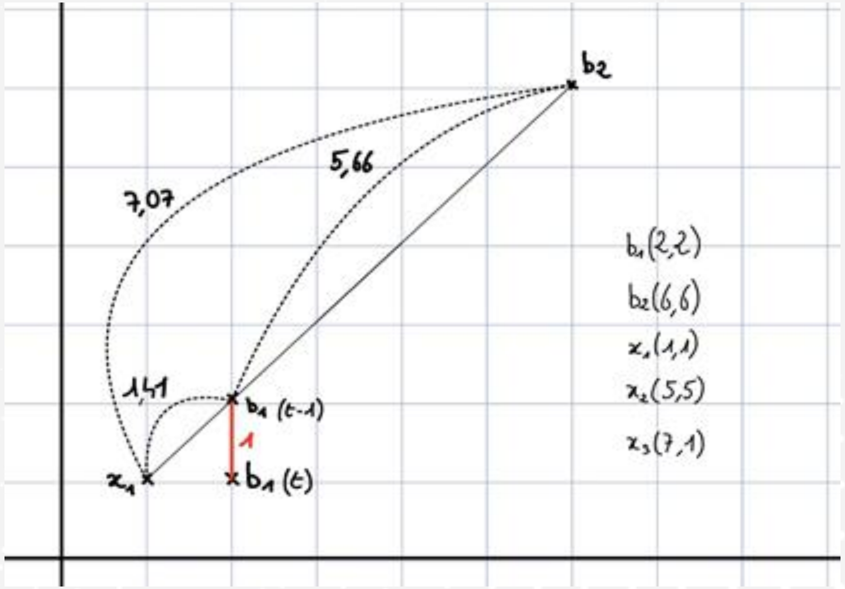
$$\text{distance}(A,B) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

Lemma 2:

In plain words, the second lemma states that the distance between a point *x* and a centroid *b* at time *t* will always be larger or equal than the distance between point *x* and centroid *b* at time *t-1* minus the change in centroid *b*'s position between time *t-1* and time *t*.

$$|\text{distance}(A,B) - \text{distance}(B,C)| \leq \text{distance}(A,C) \leq \text{distance}(A,B) + \text{distance}(B,C).$$





$$0.41 = \text{dist}(x_1, b_1(t-1)) - \text{dist}(b_1(t-1), b_1(t)) \leq \text{dist}(x_1, b_1(t)) \leq \text{dist}(x_1, b_1(t-1)) + \text{dist}(b_1(t-1), b_1(t)) = 2.41$$

Computational Complexity Summary of Spectral Clustering

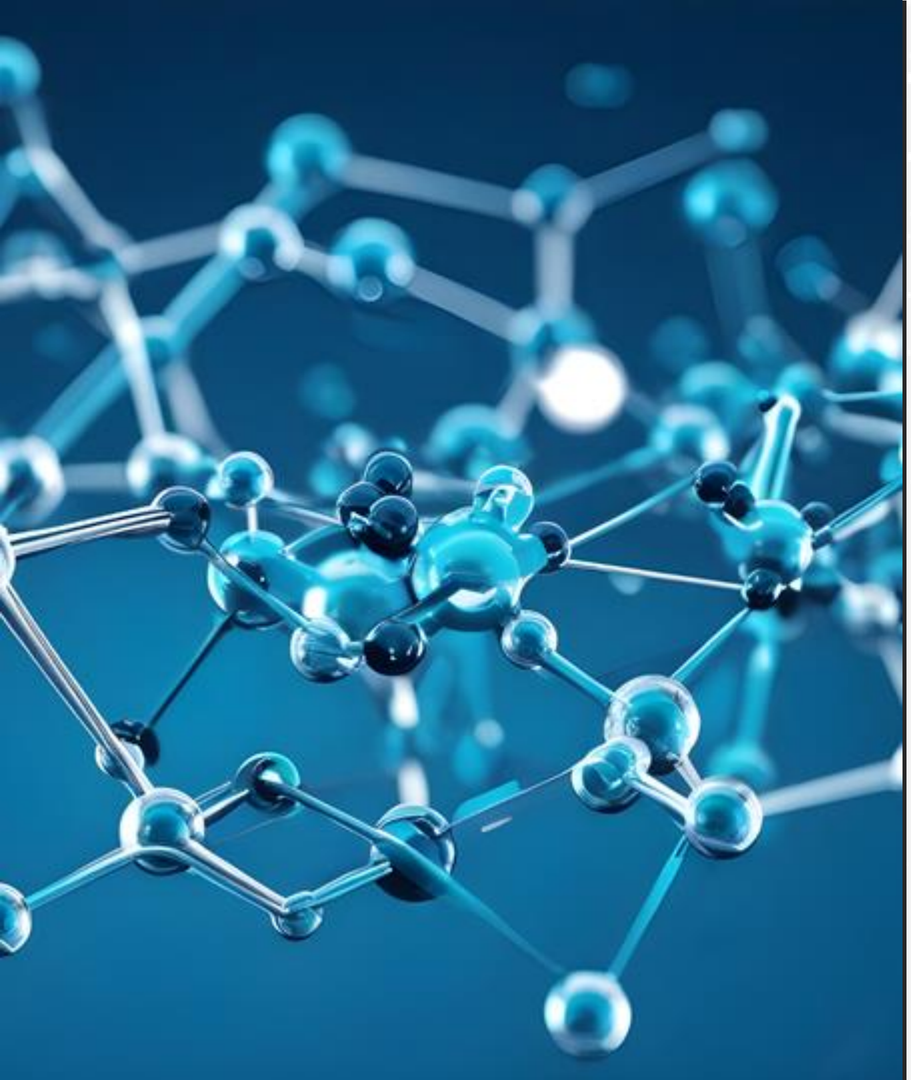
Complexity k-means with Elkan’s Optimization $\longrightarrow O(n \cdot k \cdot e)$

Complexity k-means with Elkan’s Optimization within Spectral Clustering $\longrightarrow O(nK^2e)$

Complexity of Computing Eigenvectors via IRLM $\longrightarrow O(mKh + nK^2h + K^3h)$

Worst case Time Complexity of Spectral Clustering $\longrightarrow O(mKh + nK^2h + K^3h + nK^2e)$

Spectral 1 & Spectral-2 sharing a worst-case time complexity,Comparing to Spectral-1, Spectral-2 is generally faster because it uses a greedy splitting strategy rather than recomputing clusters for all nodes at every iteration.This makes Spectral-2 a more efficient algorithm for large-scale problems



03.

Three Examples of Spectral Clustering

Examples

- Clustering words from WordNet -
- Clustering American college football teams -
- Clustering authors publishing in NIPS -

First Example

Clustering words from WordNet

What is WordNet?

A dataset of 200,000 English words with semantic relationships.

Synonymy: *big* → *large*

Antonymy: *hot* → *cold*

Hypernymy: *color* → *red*

Hyponymy: *rose* → *flower*

Meronymy: *wheel* → *car*

Troponymy: *to stroll* → *to walk*

Objective

Use a subset of WordNet to build a graph and cluster words based on their relationships.

Clustering words from WordNet

In this slide
Words are represented as nodes in the graph.
Edges are added between nodes based on semantic characteristics, such as synonymy, antonymy, or hypernymy.

The graph is converted to vectors by computing the eigenvectors of the Laplacian matrix derived from the graph.

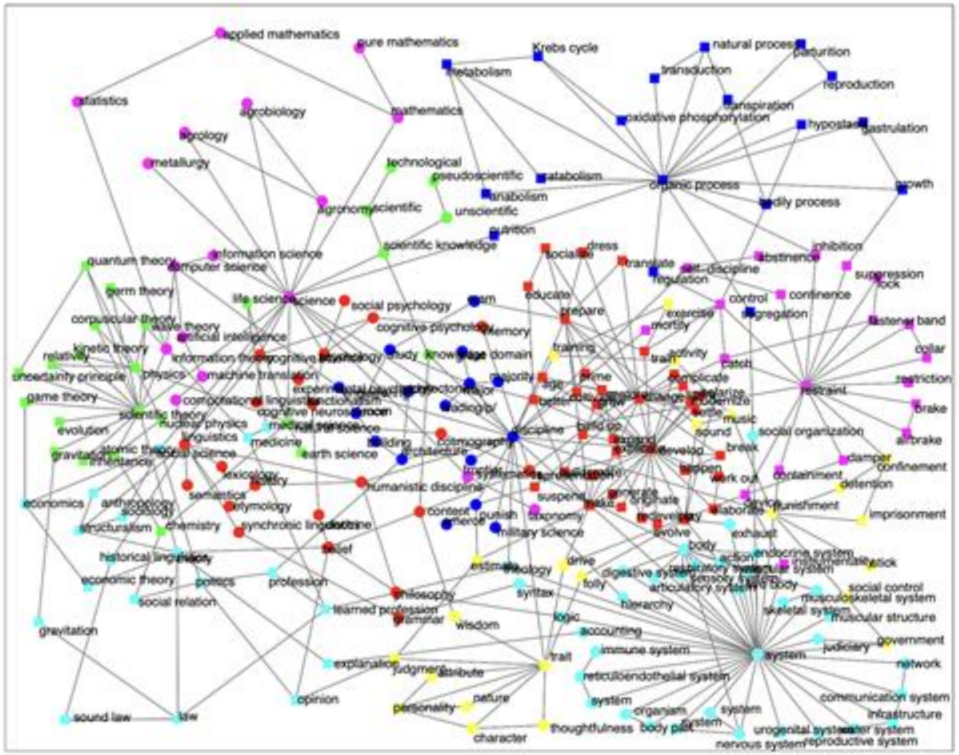


Figure 2: Clusters for WordNet data, $k = 12$ (best viewed in color).

Table 1: Sample clusters found for the WordNet data.

Hard Science	Qualities	Metabolism	Soft Science	Systems
taxonomy	attribute	regulation	social relation	organism
science	drive	reproduction	profession	body
mathematics	folly	Krebs cycle	social science	hierarchy
pure mathematics	judgment	hypostasis	law	digestive system
applied mathematics	estimate	nutrition	politics	infrastructure
statistics	trait	growth	medicine	network
information theory	personality	anabolism	theology	system
computer science	character	bodily process	opinion	water system
information science	nature	catabolism	explanation	live body
information theory	thoughtfulness	gastrulation	anthropology	sensory system

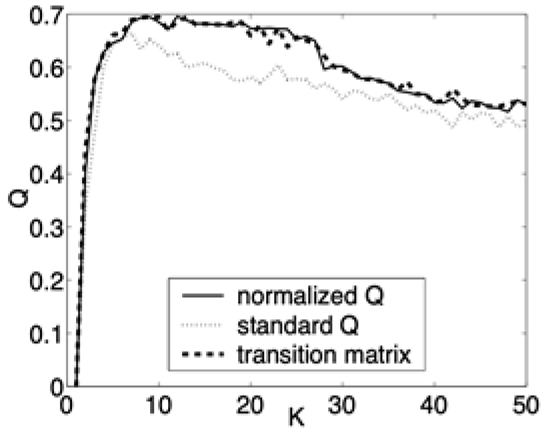


Figure 3: Q versus k for the WordNet data.

$$L_Q = D - A$$

$$L_Q^{\text{norm}} = I - D^{-1/2} A D^{-1/2}.$$

$$T = D^{-1} A.$$

In a spectral algorithm, for extracting eigenvectors is key to analyzing and partitioning a graph.
You can use one of three matrices (Standard Laplacian, Normalized Laplacian, or Transition Matrix) for extracting eigenvectors and is a key to analyzing and partitioning a graph.
That project the graph's data into a lower-dimensional space, where clustering becomes easier.

The matrix choice impacts:
1. **Cluster quality**: It determines how well-connected nodes are grouped together.
2. **Behavior of Q vs k** : Different matrices lead to different modularity Q values as the number of clusters k increases, influencing the clustering's effectiveness and the optimal k .

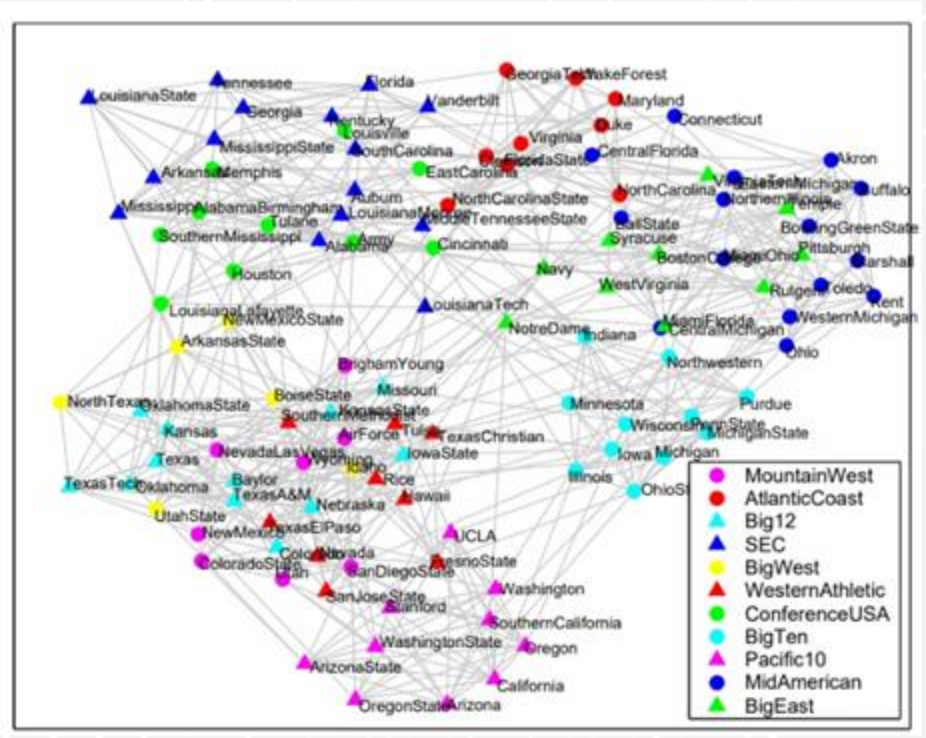
Second Example

Clustering American College Football Teams

Clustering American college football teams

The following experiment explores the effectiveness of graph clustering algorithms on a real-world dataset: American college football team scheduling that occur in the year 2000.

Clustering American college football teams



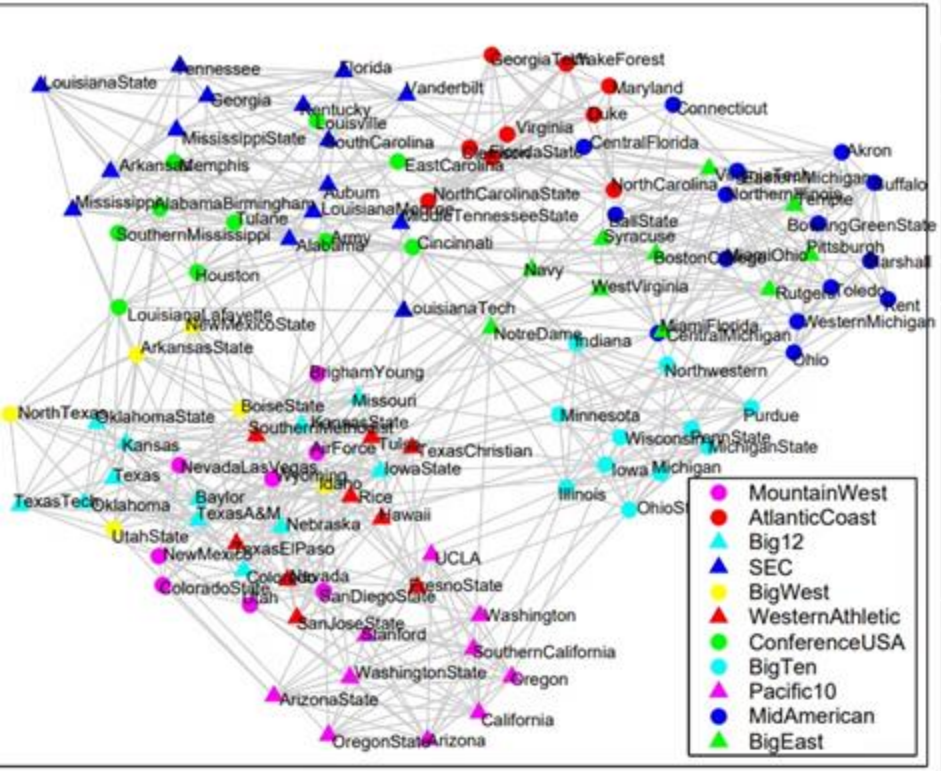
Objective:

The experiment aimed to assess the effectiveness of our clustering algorithms (Spectral-1 and Spectral-2) in grouping NCAA football teams based on their actual conference affiliations using game schedules from the year 2000.

Graph Details:

Vertices: 115 NCAA football teams.

Edges: Games played between teams.



Network Representation

Clustering Target

Since each team's true conference is known beforehand, conferences naturally form identifiable clusters in the graph.

Network Structure

Vertices represent college football teams.

Edges signify games played between teams.

From Football Teams to Graph Representation

Nodes and Edges:

- Teams are nodes
- Edges represent games played between teams.

Graph Purpose:

- Captures relationships; stronger connections within conferences.

Clustering Preparation:

- The graph's structure is converted into a set of points in space, where each team is a point, making it easier to group teams based on the games they played against each other.

Newman algorithm

In few words Newman's algorithm is used for detecting communities in networks by iteratively merging clusters to maximize modularity.
It works efficiently with a time complexity of O(n^2),
making it suitable for moderately sized networks but less for large graphs.

Approach:
- Starts with n clusters (one for each node).
- Gradually merges clusters, one at a time, until max modularity (Q) reached and no there is no improvement in Q value in next step of merging.

Limitation:
- Mistakes made early in the merging process can lead to suboptimal clustering.

Results:

1

Spectral-1:
Found 11 clusters with a modularity score of Q=0.602.

2

Spectral-2:
Found 10 clusters with a modularity score of Q=0.553

3

Newman:
Found 6 clusters with a modularity score of Q=0.556

Difference: **Spectral-1** achieved the highest modularity and most accurate clustering, while **Spectral-2** was faster but slightly less accurate, and **Newman** underperformed due to its local optimization approach.

Third Example

Clustering authors publishing in NIPS

NEURAL INFORMATION PROCESSING SYSTEMS

Objective:

The experiment analyzed the collaboration network of authors who participated in the *NIPS conference. The goal was to apply **Spectral-1** and **Spectral-2** clustering algorithms to identify meaningful clusters of authors based on their co-authorship connections in the network.

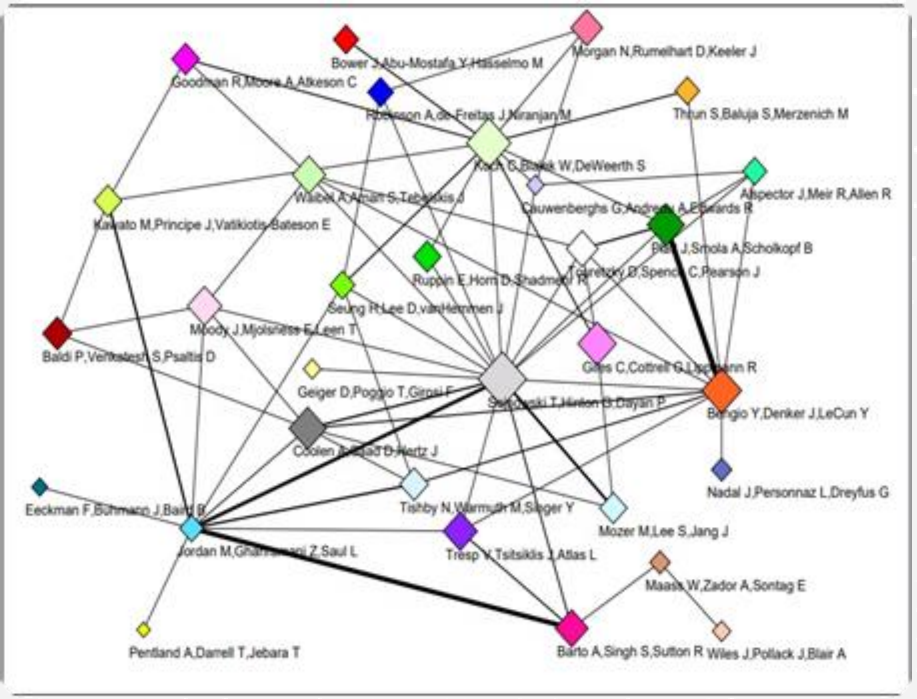
Key Results:

Input:

- A network of authors from the NIPS conference.
- K=100 (maximum possible clusters).

* NIPS is an annual event in artificial intelligence and machine learning that showcases new research and attracts scholars from around the world.

Results of Clustering NIPS Authors


Bower J,Abu-Mostafa Y,Hasselmo M
Morgan N,Rumelhart D,Keeler J
Goodman R,Moore A,Atkeson C
Robinson A,de-Freitas J,Niranjan M
Thrun S,Baluja S,Merzenich M
Koch C,Bialek W,DeWeerth S
Waibel A,Amari S,Tebelskis J
Andreou A,Edwards R
Kawato M,Principe J,Vatikiotis-Bateson E
Cauwenberghs G,Andreou A,Edwards R
Alspector J,Meir R,Allen R
Platt J,Smola A,Scholkopf B
Ruppin E,Horn D,Shadmehr R
Touretzky D,Spence C,Pearson J
Seung R,Lee D,vanHemmen J
Moody J,Mjolsness E,Leen T
Baldi P,Venkatesh S,Psaltis D
Giles C,Cottrell G,Lippmann R
Geiger D,Poggio T,Girosi F
Sejnowski T,Hinton G,Dayan P
Bengio Y,Denker J,LeCun Y
Coolen A,Saad D,Hertz J
Nadal J,Personnaz L,Dreyfus G
Eeckman F,Buhmann J,Baird B
Tishby N,Warmuth M,Singer Y
Mozer M,Lee S,Jang J
Jordan M,Ghahramani Z,Saul L
Tresp V,Tsitsiklis J,Atlas L
Maass W,Zador A,Sontag E
Pentland A,Darrell T,Jebara T
Barto A,Singh S,Sutton R
Wiles J,Pollack J,Blair A

Spectral-1 achieved a **modularity** of **0.874** at **k=31.**

Spectral-2 achieved a **modularity** of **0.861** at **k=44**.

These results show that Spectral-1 was accurate.

On the other hand Spectral-2 worked faster,

though he was a bit less accurate.

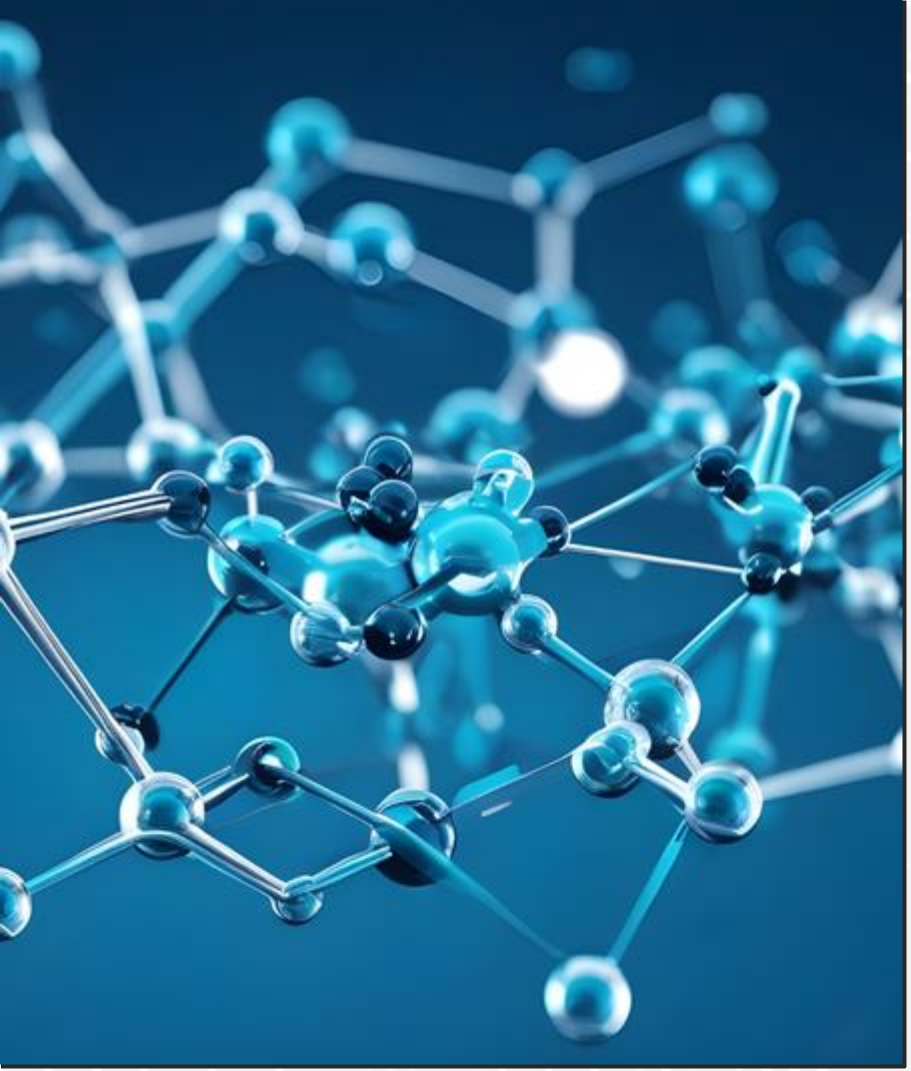
NIPS co-authorship data known in advance k = 31

(Clusters).

Image Explanation:

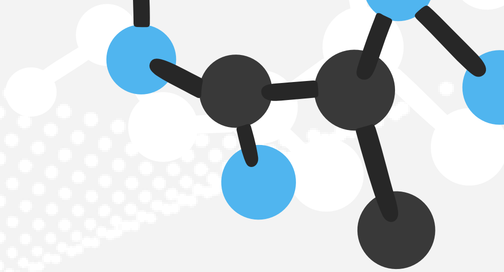
vertices = Author

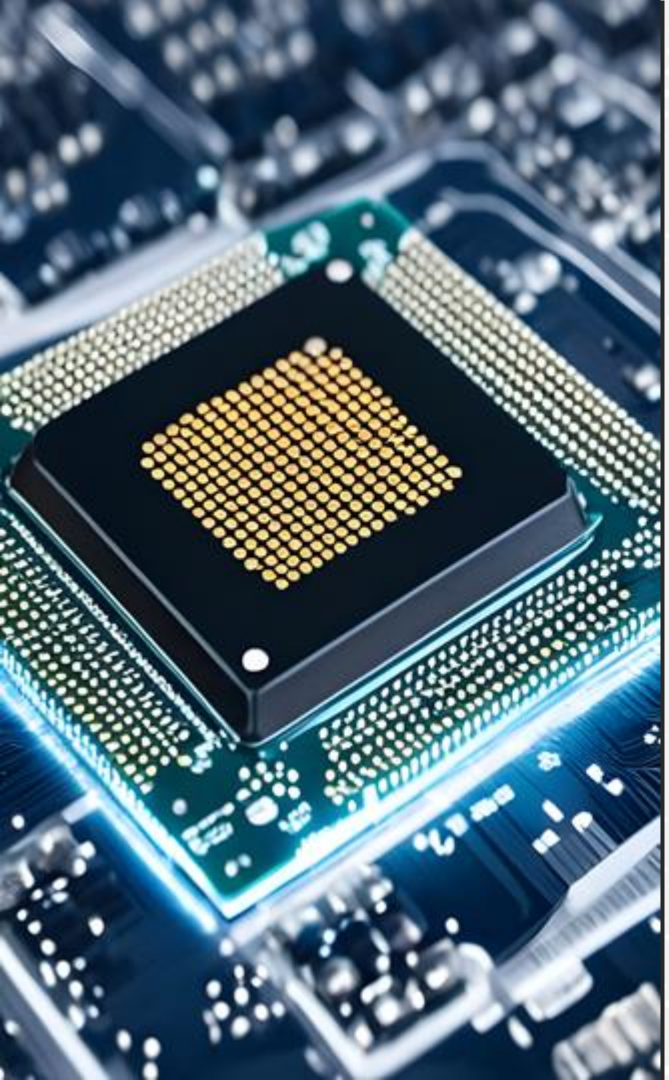
Edge = Collaboration between authors



04.

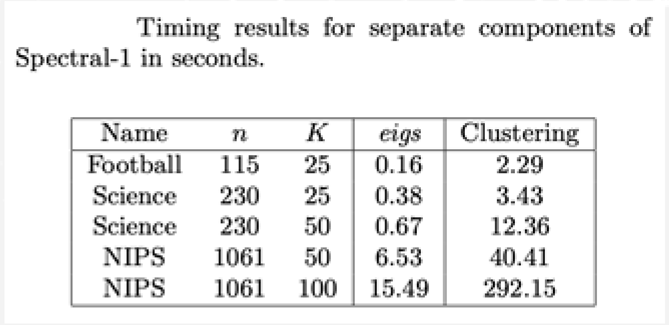
Timing Experiments





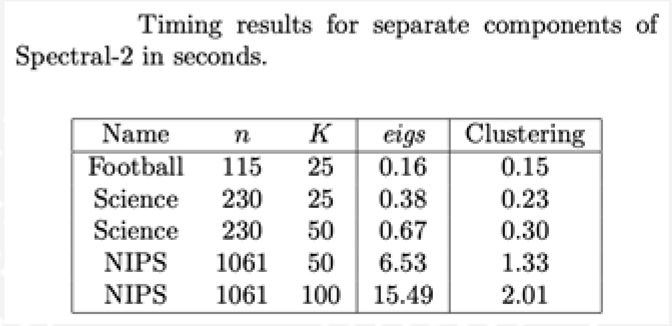
All algorithms were implemented in Matlab and run on a 1.2 GHz Pentium II laptop.

We used the eigen decomposition routine eigs in Matlab to compute the eigenvectors using the IRLM.



Timing results for separate components of Spectral-1 in seconds.

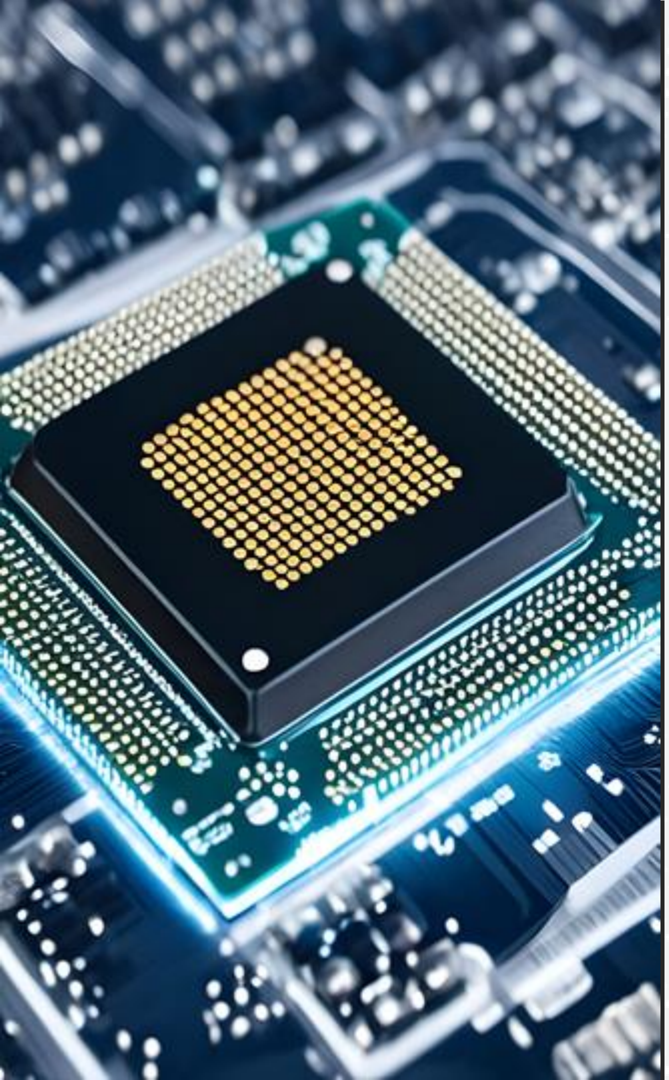
Name	n	K	*eigs*	Clustering
Football	115	25	0.16	2.29
Science	230	25	0.38	3.43
Science	230	50	0.67	12.36
NIPS	1061	50	6.53	40.41
NIPS	1061	100	15.49	292.15



Timing results for separate components of Spectral-2 in seconds.

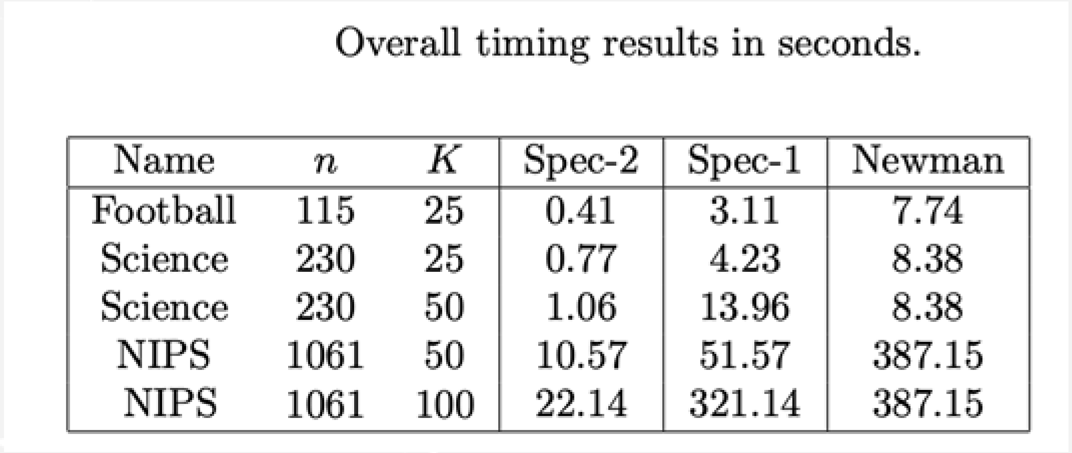
Name	n	K	*eigs*	Clustering
Football	115	25	0.16	0.15
Science	230	25	0.38	0.23
Science	230	50	0.67	0.30
NIPS	1061	50	6.53	1.33
NIPS	1061	100	15.49	2.01

- **Eigs**: Time taken to compute eigenvectors of the graph matrix.
- **Clustering**: Time taken to run k-means after obtaining the eigenvectors
- **n**: number of data nodes
- **K**: number of clusters



The table presents the overall timing results for three algorithms: **Spectral-1**, **Spectral-2**, and **Newman**.
The timing corresponds to the sum of the time required for computing eigenvectors and the time taken to perform k-means clustering for each algorithm.

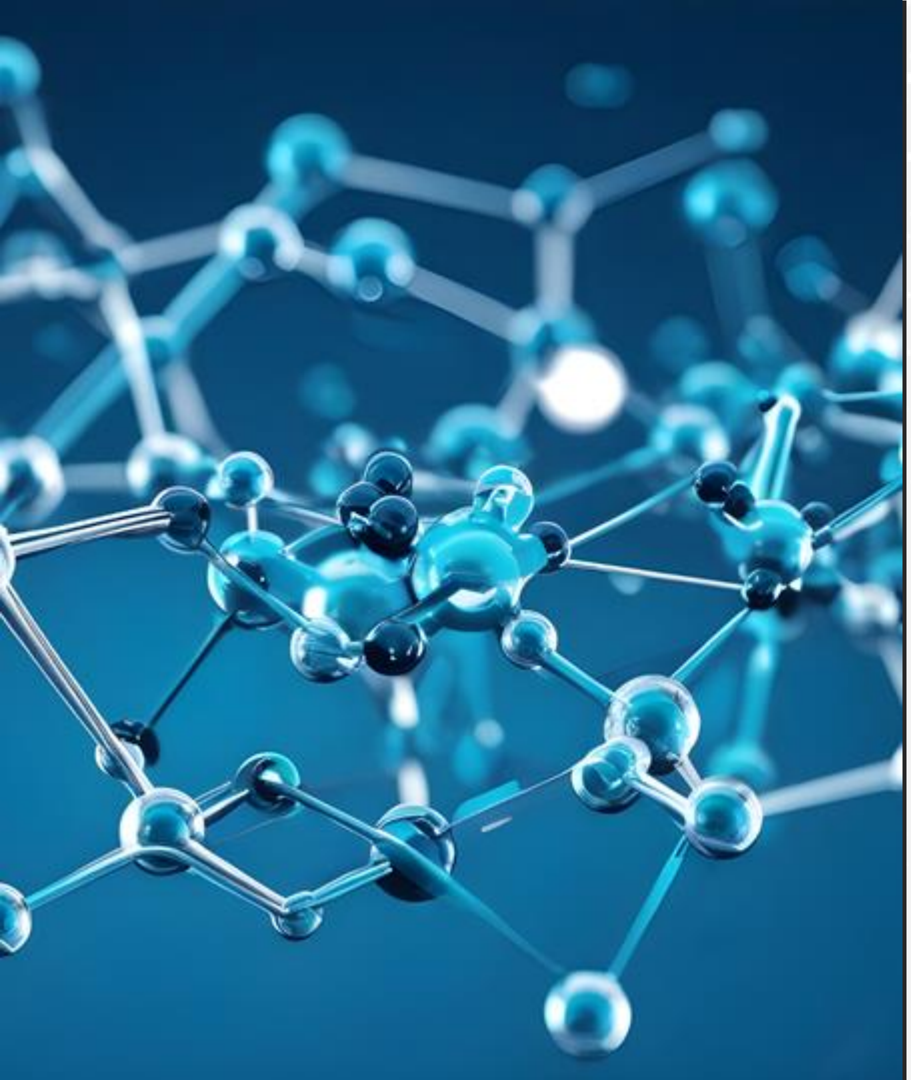
The results are displayed for different datasets, with varying numbers of nodes and clusters. The total time reflects the computational cost for each specific example.



Overall timing results in seconds.

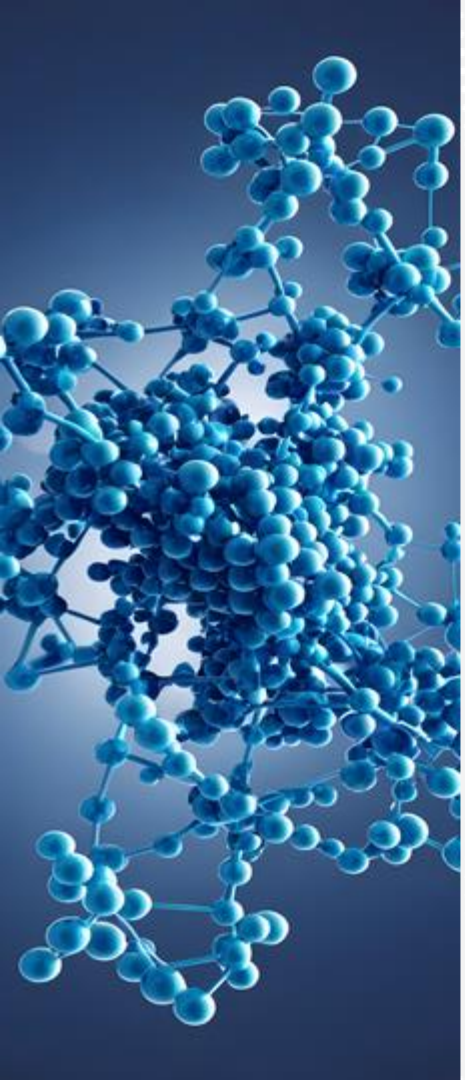
Name	n	K	Spec-2	Spec-1	Newman
Football	115	25	0.41	3.11	7.74
Science	230	25	0.77	4.23	8.38
Science	230	50	1.06	13.96	8.38
NIPS	1061	50	10.57	51.57	387.15
NIPS	1061	100	22.14	321.14	387.15

- **n**: number of data nodes
- **K**: number of clusters



05.

Discussion



DISCUSSION

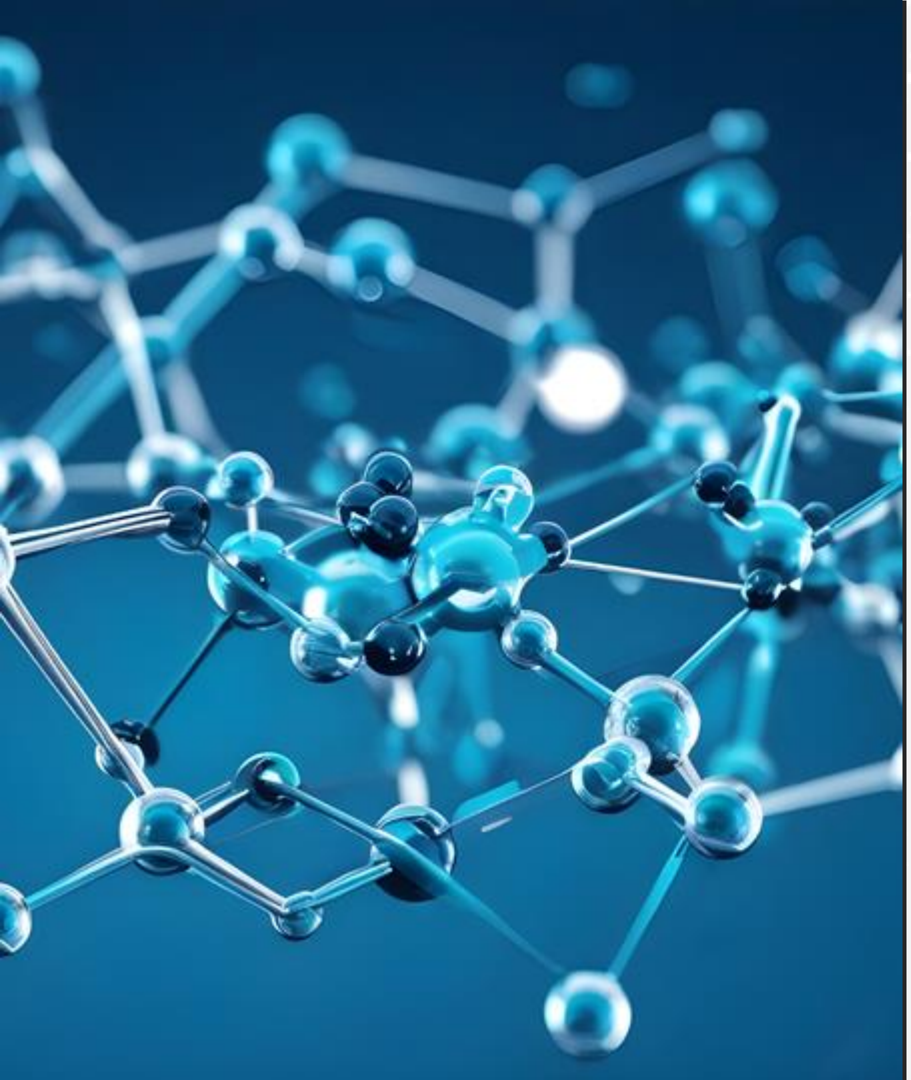
- Spectral clustering simplifies graph partitioning using eigenvector-based geometric clustering in Euclidean space.

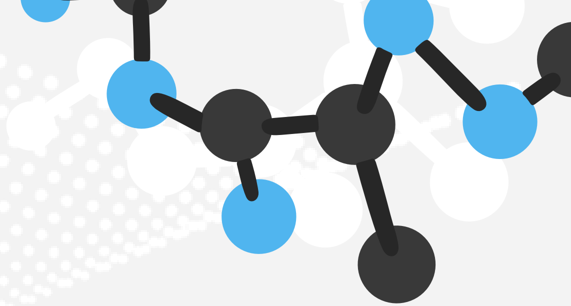
Spectral-1 highly accurate because it looks at the entire dataset to find the best possible clustering. This makes it useful for tasks that need precise grouping, like studying genetic patterns or finding communities in social networks.(Dividing a City into Public Transportation Zones)

Spectral-2 balances speed and quality, making it suitable for large datasets or real-time applications.(Example - Easy app)

Newman's algorithm, while less accurate, is practical for small-scale or exploratory clustering tasks.(Example - class students groups)

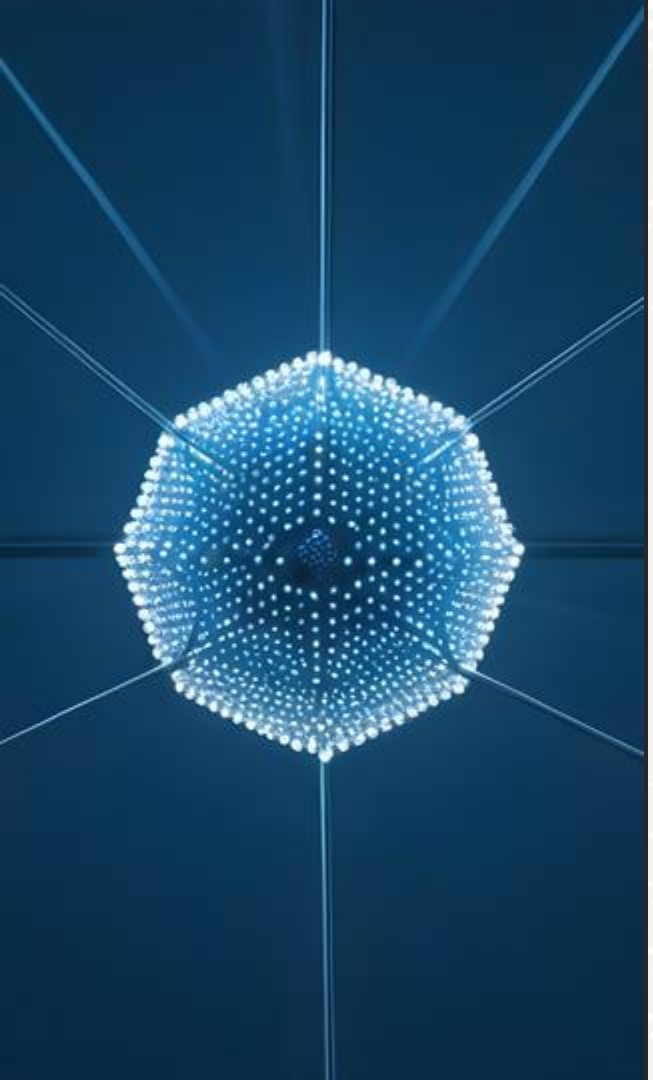
Future exploration can combine these methods to optimize speed and accuracy for diverse applications.





06.

Conclusion



Conclusion

The paper demonstrates how the Q function can identify high-quality graph clusters by approximating its NP-Complete discrete maximization through a continuous optimization using the Q-Laplacian matrix.

Spectral-1:

Highest accuracy for complex tasks needing detailed clustering.

Spectral-2:

Fastest runtime, effective for large-scale and real-time applications.

Spectral clustering offers a versatile, efficient approach for graph-based problems, combining computational efficiency with high clustering quality. To conclude Spectral clustering algorithms offer a promising approach for graph clustering, demonstrating accuracy and computational efficiency.

Thanks !

References

C. Elkan. Using the triangle inequality to accelerate k-Means. In Proceedings of the Twentieth International Conference on Machine Learning, 2003, pp. 147-153.

M. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E, 69, 026113 (2004).

M. Newman. Fast algorithm for detecting community structure in networks. Physical Review E, 69, 066133 (2004).