LABORATOIRE D'INFORMATIQUE GASPARD-MONGE
Sous la co-tutelle de :
CNRS
ÉCOLE DES PONTS PARISTECH
ESIEE PARIS
UPEM • UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

INS2I

Ultrametric Fitting by Gradient Descent

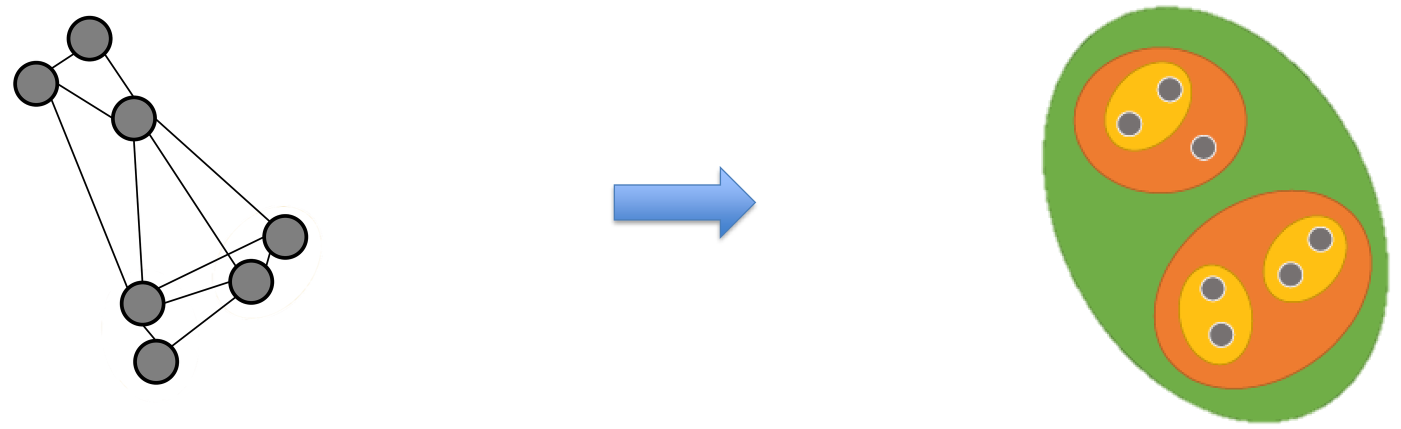
Giovanni Chierchia and Benjamin Perret

LIGM, CNRS, ESIEE Paris, Université Paris Est, France

ESIEE PARIS
UNIVERSITÉ PARIS-EST

Hierarchical clustering

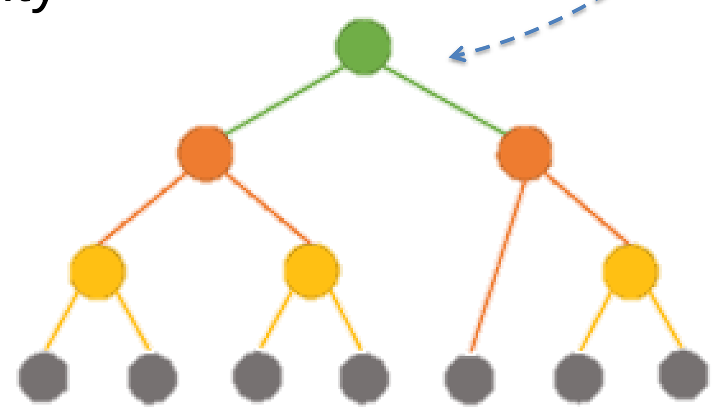
- **Input:** Undirected graph **G(V,E)** with dissimilarity edge weights **w**
- **Output:** Hierarchy of clusters built on the graph nodes



What is an ultrametric?

- Mathematical representation of hierarchical clustering
- Distance that satisfies the ultrametric inequality

$$d(x,x) = 0$$
$$d(x,y) \geq 0$$
$$d(x,y) = d(y,x)$$
$$d(x,y) \leq \max\{d(x,z), d(y,z)\}$$

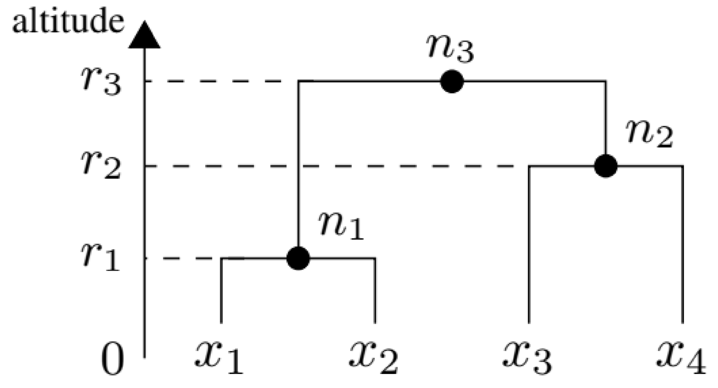


- Intuitively described by a "special tree" called *dendrogram*
- It can be represented as the edge weights of a (connected) graph

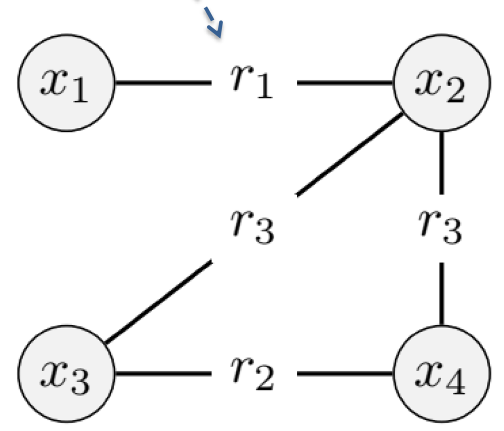
$$u = \left[u(e)\right]_{e\in E} \in \mathbb{R}^{|E|}$$

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & r_1 & r_3 & r_3 \\ x_2 & r_1 & 0 & r_3 & r_3 \\ x_3 & r_3 & r_3 & 0 & r_2 \\ x_4 & r_3 & r_3 & r_2 & 0 \end{array}$$

(a) Ultrametric



(b) Dendrogram



(c) Ultrametric on a graph

Ultrametric fitting

- **Goal:** Find the ultrametric that "best" represents a dissimilarity graph

$$\underset{u\in\mathbb{R}^{|E|}}{\text{minimize}} \quad J(u;w) \quad \text{s.t.} \quad u \text{ is an ultrametric on } \mathcal{G}(V,E)$$

- **Problem:** How to handle the ultrametric constraint?
  - *Agglomerative or divisive heuristics [Dasgupta 2016; Kobren 2017; ...]*
  - *Integer linear programming [Yarkony 2015; Roy 2016; ...]*
  - *Continuous relaxations [Charikar 2017; Monath 2017 & 2019; ...]*
  - *Probabilistic formulations [Vikram 2016; ...]*

Proposed approach

- An edge-weight vector **u** of **G(V,E)** can be transformed into an ultrametric on **G** via a differentiable operation *(e.g., min-max)*

$$(\forall (x,y)\in V^2) \qquad d_u(x,y) = \min_{P\in \text{paths}_{\mathcal{G}}(x,y)} \max_{e'\in P} u(e')$$

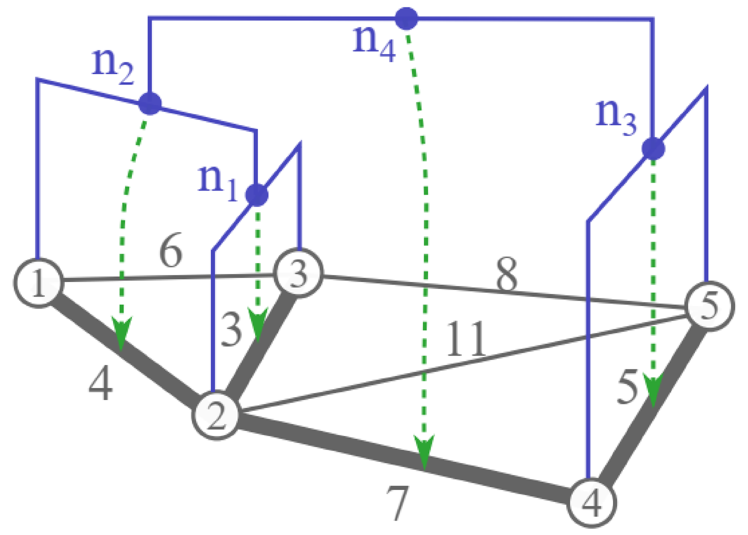
- **Idea:** Replace the constraint with an ultrametric transform

$$\boxed{\underset{u\in\mathbb{R}^{|E|}}{\text{minimize}} \quad J\big(d_u;w\big)}$$

- … and solve the problem with **gradient descent**

$$u^{[t+1]} = u^{[t]} - \tau\nabla_u J(d_{u^{[t]}};w)$$

- Ultrametric transforms are fast
  - *Single linkage → O(N log N)*
  - *Average linkage → O(N²)*
  - *Exponential linkage [Yadav, 2019] → O(N²)*



Cost functions

- **Data fidelity:** Find the closest ultrametric

$$J_{\text{closest}}(d_u;w) = \sum_{e_{xy}\in E}\Big(d_u(x,y) - w(e_{xy})\Big)^2$$

- **Regularization:** Penalize small clusters at high scales

$$J_{\text{size}}(d_u) = \sum_{e_{xy}\in E}\frac{d_u(x,y)}{\gamma(\text{lca}(x,y))}$$

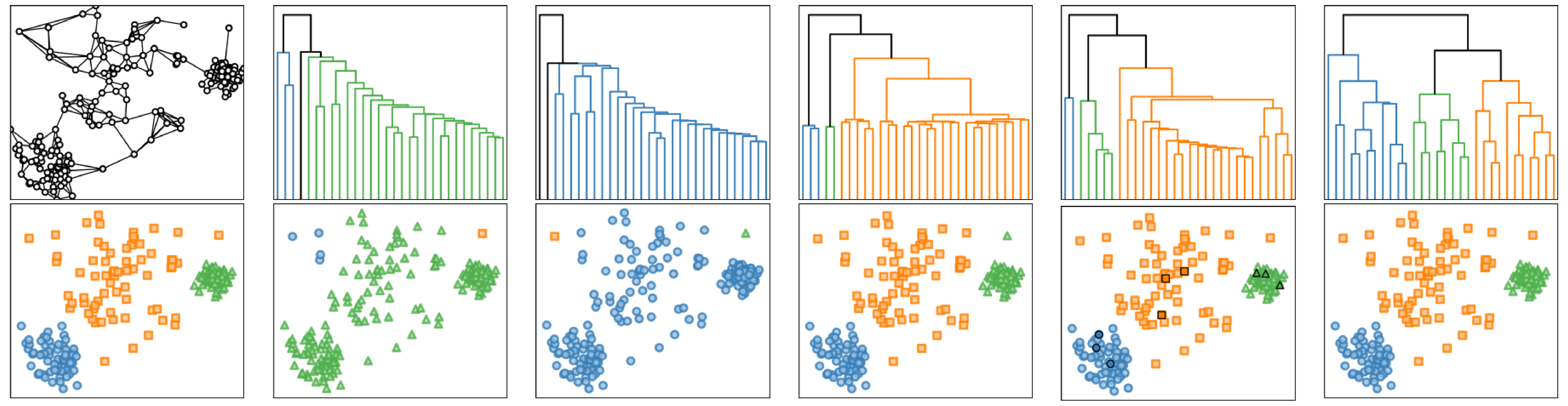
Small for clusters with unbalanced subclusters

- **Semi-supervision:** Take into account triplets of labeled points

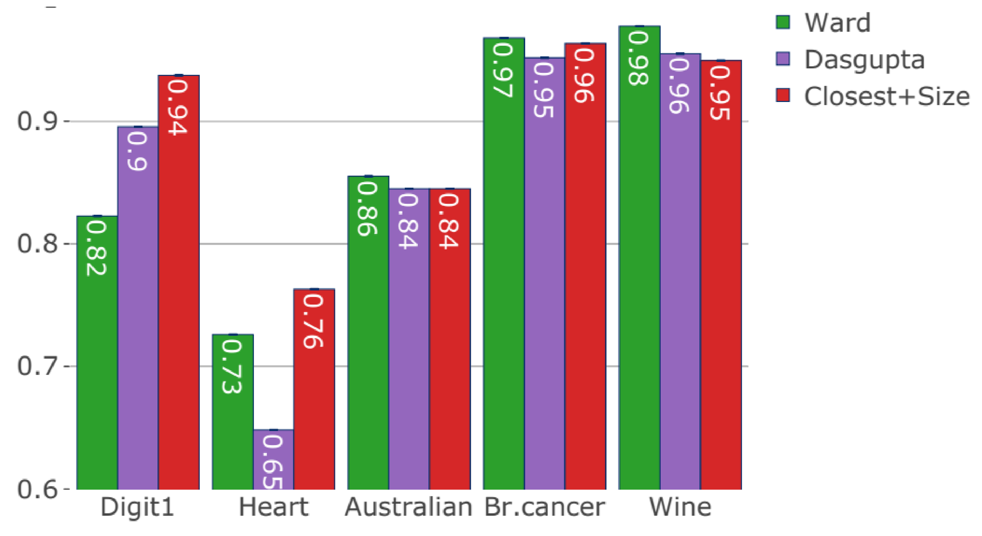
$$J_{\text{triplet}}(d_u) = \sum_{(\text{ref},\text{pos},\text{neg})\in\mathcal{T}} \max\{0, \alpha + d_u(\text{ref},\text{pos}) - d_u(\text{ref},\text{neg})\}$$

- We also propose a relaxation of **Dasgupta cost function**

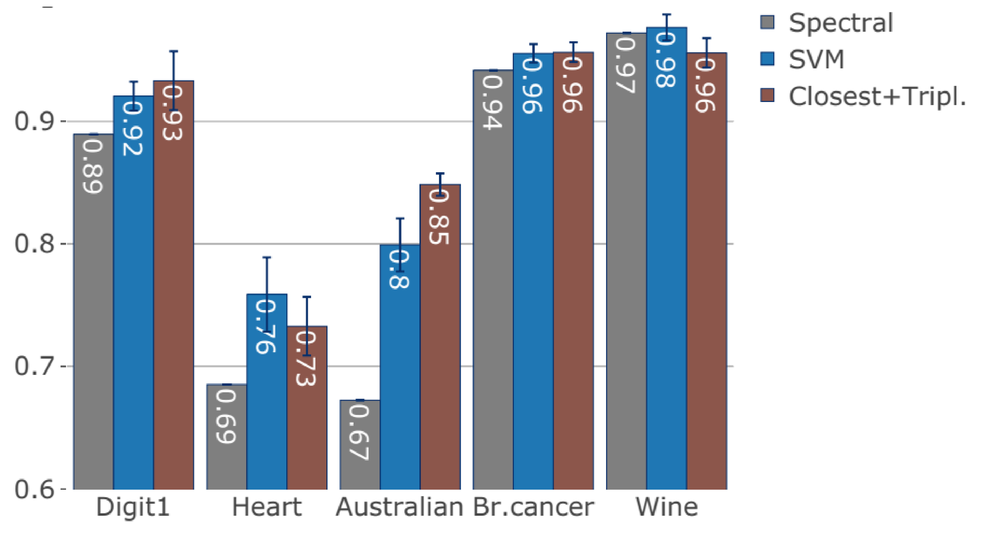
Numerical results



(a) Graph/Labels (b) Average link (c) Closest (d) Closest+Size (e) Closest+Triplet (f) Dasgupta



(a) Hierarchical clustering (unsupervised)



(b) Semi-supervised clustering