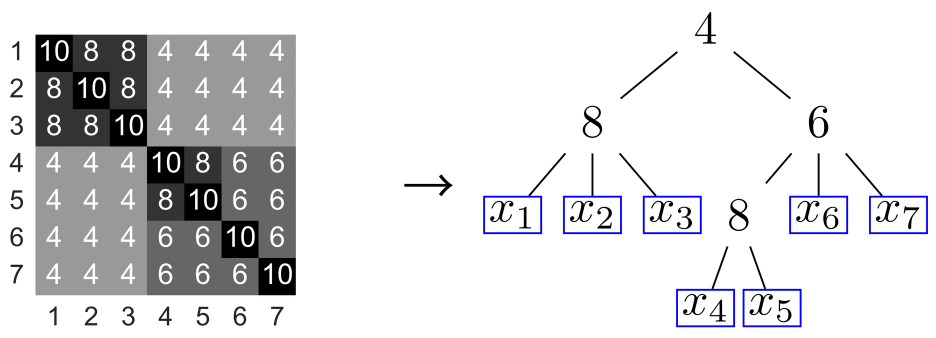
Beyond Binary Trees: Finding General Hierarchies

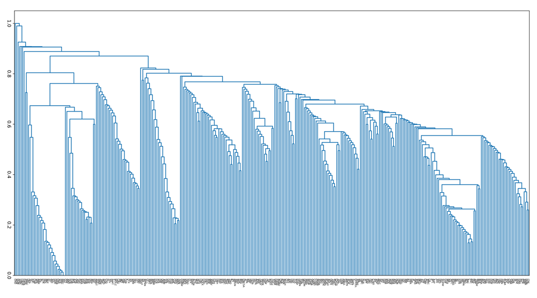
Maximilien Dreveton, Matthias Grossglauser, Daichi Kuroda, Patrick Thiran
INDY (Information and Network Dynamics), EPFL

Hierarchical Clustering

Goal: Finding a good tree representation of the given data ($\to$ similarities) so that tree represents which object is close to which

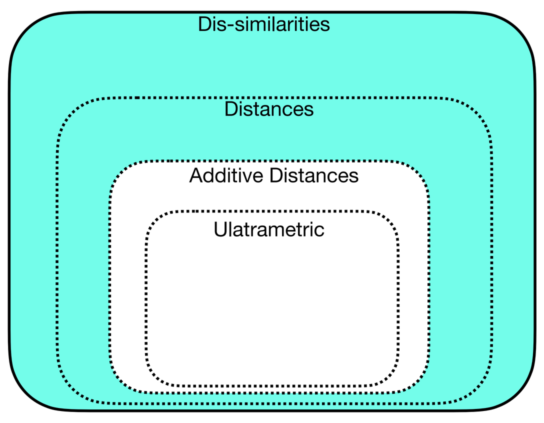


Similarities s — Tree

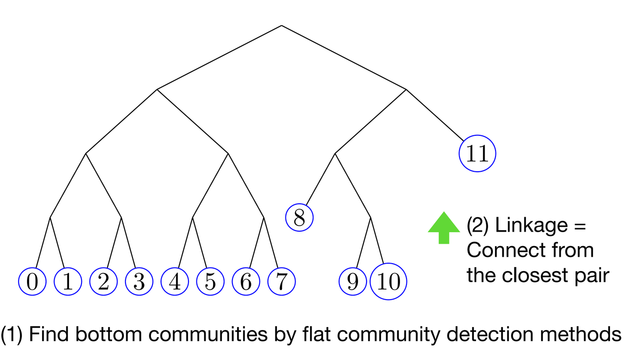


Current Methods and Limitations

Ultrametric Tree & Additive Tree



- Ultrametric Distance: $d(x_1, x_3) \le \max\{d(x_1, x_2), d(x_2, x_3)\}$
- Additive Distance: $d(x_1, x_2) + d(x_3, x_4) \le \max\{d(x_1, x_2) + d(x_2, x_4), d(x_2, x_3) + d(x_1, x_4)\}$
- Triangular Inequality: $d(x_1, x_3) \le d(x_1, x_2) + d(x_2, x_3)$



Linkage

Popular methods

- Linkage [1]
- Dasguptuta Cost [2]
- Top-Down [3]

Limitations

- Overfit to Binary tree
  - A lot of hallucinated levels
  - Cannot Distinguish with/without hierarchy
- Not well-defined outside ultrametric/additive distance

Valid Hierarchies

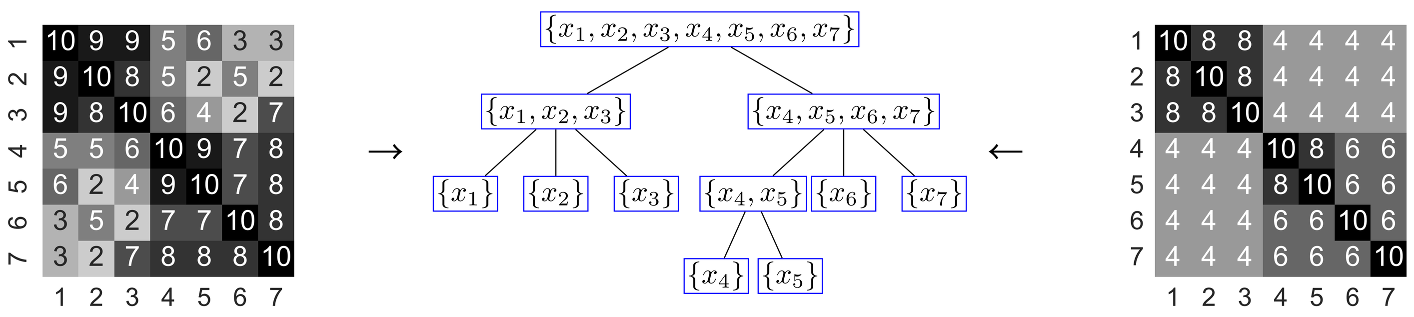
Definition: Valid Hierarchies: $\mathcal{H}(\mathcal{X}, s)$

$T \in \mathcal{H}(\mathcal{X}, s)$ is a tree s.t., from $\forall x_1 \in \mathcal{X}$

- if x_2 is closer* than x_3 on T
  $\to x_2$ is closer than x_3 also w.r.t. $s(\cdot,\cdot)$
- if x_2 and x_3 are equally close* on T
  $\to$ no info on which one is closer w.r.t. $s(\cdot,\cdot)$
  $\not\to$ But NOT: equally close w.r.t. $s(\cdot,\cdot)$

Notation

- $\mathcal{H}(\mathcal{X}, s)$: set of valid hierarchies for $\mathcal{X}$ w.r.t. $s(\cdot,\cdot)$
- $\mathcal{X}$: set of base objects
- $s(\cdot,\cdot)$: similarity measurement for $\mathcal{X} \times \mathcal{X}$

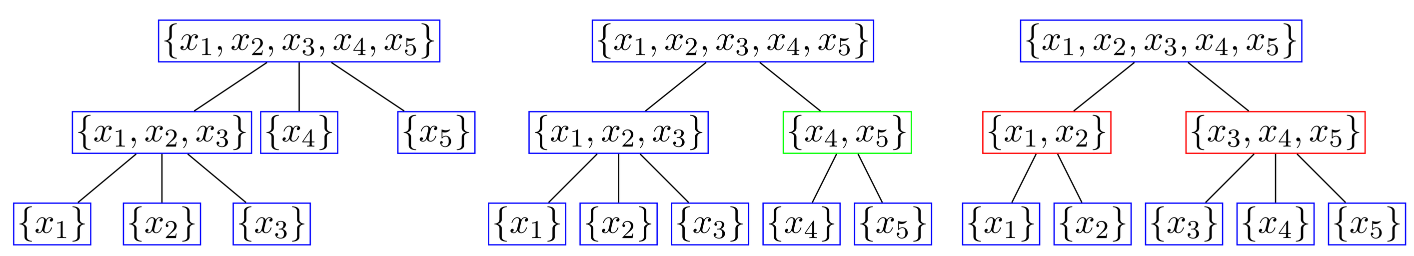


Formally, $T \in \mathcal{H}(\mathcal{X}, s)$ satisfies for any $t \in T$:

$$\min_{x_1, x_2 \in t, x_3 \in \mathcal{X} \setminus t} s(x_1, x_2) > s(x_1, x_3) > 0.$$

There are often multiple trees in $\mathcal{H}(\mathcal{X}, s)$.

Partial Order for Trees



(a) T_1 (b) T_2 (c) T_3

- $\mathcal{X} = \{x_1, x_2, \cdots, x_5\}$
- $T_1 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_1, x_2, x_3\}, \{x_1, x_2, x_3, x_4, x_5\}\}$
- $T_2 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_1, x_2, x_3\}, \{x_4, x_5\}, \{x_1, x_2, x_3, x_4, x_5\}\}$
- $T_3 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_1, x_2, x_3, x_4, x_5\}\}$

$(\mathcal{T}(\mathcal{X}, s), \subsetneq)$ is a partially ordered set.
$\to T_1 \subsetneq T_2$ but $T_1 \not\subseteq T_3$ and $T_3 \not\subseteq T_1$

Most Informative Valid Hierarchy

Most Informative Valid Hierarchy T_*

$(\mathcal{H}(\mathcal{X}, s), \subsetneq)$ is a partially ordered set and it has a greatest element T_*.

$$T_* = \operatorname*{arg\,max}_{T \in \mathcal{H}(\mathcal{X}, s)} |T|$$

- $|T|$: # of vertices of a tree T
- $\mathcal{H}(\mathcal{X}, s)$: the set of valid hierarchies

Properties

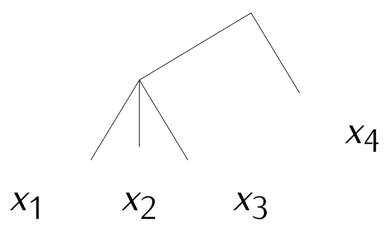
1. $T \subseteq T_*$ for any $T \in \mathcal{H}(\mathcal{X}, s)$ and T_* is uniquely defined for a given $(\mathcal{X}, s)$
2. Flat communities = T_* being a star graph tree
3. It coincides with the ultrametric tree when s is ultrametric
4. $T_* \subset T_{linkage}$ & T_* can be reconstructed by:
   (a) Apply linkage to get $T_{linkage}$
   (b) trimming unqualified verticies of $T_{linkage} \Rightarrow T_*$

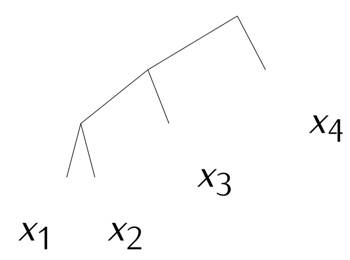
Handling Noise

$T_\epsilon \in \mathcal{H}(\mathcal{X}, s, \epsilon)$ satisfies for any $t \in T_\epsilon$:

$$\min_{x_1, x_2 \in t, x_3 \in \mathcal{X} \setminus t} s(x_1, x_2) > s(x_1, x_3) > \overset{\epsilon}{0}.$$

$$\begin{pmatrix} 3 & 2 & 2 & 1 \\ 2 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix} \qquad \begin{pmatrix} 3 & 2.2 & 1.8 & 1.1 \\ 2.2 & 3 & 1.9 & 0.9 \\ 1.8 & 1.9 & 3 & 1 \\ 1.1 & 0.9 & 1.05 & 4 \end{pmatrix}$$





Similarity s — Similarity $\hat{s}$. Noises $[-0.2, 0.2]$ are added. — $T_*(\mathcal{X}, s)$. $0.4 \le \epsilon < 0.7$ — $\epsilon < 0.4$

Application: Hierarchical Community Detection

Based on [3], propose the following algorithm:

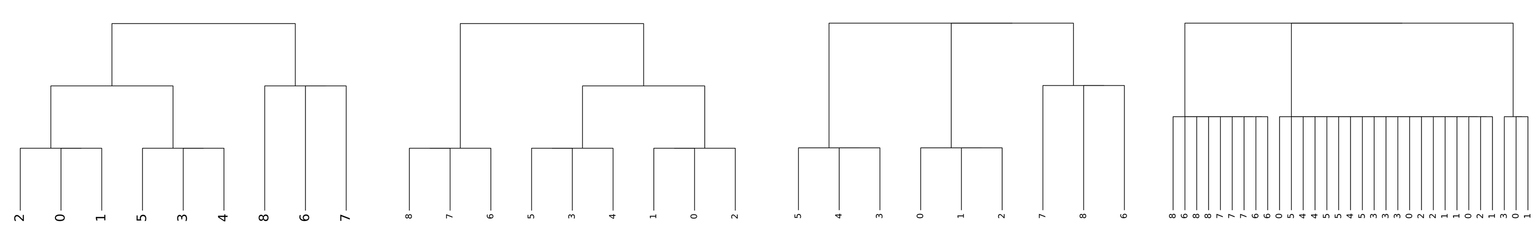
1. Detect bottom communities $\hat{\mathcal{X}}$ by Bethe-Hessian [4]
2. Define $\hat{s} : \hat{\mathcal{X}} \times \hat{\mathcal{X}} \to \mathbb{R}$ as edge densities between $\hat{\mathcal{X}}$
3. Apply average linkage to obtain $\hat{T}_{\text{linkage}}$
4. Trim $t \in \hat{T}_{\text{linkage}}$ that violates ϵ-strong condition and get $\hat{T}$

Table 1: Performance of HCD algorithms on 20 ABCD [5] graphs (communities without hierarchies).

	$\hat{k}$	$\hat{k} = k$	$\rho(z^*, \hat{z})$	$\rho(T^*, \hat{T})$	$\hat{T} = T^*$
Our Algorithm	10 ± 0	20/20	0.97 ± 0.0038	0.97 ± 0.0038	20/20
Nested sEEP [6]	10 ± 0	20/20	0.97 ± 0.0038	0.95 ± 0.051	16/20
Nested DCBM [7]	10 ± 0.77	18/20	0.99 ± 0.0047	0.97 ± 0.077	18/20

Table 2: Performance of HCD algorithms on HDCBMs with all possible shapes of hierarchies with 9 leaves.

	$\hat{k}$	$\hat{k} = k$	$\rho(z^*, \hat{z})$	$\rho(T^*, \hat{T})$	$\hat{T} \simeq T^*$
Our Algorithm	8.4 ± 0.84	62%	0.93 ± 0.10	0.98 ± 0.044	61.6%
Nested sEEP [6]	8.4 ± 0.84	62%	0.93 ± 0.10	0.94 ± 0.0812	36.8 %
Nested DCBM [7]	20 ± 10.3	0.01%	0.81 ± 0.098	0.64 ± 0.154	0.0 %



True hierarchy — Our Algorithm — Nested sEEP [6] — Nested DCBM [7]

Current Limitations & Future Work

1. Choosing a good ϵ
2. Robust-to-outliers practical extension
3. Apply to general clustering context
4. Starting from node of the graph (work in progress)

References

1. M. Dreveton et al., 2023. arXiv: 2306.00833 [cs.SI].
2. S. Dasgupta, *STOC '16*, 2016, pp. 118–127
3. T. Li et al., *Journal of the American Statistical Association*, 117(538):951–968, 2022.
4. L. Dall'Amico et al., *J. Mach. Learn. Res.*, 22:217–1, 2021.
5. B. Kamiński et al., *Network Science*, vol. 9, no. 2, pp. 153–178, 2021.
6. M. T. Schaub et al., *Physical Review E*, 107(5):054305, 2023.
7. T. P. Peixoto, *Physical Review X*, 4.1, ,2014: 011047.