CSE8803/CX4803 Machine Learning in Computational Biology

Lecture 11: Clustering III

Xiuwei Zhang

School of Computational Science and Engineering

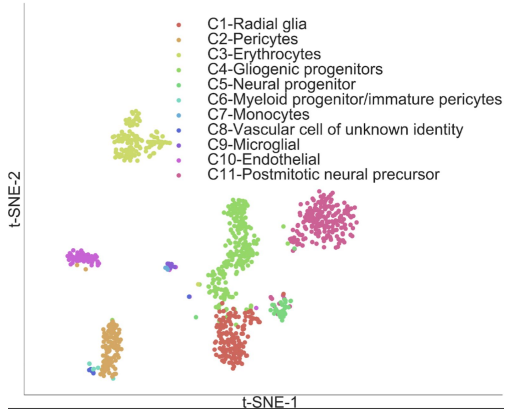
Course logistics

- Final exam: take home and timed
  - 24 hour time window: May 4, 10:00am to May 5, 10:00am
  - 4 hour duration including the time to scan and submit your solutions to Canavs quizzes. You must successfully submit your solutions to Canvas or it is considered as late submission.
- Paper presentation comments
  - Each student in the audience can submit optional comments on paper presentations
  - Some comments will be considered when evaluating the presentation
  - Reasonable and meaningful comments on the presentation are also counted towards your “participation” points. You can also submit insights on the paper (that is not discussed during the presentation) and that can also be considered towards participation points.
  - Comments are due by the end of the next day (eg. deadline for comments on presentations on 2/21 are due on 2/22 11:59pm).

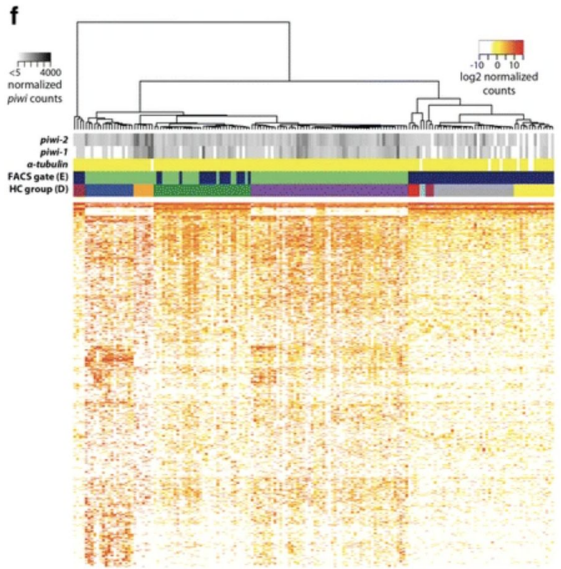
Clustering algorithms

Clustering - unsupervised learning methods

- K-means (and fuzzy k-means)
- Hierarchical clustering
- Spectral clustering
- Louvain, Leiden clustering



https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007040

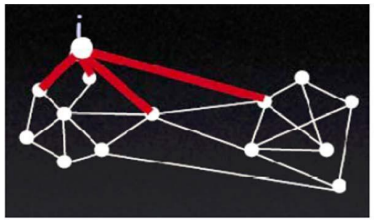


https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0937-9/figures/1

Spectral clustering: notations

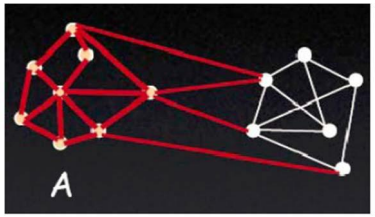
- Degree of nodes

$$d_i = \sum_j w_{i,j}$$



- Volume of a set

$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$



W: adjacency matrix of the graph

D: diagonal matrix where D(i,i)=d_i

L: L=D-W

Spectral clustering: steps

Objective of spectral clustering: minimize normalized cut

$$\text{Ncut}(A, B) = \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

Steps:

1. Calculate D, $D(i,i) = \sum_j W(i,j)$
2. Calculate L, L = D - W, and L', L'= $D^{-1/2}LD^{-1/2}$
3. Calculate eigenvalues and eigenvectors of L'. Sort the eigenvalues from low to high values.
4. Take the corresponding first k eigenvectors
5. Perform k-means on these eigenvectors and obtain clusters

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\text{Ncut}(A, B) = \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

$$= \sum_{i \in A, j \in B} w_{ij}\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

$$\text{Ncut}(A, B) = \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$$

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{ij} w_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2 = \sum_{i \in A, j \in B} w_{ij}\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)^2$$

$$\mathbf{f}^T\mathbf{D}\mathbf{f} = \sum_j d_i \mathbf{f}_i^2 = \sum_{i \in A} \frac{d_i}{\text{vol}(A)^2} + \sum_{j \in B} \frac{d_i}{\text{vol}(B)^2} = \frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}$$

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\min \mathrm{Ncut}(A, B) = \min_{\mathbf{f}} \frac{\mathbf{f}^T \mathbf{L}\mathbf{f}}{\mathbf{f}^T \mathbf{D}\mathbf{f}}$$

$\mathbf{f}^T\mathbf{D}\mathbf{1}=0$, that is, $\Sigma_i d_i f_i = 0$

$$f_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$$

$$\Rightarrow \quad \min_f f^T L f \quad \text{s.t. } f^T D f = 1$$

With relaxation on **f** from discrete to continuous space

Once this is solved, we can obtain cluster labels by the sign of f_i ($f_i > 0$ or $f_i < 0$)

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\min \mathrm{Ncut}(A, B) = \min_{\mathbf{f}} \frac{\mathbf{f}^T \mathbf{L}\mathbf{f}}{\mathbf{f}^T \mathbf{D}\mathbf{f}}$$

$\mathbf{f}^T\mathbf{D}\mathbf{1}=0$, that is, $\Sigma_i d_i f_i = 0$

$$f_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$$

$$\Rightarrow \quad \min_f f^T L f \quad \text{s.t. } f^T D f = 1$$

Now let $u=D^{1/2}f$, then $u^T=f^TD^{1/2}$, thus the constraint becomes:

$$f^T D^{\frac{1}{2}} D^{\frac{1}{2}} f = 1$$

$$u^T u = 1$$

The objective thus becomes

$$f^T L f = f^T D^{1/2} D^{-1/2} L D^{-1/2} D^{1/2} f$$

$$= u^T D^{-1/2} L D^{-1/2} u$$

Let $L'=D^{-1/2}LD^{-1/2}$, the optimization problem becomes:

$$\min_u u^T L' u$$

$$\text{s.t. } u^T u = 1$$

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\min_u u^T L' u$$
$$\text{s.t. } u^T u = 1$$

Looks familiar?

Objective function:

$$\max_{w:\|w\|\leq 1} \mathbf{w}^T \hat{\mathbf{R}} \mathbf{w}$$

PCA proof

The eigen decomposition of L' will give us the solution.
As we are minimizing, we will sort eigenvalues from small to large.
However, can we use the smallest eigenvalue?

Observation. L has an eigenvalue (smallest) which is 0, corresponding to an eigenvector with all 1s.

$$(D - W)\vec{1} = D\vec{1} - W\vec{1} = \vec{0}$$
$$L\vec{1} = 0 \cdot \vec{1}$$

What about L'?

$$L' D^{1/2}\vec{1}$$
$$= D^{-1/2} L D^{-1/2} D^{1/2}\vec{1}$$
$$= D^{-1/2}\vec{0}$$
$$= 0 \cdot \left(D^{1/2}\vec{1}\right)$$

L' also has a 0 eigenvalue and the corresponding eigenvector is $D^{1/2}\vec{1}$

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\min_u u^T L' u$$
$$\text{s.t. } u^T u = 1$$

Looks familiar?

Objective function:

$$\max_{w:\|w\|\leq 1} \mathbf{w}^T \hat{\mathbf{R}} \mathbf{w}$$

PCA proof

The eigen decomposition of L' will give us the solution.
As we are minimizing, we will sort eigenvalues from small to large.
However, can we use the smallest eigenvalue?

If we take this eigenvalue, u=$D^{1/2}\vec{1}$
What is f?

u=$D^{1/2}$f
f=$D^{-1/2}$u=$D^{-1/2}D^{1/2}$ **1** = **1**

This f is not informative for us to find cluster assignment for the nodes!

What about L'?

$$\begin{aligned} & L' D^{1/2}\vec{1} \\ &= D^{-1/2} L D^{-1/2} D^{1/2}\vec{1} \\ &= D^{-1/2}\vec{0} \\ &= 0 \cdot \left(D^{1/2}\vec{1} \right) \end{aligned}$$

L' also has a 0 eigenvalue and the corresponding eigenvector is $D^{1/2}\vec{1}$

Spectral clustering: solving the objective

Objective of spectral clustering: minimize normalized cut

$$\min_u u^T L' u$$
$$\text{s.t. } u^T u = 1$$

Looks familiar?

Objective function:

$$\max_{w:\|w\|\leq 1} \mathbf{w}^T \hat{\mathbf{R}} \mathbf{w}$$

PCA proof

The eigen decomposition of L' will give us the solution.
As we are minimizing, we will sort eigenvalues from small to large.
However, can we use the smallest eigenvalue?

If we take this eigenvalue, u=$D^{1/2}\vec{1}$
What is f?

$u=D^{1/2}f$
$f=D^{-1/2}u=D^{-1/2}D^{1/2}$ **1** = **1**

This f is not informative for us to find cluster assignment for the nodes!

We will use the **2nd smallest** eigenvalue of L' and its corresponding eigenvector to obtain u and f, and then the cluster assignment.

Spectral clustering: partition a graph into 2 clusters

Objective of spectral clustering: minimize normalized cut

$$\text{Ncut}(A, B) = \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$
$$= \sum_{i \in A, j \in B} w_{ij}\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

$$\min_f f^T L f \quad \text{s.t. } f^T D f = 1$$

$$\min_u u^T L' u$$
$$\text{s.t. } u^T u = 1$$

Solution:

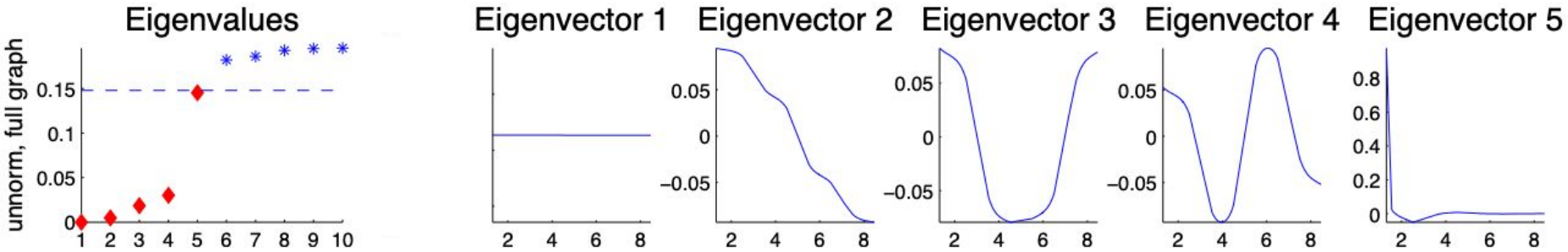
1. Calculate the eigenvalues and eigenvectors of L'
2. u = the 2nd smallest eigenvalue
3. $f = D^{-1/2}u$
4. Apply a threshold on f to determine the cluster assignment of nodes.

Other versions of spectral clustering: Perform eigen-decomposition on L or other versions of L.

From 2-way partition to k-way partition

Cluster multiple eigenvectors

- Build a reduced space from multiple eigenvectors.
- (its like doing dimension reduction then k-means)



- 1st Eigenvector is the all ones vector ***1*** (if graph is connected)
- 2nd Eigenvector thresholded at 0 separates first two clusters from last two
- k-means clustering of the 4 eigenvectors identifies all clusters

Figure from Ulrike von Luxburg 2007

Spectral clustering: steps

Objective of spectral clustering: minimize normalized cut

$$\text{Ncut}(A, B) = \text{cut}(A, B)\left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)}\right)$$

Steps:

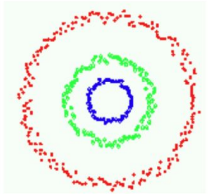
1. Calculate D, $D(i,i) = \sum_j W(i,j)$
2. Calculate L, L = D - W, and L', L'= $D^{-1/2}LD^{-1/2}$
3. Calculate eigenvalues and eigenvectors of L'. Sort the eigenvalues from low to high values.
4. Take the corresponding first k eigenvectors and obtain $U = \left[u_1 \; u_2 \; \ldots \; u_k \right]_{\text{nxk}}$
5. Perform k-means on these eigenvectors and obtain clusters

There are different normalized forms of L

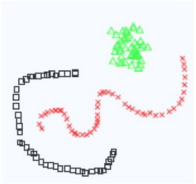
Normalize each row of U

Spectral clustering: more fun facts

- If graph is connected, first Laplacian evec is constant (all 1s)
- If graph is disconnected (k connected components), Laplacian is block diagonal and first k Laplacian evecs are:



OR



$$L = \begin{bmatrix} L_1 & & 0 \\ & L_2 & \\ 0 & & L_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

First three eigenvectors

First 3 eigenvalues are 0s!

Slide credit: Aarti Singh

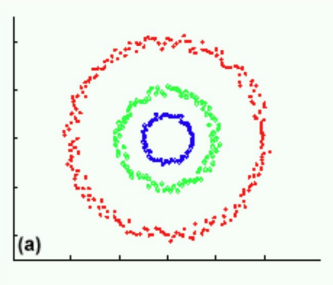
Spectral clustering: why it works

Steps:

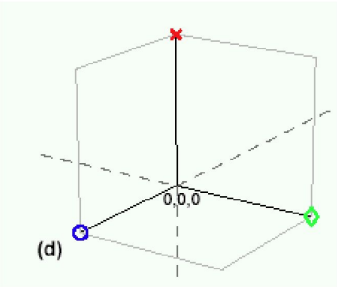
1. Calculate D, $D(i,i) = \sum_j W(i,j)$
2. Calculate L, L = D - W
3. Calculate eigenvalues and eigenvectors of L. Sort the eigenvalues from low to high values.
4. Take the corresponding first k eigenvectors
5. Perform k-means on these eigenvectors and obtain clusters

Non-linear embedding

Original data

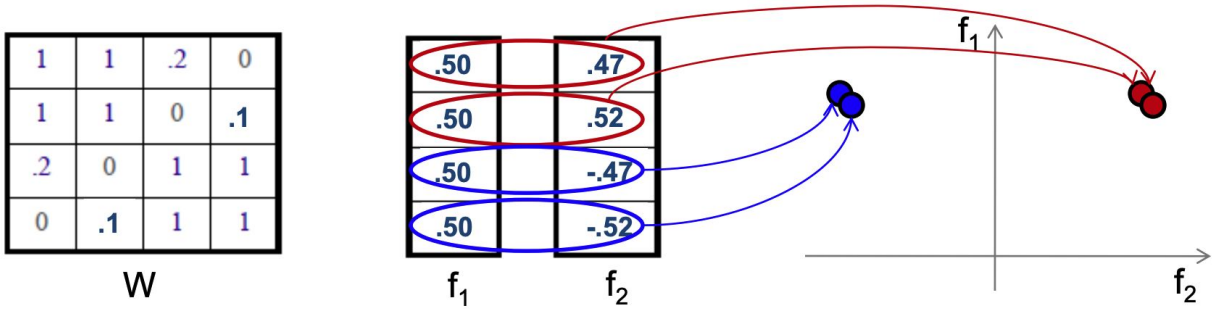


Projected data



Spectral clustering: why it works

Can put data points into blocks using eigenvectors:



Non-linear embedding

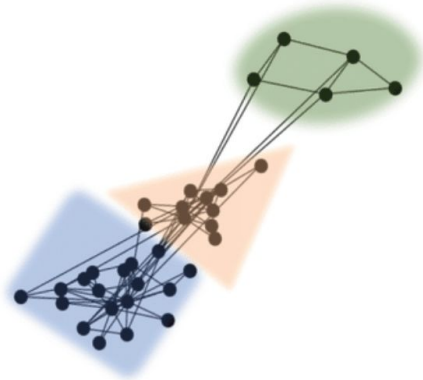
Slide credit: Aarti Singh

Graph based clustering methods

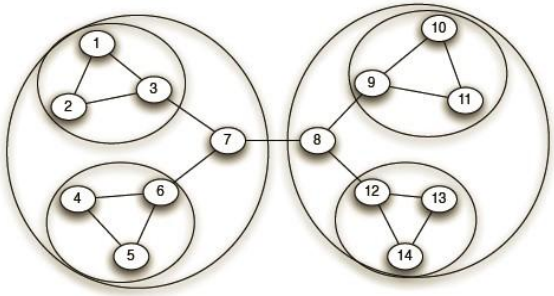
Spectral clustering

Louvain/Leiden clustering methods:

Optimize *Modularity*



Louvain clustering



Objective function

Maximize *modularity*

Sum over communities:

$$\frac{1}{2m}\sum_c (e_c - \gamma\frac{K_c^2}{2m})$$

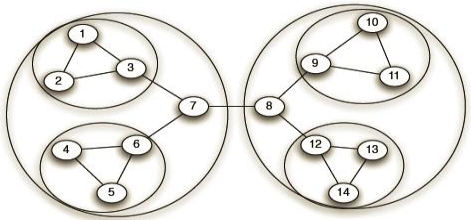
Kc: the sum of the degrees of the nodes in community *c* (volume of *c*)
e_c: number of edges in c. (doesn't include edges between a node in c and a node outside c)

Louvain clustering

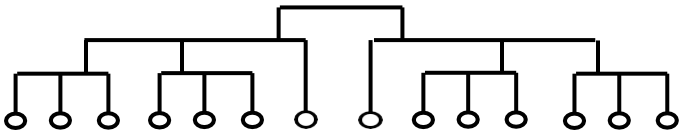
- **Greedy algorithm** for community detection
  - $O(n \log n)$ run time
- Supports weighted graphs
- Provides hierarchical communities

- Widely utilized to **study large networks** because:
  - Fast
  - Rapid convergence
  - High modularity output

Network and communities:



Dendrogram:



"Fast unfolding of communities in large networks" Blondel et al. (2008)

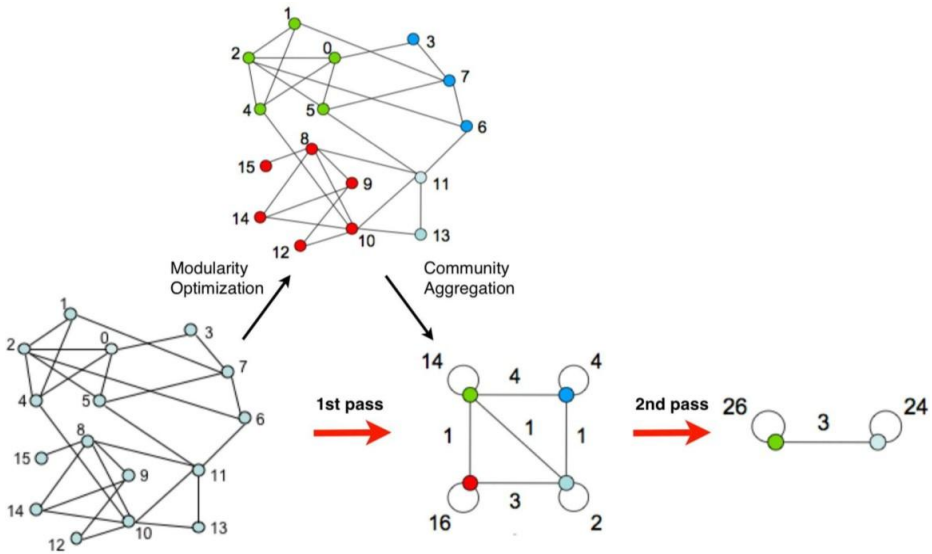
Slide credit: Jure Leskovec

Louvain clustering: high-level algorithm

Iterative algorithm

- Phase 1 - Modularity optimization: make local changes to the community (cluster) memberships of nodes
- Phase 2 - Community aggregation: a community is aggregated into a super-node and we build a new network

Repeat Phases 1 and 2 until the modularity score stops increasing.



"Fast unfolding of communities in large networks" Blondel et al. (2008)

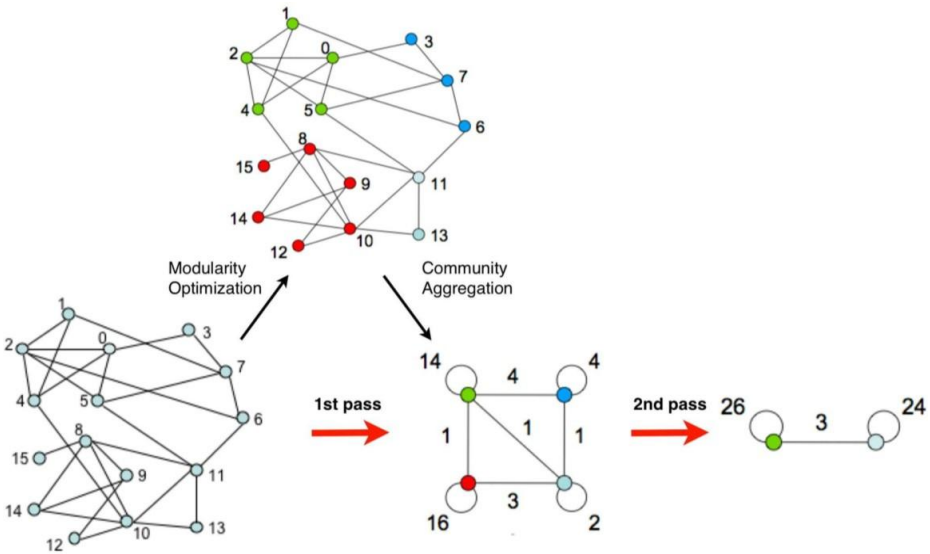
Louvain clustering: phase 1

1. Initially, Put each node in a graph into a distinct community (one node per community)
2. For each node i, the algorithm performs two calculations:
   - Compute the modularity delta (ΔQ) when putting node i into the community of some neighbor j
   - Move i to a community of node j that yields the largest gain in ΔQ
3. Step 2 is repeated for all the nodes in a sequential manner. This is called one iteration. The iteration is also repeated till no improvement in modularity can be achieved.
   - It stops when a local maxima of the modularity is attained, i.e., when no individual node move can improve the modularity.
   - Note that the output of the algorithm depends on the order in which the nodes are considered.
   - Research indicates that the ordering of the nodes does not have a significant influence on the overall modularity that is obtained.

Slide credit: Jure Leskovec

Louvain clustering: phase 2

- The communities obtained in the first phase are contracted into *super-nodes*, and the network is created accordingly:
  - Super-nodes are connected if there is at least one edge between the nodes of the corresponding communities
  - The weight of the edge between the two super-nodes is the sum of the weights from all edges between their corresponding communities
- Phase 1 is then run on the new super-node network



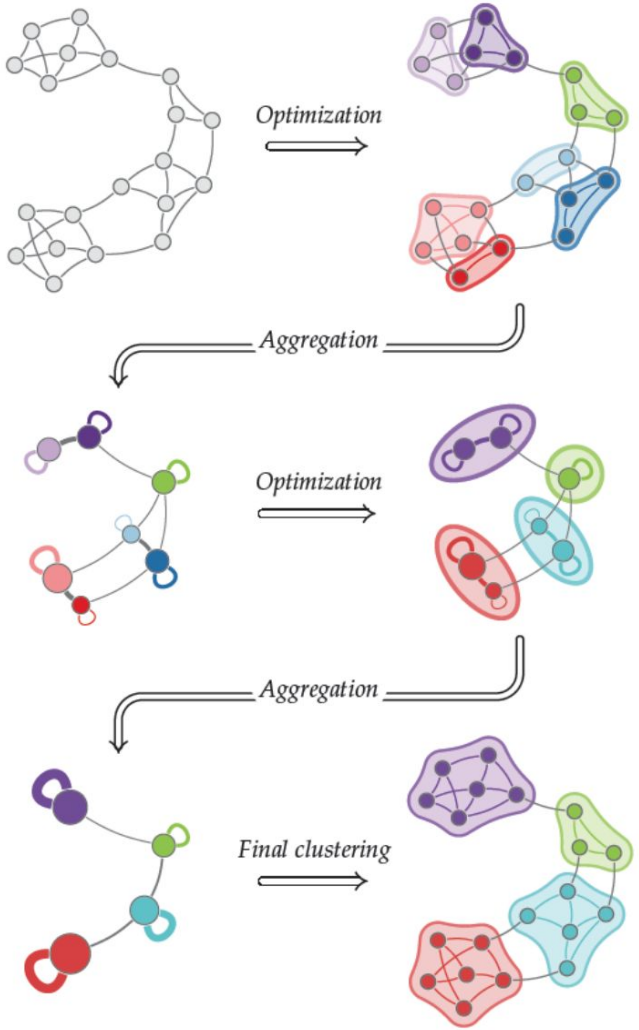
Louvain clustering

Repeat phase 1 and phase 2, until no more moves can be made.

Resolution parameter:

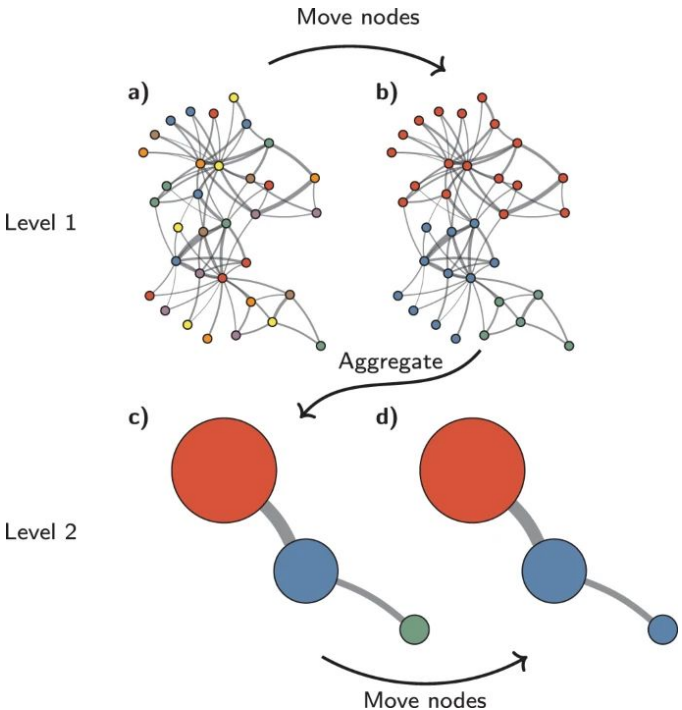
$$\frac{1}{2m}\sum_c (e_c - \gamma\frac{K_c^2}{2m})$$

High resolution → more clusters
Low resolution → less clusters

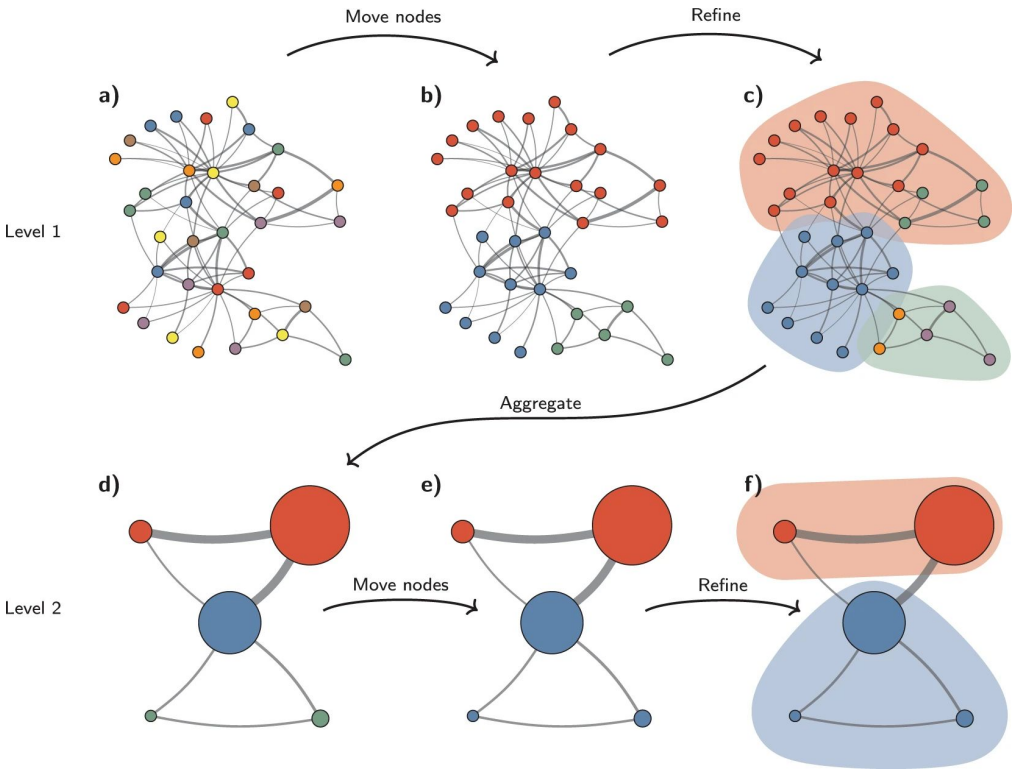


Leiden clustering

Louvain



Leiden



Traag et al, 2019

References

- Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 1997, pp. 731-737, doi: 10.1109/CVPR.1997.609407.
- Barton, T., Bruna, T. & Kordik, P. Chameleon 2: An Improved Graph-Based Clustering Algorithm. *ACM Trans. Knowl. Discov. Data* **13**, 1–27 (2019)
- von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416 (2007)
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008)
- Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019)