When Does Bottom-up Beat Top-down in Hierarchical Community Detection?

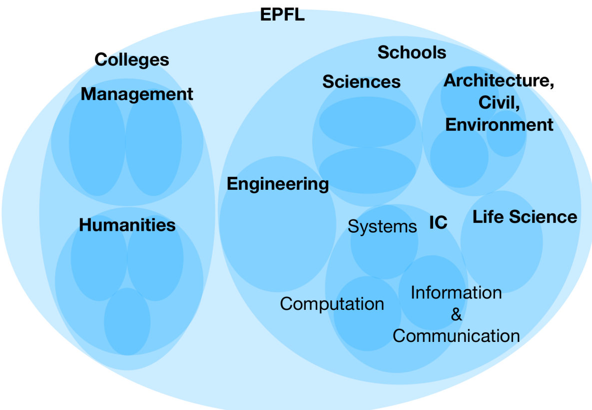
Maximilien Dreveton Daichi Kuroda Matthias Grossglauser Patrick Thiran

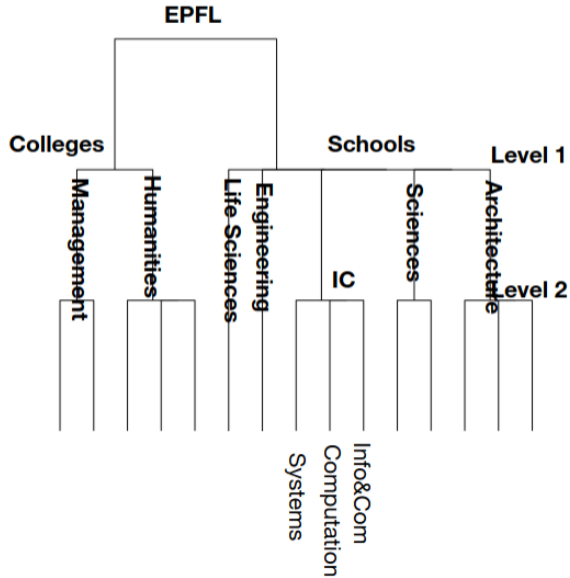
EPFL

Bernoulli-IMS World Congress in Probability and Statistics 2024
August 16, 2024

Table of contents

Hierarchical Community Detection (HCD)





Top-down (divisive) algorithms

Algorithm 0: *Top-down* hierarchical community detection: recursive bi-partitioning

1. Apply a *detection rule* to see if the graph contains at least 2 communities;
2. If step 1 is yes, split the network into two communities by a *bipartition clustering algorithm*;
3. Recursively repeat steps 1 and 2 with the 2 subgraphs composed of the detected communities as inputs. Continue until there are no more communities to split.

Remarks

- *Detection rule*: spectrum of Bethe-Hessian or non-backtracking matrix [1];
- *Bi-partition clustering algorithm*: spectral clustering [2].

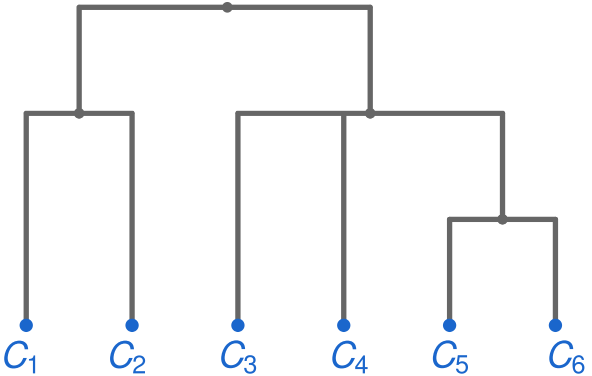
Main reference for top-down HCD: Li, Lei, Bhattacharyya, Van den Berge, Sarkar, Bickel, Levina (2022). Hierarchical Community Detection by Recursive Partitioning. *Journal of the American Statistical Association*.

[1] Le, Levina (2022). Estimating the number of communities by spectral methods. *Electronic Journal of Statistics*, 16(1), 3315-3342.
[2] Von Luxburg (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17, 395-416.

Top-down (divisive) algorithms

Step-by-step example

Recursive algo: 1) Are there communities? 2) If yes, bipartition.



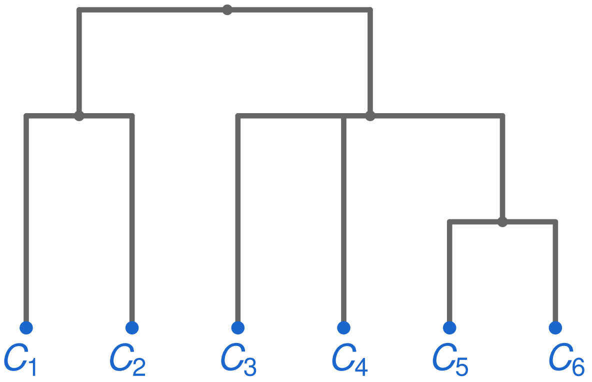
Correct tree

predicted tree

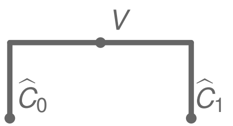
Top-down (divisive) algorithms

Step-by-step example

Recursive algo: 1) Are there communities? 2) If yes, bipartition.



Correct tree

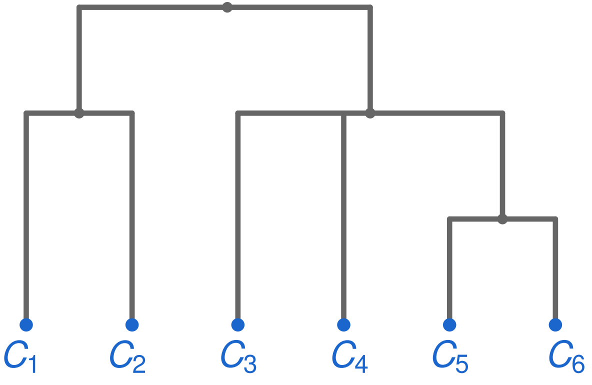


predicted tree

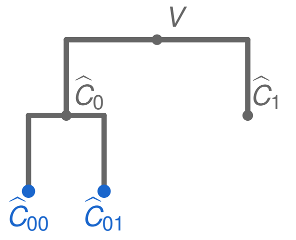
Top-down (divisive) algorithms

Step-by-step example

Recursive algo: 1) Are there communities? 2) If yes, bipartition.



Correct tree

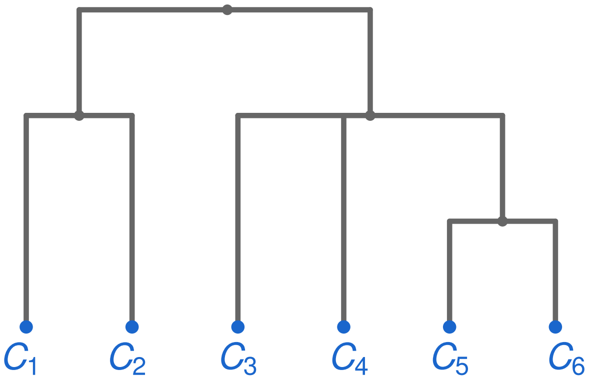


predicted tree

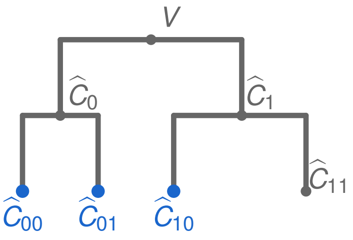
Top-down (divisive) algorithms

Step-by-step example

Recursive algo: 1) Are there communities? 2) If yes, bipartition.



Correct tree

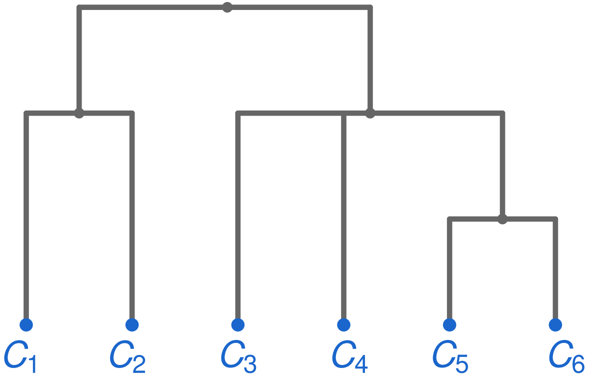


predicted tree

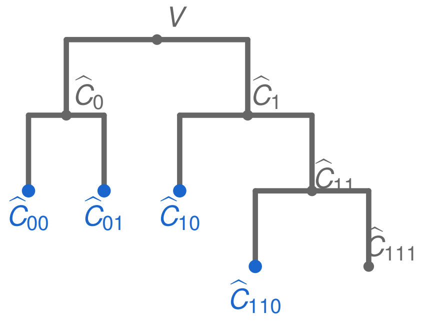
Top-down (divisive) algorithms

Step-by-step example

Recursive algo: 1) Are there communities? 2) If yes, bipartition.

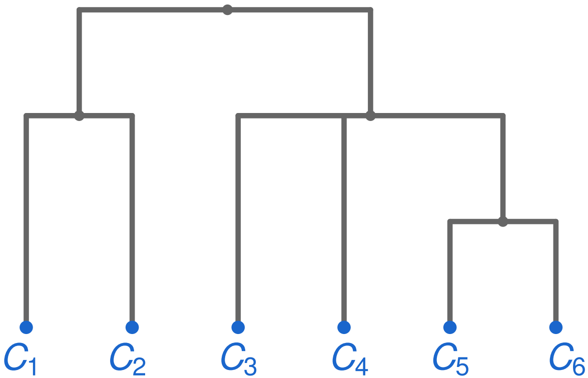


Correct tree

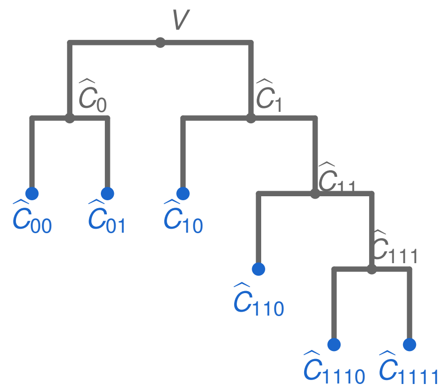


predicted tree

Recursive algo: 1) Are there communities? 2) If yes, bipartition.



Correct tree



predicted tree

Bottom-up (agglomerative) algorithms

Algorithm 1: *Bottom-up* hierarchical community detection.

Input: Graph $G = (V, E)$, flat graph clustering algorithm ALGO.

Process:

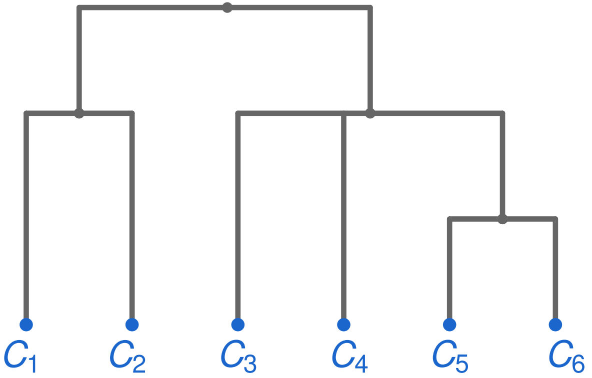
1. Apply a graph clustering algorithm to infer $\widehat{K}$ bottom communities $\widehat{\mathcal{C}} = \left(\widehat{C}_1, \cdots, \widehat{C}_{\widehat{K}}\right)$;
2. Compute the *edge density* between any pairs of communities as $\rho\left(\widehat{C}_a, \widehat{C}_b\right) = \frac{\sum_{i \in \widehat{C}_a, j \in \widehat{C}_b} A_{ij}}{|\widehat{C}_a| \cdot |\widehat{C}_b|}$;
3. Find $a^*, b^* = \underset{1 \le a < b \le \widehat{K}}{\arg\max}\, \rho(\widehat{C}_a, \widehat{C}_b)$ and merge to a *super-community* $\widehat{C}_{a^* \cup b^*} = \widehat{C}_{a^*} \cup \widehat{C}_{b^*}$;
4. Recompute the edge density between the new super community and the other remaining communities by *average-linkage*:

$$\rho\left(\widehat{C}_{a \cup b}, \widehat{C}_c\right) = \frac{\left|\widehat{C}_a\right|}{\left|\widehat{C}_{a \cup b}\right|} \rho\left(\widehat{C}_a, \widehat{C}_c\right) + \left(1 - \frac{\left|\widehat{C}_a\right|}{\left|\widehat{C}_{a \cup b}\right|}\right) \rho\left(\widehat{C}_b, \widehat{C}_c\right). \quad (1.1)$$

5. Repeat steps 3-4 until only one community remains.

Bottom-up (agglomerative) algorithms

Step-by-step example

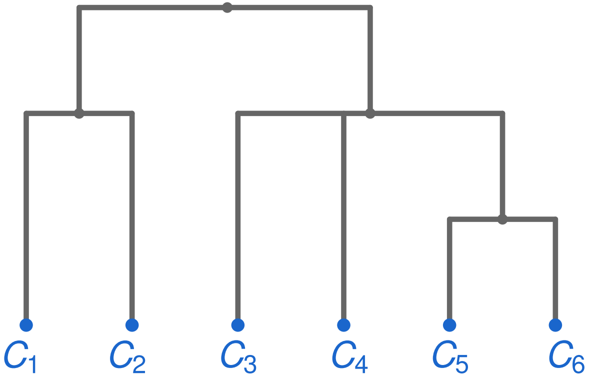


Correct tree

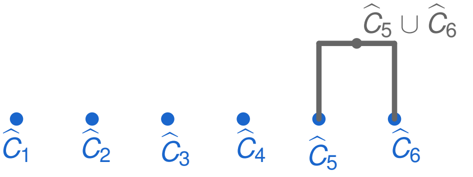
Predicted tree (dendrogram)

Bottom-up (agglomerative) algorithms

Step-by-step example



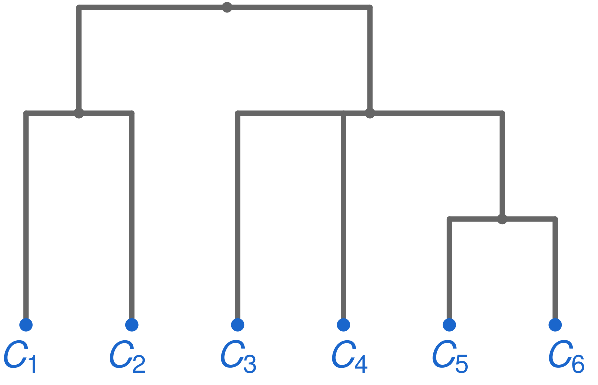
Correct tree



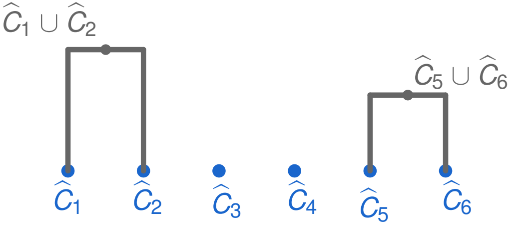
Predicted tree (dendrogram)

Bottom-up (agglomerative) algorithms

Step-by-step example



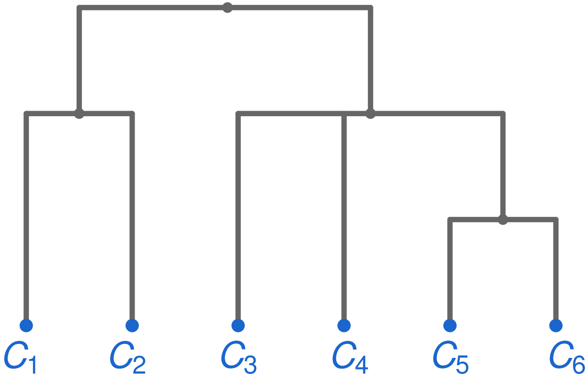
Correct tree



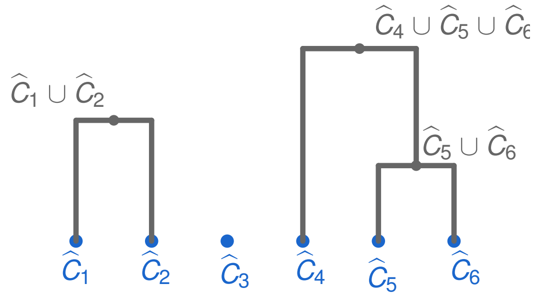
Predicted tree (dendrogram)

Bottom-up (agglomerative) algorithms

Step-by-step example



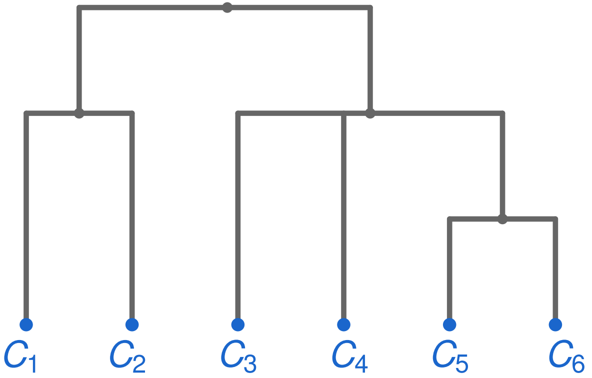
Correct tree



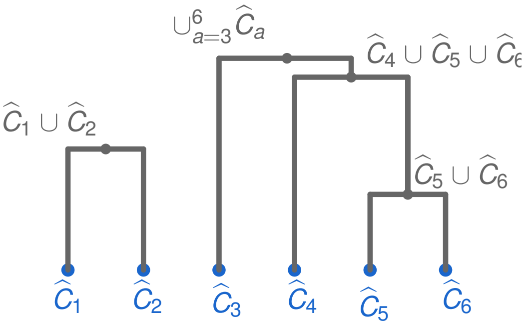
Predicted tree (dendrogram)

Bottom-up (agglomerative) algorithms

Step-by-step example



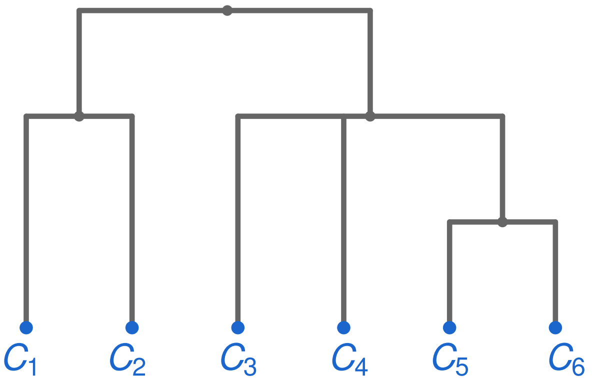
Correct tree



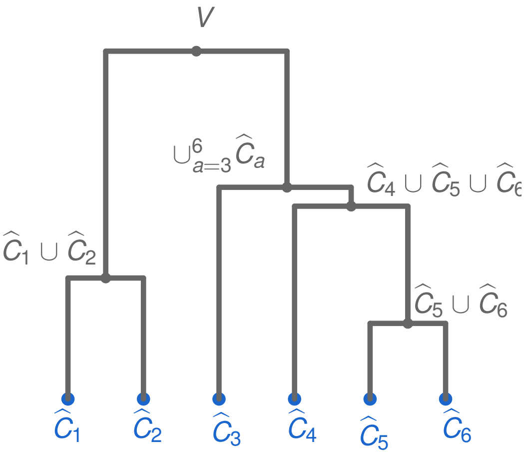
Predicted tree (dendrogram)

Bottom-up (agglomerative) algorithms

Step-by-step example



Correct tree



Predicted tree (dendrogram)

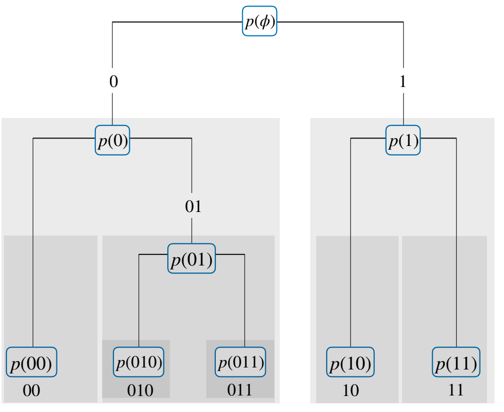
Table of contents

Binary Tree Stochastic Block Model

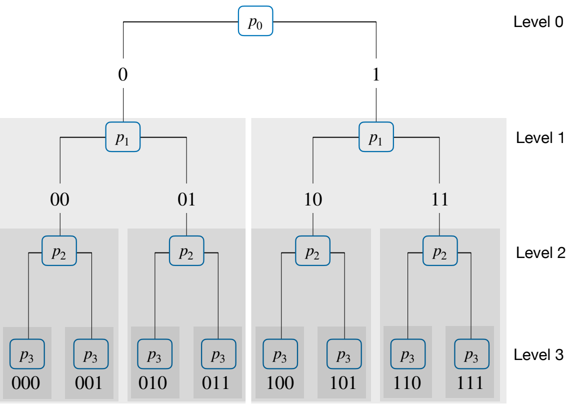
Definition 2.1

Model parameters: n nodes, a rooted binary tree $\mathcal{T}$ whose leaves are $\mathcal{L}_\mathcal{T}$; a function $p\colon \mathcal{T} \to [0,1]$).

1. *Each node $i \in [n]$ is independently assigned to a community C_a ($a \in \mathcal{L}_\mathcal{T}$) uniformly at random;*
2. *Two nodes $i \in C_a$ and $j \in C_b$ are connected with probability $p(\mathrm{lca}(a,b))$.*



(a) Hierarchical stochastic block model.



(b) Binary tree SBM of depth 3.

Recovery regimes

Let $\widehat{\mathcal{C}}$ be an estimator of $\mathcal{C}$. The *fraction of misclassified nodes* by $\widehat{\mathcal{C}}$ is:

$$\operatorname{loss}(\mathcal{C}, \widehat{\mathcal{C}}) \;=\; \frac{1}{n} \min_{\tau \in \mathrm{Sym}(k)} \sum_{i=1}^{n} \mathbb{1}\left\{ i \in C_a \cap \widehat{C}_{\tau(a)}^{c} \right\}.$$

We say that:

- $\widehat{\mathcal{C}}$ is an *exact estimator* of $\mathcal{C}$ if $\mathbb{P}\left(\operatorname{loss}(\mathcal{C}, \widehat{\mathcal{C}}) = 0 \right) = 1 - o(1)$;
- $\widehat{\mathcal{C}}$ is an *almost exact estimator* of $\mathcal{C}$ if $\mathbb{P}\left(\operatorname{loss}(\mathcal{C}, \widehat{\mathcal{C}}) = o(1) \right) = 1 - o(1)$;
- $\widehat{\mathcal{C}}$ is a *weak estimator* of $\mathcal{C}$ is $\mathbb{P}\left(\operatorname{loss}(\mathcal{C}, \widehat{\mathcal{C}}) \leq \eta \right) = 1 - o(1)$ for some $\eta \in (0, \frac{1}{k})$.

Tree recovery from the bottom

Theorem 1 (Dreveton, Kuroda, Grossglauser, Thiran 2023)

Consider a BTSBM whose tree has $K = \Theta(1)$ leaves, and suppose $p_\ell = a_\ell \delta_n$ where a_ℓ is an increasing sequence and $\delta_n \gg \left(\frac{n}{K}\right)^{-2}$. Starting from an estimator $\widehat{\mathcal{C}}$ of the bottom-communities (with $\widehat{K} = K$), bottom-up (average-linkage) correctly recovers the tree $\mathcal{T}$ if one of the following holds:

- *$\widehat{\mathcal{C}}$ is an almost exact estimator of $\mathcal{C}$ and $n\delta_n \gg 1$;*
- *$\widehat{\mathcal{C}}$ is a weak estimator of $\mathcal{C}$, uncorrelated with the graph G, and whose errors are uniformly distributed across all clusters, i.e., for any misclustered node $i \in C_a$, we have $\mathbb{P}(i \in \widehat{C}_b) = \frac{1}{K-1}$ for all $b \neq a$.*

Tree recovery from the bottom

Theorem 1 (Dreveton, Kuroda, Grossglauser, Thiran 2023)

Consider a BTSBM whose tree has $K = \Theta(1)$ leaves, and suppose $p_\ell = a_\ell \delta_n$ where a_ℓ is an increasing sequence and $\delta_n \gg \left(\frac{n}{K}\right)^{-2}$. Starting from an estimator $\widehat{\mathcal{C}}$ of the bottom-communities (with $\widehat{K} = K$), bottom-up (average-linkage) correctly recovers the tree $\mathcal{T}$ if one of the following holds:

- *$\widehat{\mathcal{C}}$ is an almost exact estimator of $\mathcal{C}$ and $n\delta_n \gg 1$;*
- *$\widehat{\mathcal{C}}$ is a weak estimator of $\mathcal{C}$, uncorrelated with the graph G, and whose errors are uniformly distributed across all clusters, i.e., for any misclustered node $i \in C_a$, we have $\mathbb{P}(i \in \widehat{C}_b) = \frac{1}{K-1}$ for all $b \neq a$.*

Remarks

1. We can obtain almost exact estimators of $\mathcal{C}$ when $n\delta_n \gg 1$ and $K = \Theta(1)$ [1,2];
2. We can obtain weak estimators of $\mathcal{C}$ when $n\delta_n \asymp 1$ above the Kesten-Stigum threshold [1];
3. Previous work required $n\delta_n \gtrsim \sqrt{n \log n}$ for *bottom-up* [3] and $n\delta_n \gtrsim \log n$ for *top-down* [4,5].

[1] Abbe, Sandon (2015). Recovering communities in the general stochastic block model without knowing the parameters. NeurIPS.
[2] Yun, Proutiere (2016). Optimal cluster recovery in the labeled stochastic block model. NeurIPS.
[3] Cohen-Addad, Kanade, Mallmann-Trenn, Mathieu (2019). Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM*.
[4] Li et al (2022) Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*.
[5] Lei, Li, Lou (2021). Consistency of spectral clustering on hierarchical stochastic block models, *arXiv:2004.14531v2*.

Exact recovery at intermediate levels

Suppose $p_k = a_k \frac{\log n}{n}$ where a_k is an increasing sequence. Define

$$J_q = \frac{1}{2^d} \left(\left(\sqrt{a_{q-1}} - \sqrt{a_d}\right)^2 + \sum_{k=1}^{d-q} 2^{k-1} \left(\sqrt{a_{q-1}} - \sqrt{a_{d-k}}\right)^2 \right). \tag{2.1}$$

Theorem 2 (Dreveton, Kuroda, Grossglauser, Thiran 2023)

Let G be an HSBM whose latent binary tree $\mathcal{T}$ has depth $d_{\mathcal{T}} = \Theta(1)$. Let $q \in \{1, \cdots, d_{\mathcal{T}}\}$. Exact recovery of the super-communities at level q is impossible if $J_q < 1$ and possible if $J_q > 1$.

Remarks

- When $q = d_{\tau}$, the quantity I_q reduces to the *Chernoff-Hellinger divergence* [1];
- Ref. [2,3] showed that *top-down* exactly recovers the community at level q if $\min_{\ell \in [q]} J_\ell^{\mathrm{td}} > 1$, where $J_q^{\mathrm{td}} = \frac{1}{2^d} \left(\sqrt{a_d + \sum_{k=1}^{d-q} 2^{k-1} a_{d-k}} - \sqrt{2^{d-q} a_{q-1}} \right)^2$. Since $J_q^{\mathrm{td}} > J_q$, this is a more restrictive condition than Theorem 2.

[1] Abbe, Sandon (2015). Recovering communities in the general stochastic block model without knowing the parameters. NeurIPS.
[2] Li et al (2022). Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association.*
[3] Lei, Li, Lou (2020). Consistency of spectral clustering on hierarchical stochastic block models. arXiv:2004.14531.

TABLE OF CONTENTS

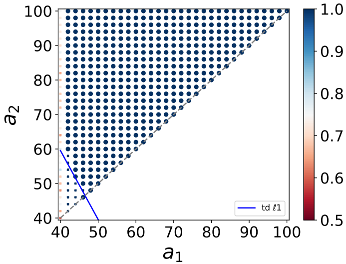
ALGORITHMS

Implementation

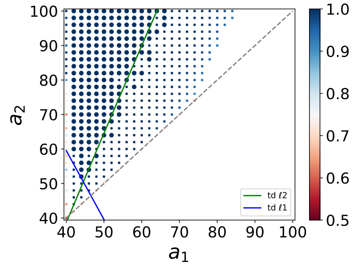
- Top-down: *recursive spectral bi-partitioning*, same as [1]:
  - bi-partitioning: spectral clustering on the normalized Laplacian with a regularization term [2,3,4];
  - selection rule: eigenvectors of the Bethe-Hessian [5];
- Bottom-up:
  - bottom-communities obtained using *spectral clustering with the Bethe-Hessian* [6];
  - hierarchy obtained from the bottom using *average-linkage*.

[1] Li et al. Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association* 117.538 (2022): 951-968.
[2] Joseph, Yu (2016), Impact of Regularization on Spectral Clustering. *The Annals of Statistics*, 44, 1765–1791.
[3] Le, Levina, Vershynin (2017), Concentration and Regularization of Random Graphs. *Random Structures & Algorithms*, 51, 538–561.
[4] Zhang, Rohe (2018). Understanding regularized spectral clustering via graph conductance. *NeurIPS*.
[5] Le, Levina (2022). Estimating the number of communities by spectral methods. Electronic Journal of Statistics, 16(1), 3315-3342.
[6] Dall'Amico, Couillet, Tremblay (2021). A unified framework for spectral clustering in sparse graphs. *The Journal of Machine Learning Research*, 22(1), 9859-9914.

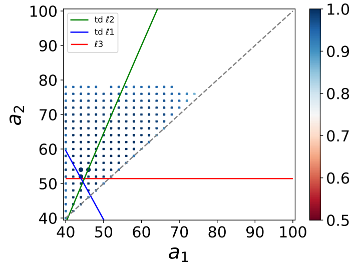
Synthetic data sets



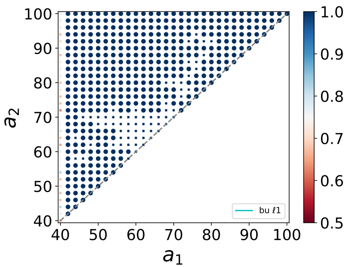
(a) *Top-down* on level 1



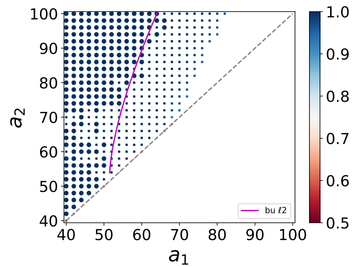
(b) *Top-down* on level 2



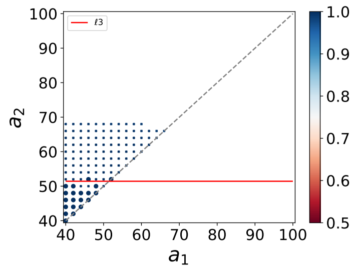
(c) *Top-down* on level 3



(d) *Bottom-up* on level 1



(e) *Bottom-up* on level 2

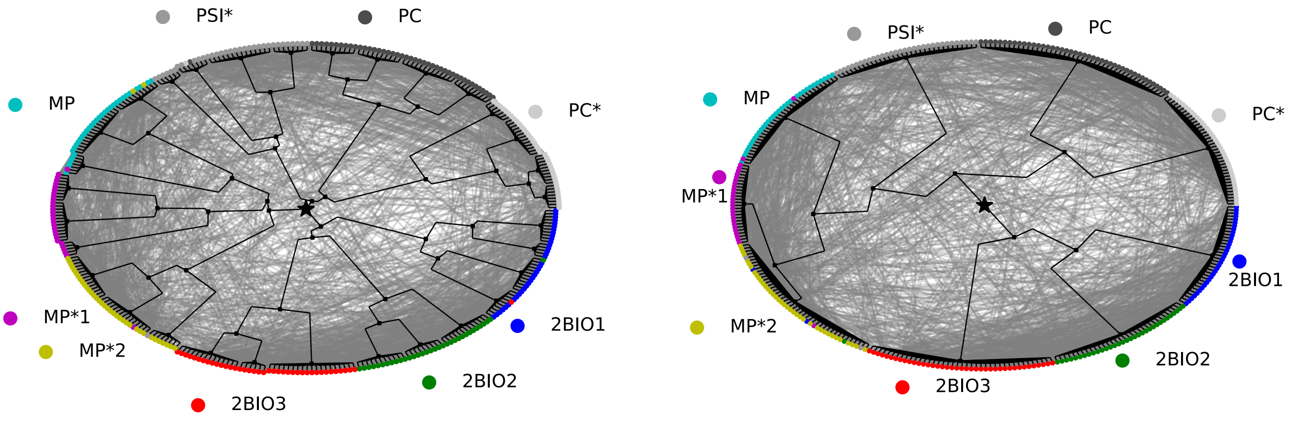


(f) *Bottom-up* on level 3

Figure. BTSBMs with 3 levels, $n = 3200$, interaction probabilities $p_\ell = a_\ell n^{-1} \log n$, with $a_0 = 40$ and $a_3 = 100$.

High school contact data set

Data sets: number of interactions (over a school week) between pair of students in a high school [1].

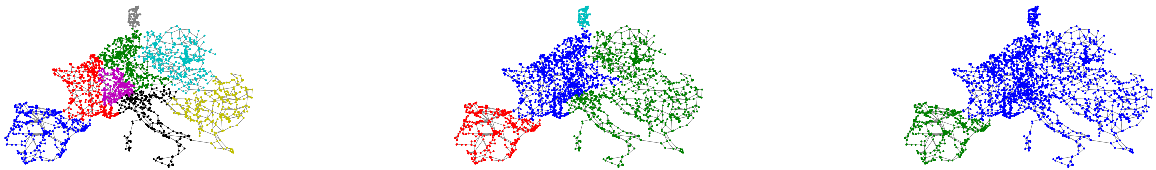


(a) *bottom-up* **(b)** *top-down*

Figure. Output of bottom-up and top-down algorithms on the high school data set. Colours correspond to the true classes, and grey edges indicate contact between the two students. The hierarchical tree is drawn in black, and its root is marked by a star symbol.

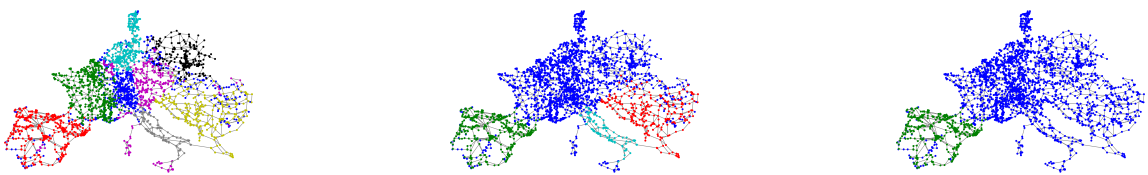
[1] Mastrandrea, Fournet, Barrat (2015). Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLOS ONE*.

European powergrid (1)



(a) 8 communities. **(b)** 4 communities. **(c)** 2 communities.

Figure. Output of *bottom-up* algorithm on the power grid network.



(a) 8 communities. **(b)** 4 communities. **(c)** 2 communities.

Figure. Output of *top-down* algorithm on the *power grid* network.

European powergrid (2)

(a) *Bottom-up*. **(b)** *Top-down*.

Figure. Comparison of the dendrograms obtained by *bottom-up* and *top-down* on *powergrid* data set.

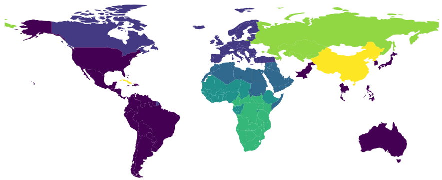
Observations

- Dendrograms obtained by *top-down* algorithms can show *inversions*.
- The inversion phenomenon is known to appear if one cluster data points in $\mathbb{R}^d$ by recursively applying K-Means with $K = 2$ [1,2].

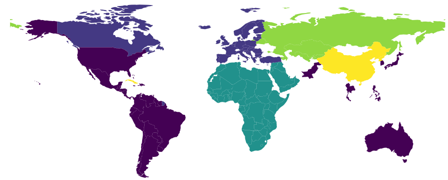
[1] Hastie, Tibshirani, Friedman, Friedman (2009). The elements of statistical learning: data mining, inference, and prediction.
[2] Macnaughton-Smith et al (1964). Dissimilarity analysis: a new technique of hierarchical sub-division. *Nature*.

Military alliance network: Bottom-up

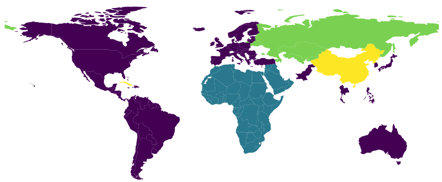
Data from the *Alliance Treaty Obligations and Provisions* (ATOP) project (2018)



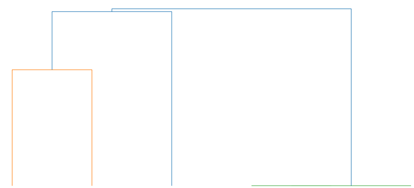
(a) Highest depth.



(b) Middle depth (after 2 merges).



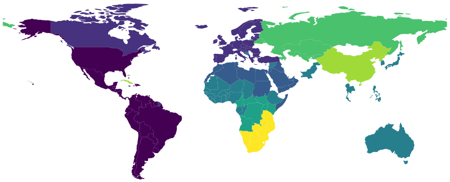
(c) Lowest depth (after 3 merges).



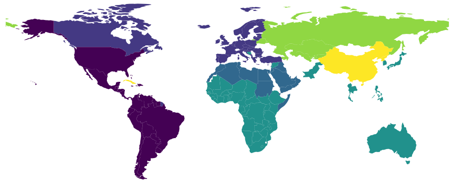
(d) Dendrogram.

Figure. Output of *bottom-up* algorithm on the military alliance network. The dendrogram does not show the disconnected component (China, Cuba, North Korea).

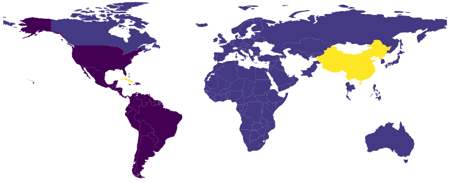
Military alliance network: Top-down



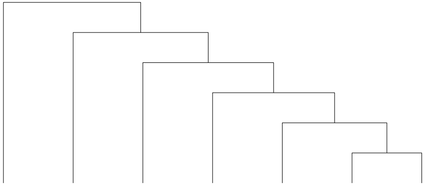
(a) Highest depth (8 clusters).



(b) Middle depth (6 clusters).



(c) Lowest depth (3 clusters).



(d) Dendrogram.

Figure. Output of *top-down* algorithm on the military alliance network. The dendrogram does not show the disconnected component (China, Cuba, North Korea).

Table of contents

Directions of further research

Takeaway message

- Bottom-up via *linkage* is very efficient in recovering the hierarchy (both in theory and practice)
- Hierarchies obtained by top-down suffer from inversion

Future work

- Dealing with trees with infinite depth and different scaling for p_t
- Weak recovery at intermediate levels
- Non-binary hierarchies

Reference: Dreveton, Kuroda, Grossglauser, Thiran (2023). When Does Bottom-up Beat Top-down in Hierarchical Community Detection? arXiv:2306.00833.