# A $(1+\epsilon)$-Approximation for Ultrametric Embedding in Subquadratic Time

LaBRI

Gabriel Bathie, Guillaume Lagarde
Univ. Bordeaux, LaBRI & DI ENS PSL, France

## Ultrametrics

**Ultrametric:** *metric + ultrametric inequality.*

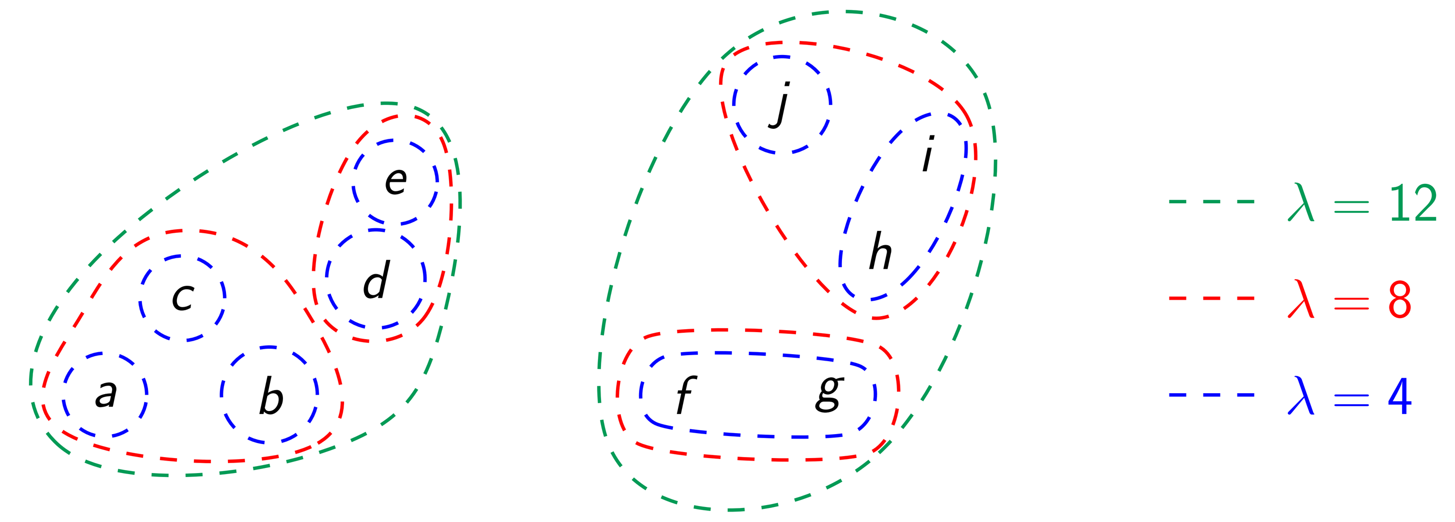
$\forall x, y, z, \Delta(x,y) \geq 0$ $\quad \Delta(x,y) = 0 \iff x = y$

$\Delta(x,y) = \Delta(y,x)$ $\quad \Delta(x,y) \leq \Delta(x,z) + \Delta(z,y)$

Ultrametric: $\Delta(x,y) \leq \max(\Delta(x,z), \Delta(z,y))$

## Ultrametrics and Hierarchical Clustering

**Ultrametrics:** Mathematical formalization of Hierarchical Clustering




## Ultrametric Embedding

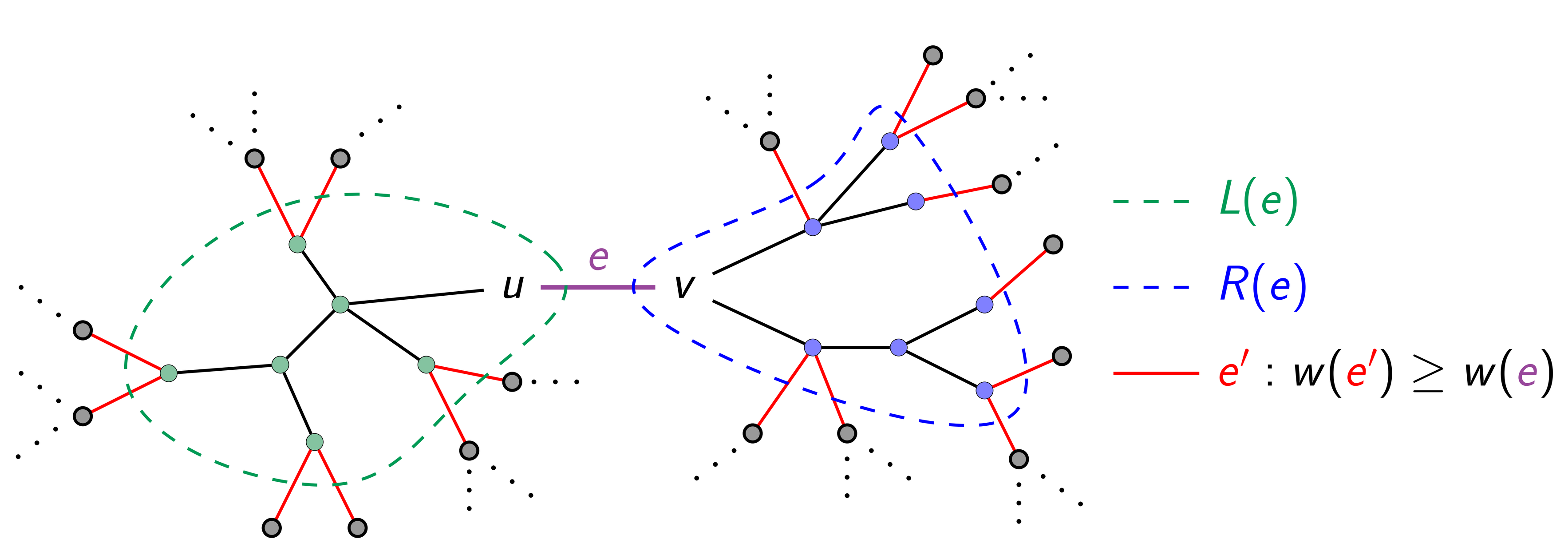
**Goal**: Given a metric space $(X, d)$, find the ultrametric $\Delta$ that agrees the most with $d$.
Minimize the **distortion** $\alpha$:

$$\forall x, y, \Delta(x,y) \leq d(x,y) \leq \alpha \cdot \Delta(x,y).$$

**Theorem**: In the **Euclidean case**, for any $c > 1$, we can compute a $c$-approximation of the best ultrametric embedding in time $\mathcal{O}\left(n^{1+1/c}\right)$.

$$\forall x, y, \Delta(x,y) \leq \ell_2(x,y) \leq c \cdot \alpha \cdot \Delta(x,y).$$

## Cut weights




$$CW(e) = \max_{x \in L(e), y \in R(e)} d(x,y)$$

**$\alpha$-Approx. cut weights:**

$$\forall e, ACW(e) \leq CW(e) \leq \alpha \cdot ACW(e)$$

## Approximate Cut Weights via Approximate Farthest Neighbor

$$CW(e) = \max_{x \in L(e), y \in R(e)} d(x,y) = \max_{x \in L(e)} d(x, Farthest_{y \in R(e)}(x))$$

$$ACW(e) = \max_{x \in L(e)} d(x, ApproxFarthest_{y \in R(e)}(x))$$

$\rightarrow$ take smallest of $L(e), R(e)$: $\mathcal{O}(n \log n)$ queries in total.

**Theorem**: Dynamic data structure for $\alpha$-AFN queries in time $\mathcal{O}(n^{1/\alpha^2})$.
$\rightarrow$ $\alpha$-AFN: point $p \in S$ s.t. $d(x,p) \geq d(x,y)/\alpha, \forall y \in S$.

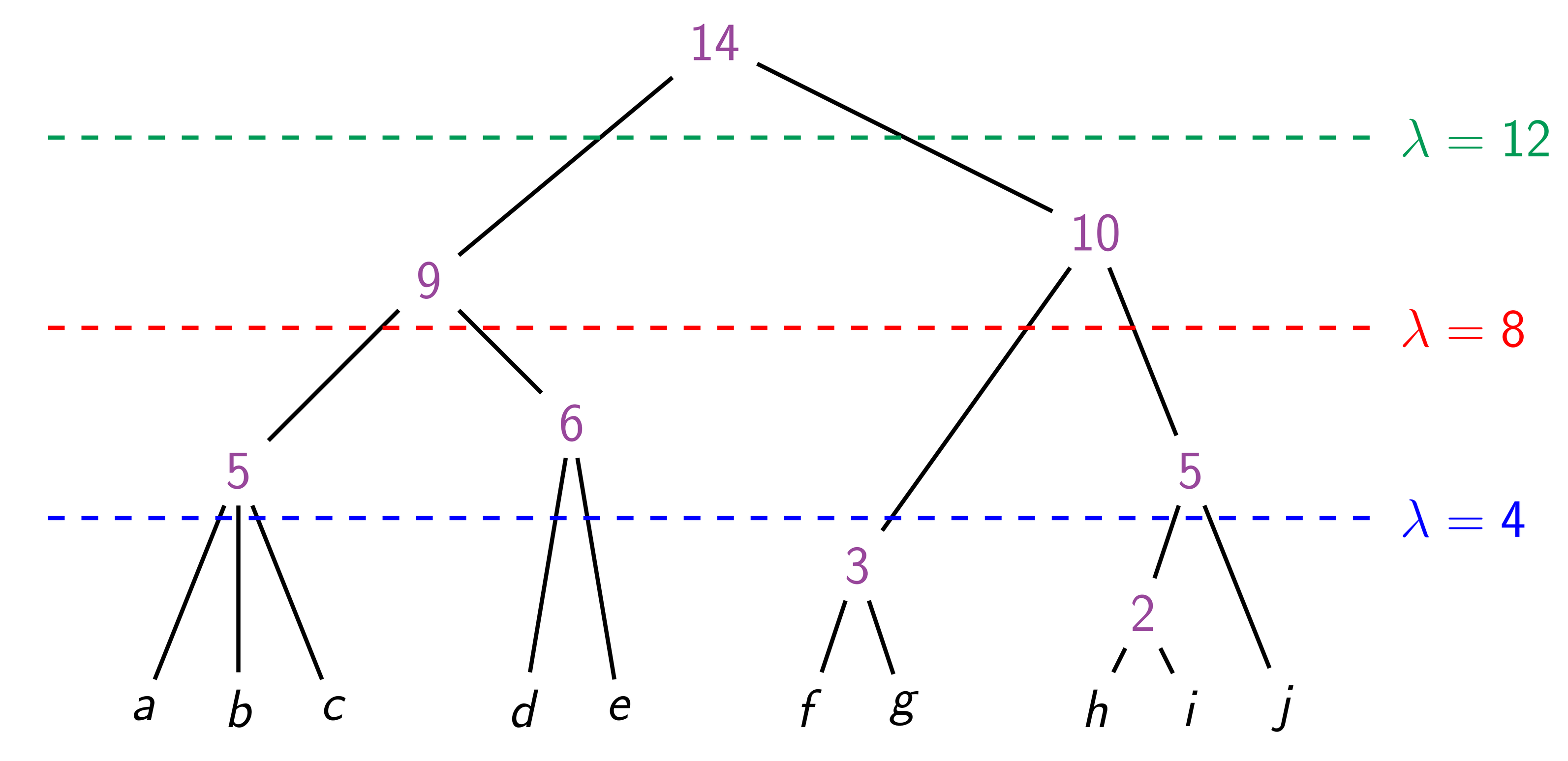
**Technique:**

- Project all points on $\mathcal{O}(n^{1/\alpha^2})$ random lines,
- Keep $\mathcal{O}(n^{1/\alpha^2})$ farthest points on each line,
- Return farthest point among those.

## Ultrametrics as Trees

$(X, \Delta)$ is an ultrametric space $\iff \exists$ tree $T = \underbrace{N}_{\text{nodes}} \cup \underbrace{X}_{\text{leaves}}$ with weights $w$

s.t. $\forall u \in T, w(u) \leq w(parent(u))$
$\forall x \in X, w(x) = 0$




Mapping: $\Delta(x,y) = w(LCA(x,y))$

## Algorithm for Ultrametric Embedding

$\mathcal{O}(n^2)$-time algorithm of Farach, Kannan and Warnow [?]:

1. Compute **minimum spanning tree** $T$ of $(X, d)$
2. Compute the **cut weights** $CW$ of $T$
3. Compute a cartesian tree $CT$ of $(T, CW) \rightarrow \Delta$

Approximation algorithm:

1. Compute a $\gamma$-**Kruskal Tree** $T$ of $(X, d)$ in $\mathcal{O}(n^{1+1/\gamma^2})$
2. Compute the $\alpha$-**approximate cut weights** $ACW$ of $T$ in $\mathcal{O}(n^{1+1/\alpha^2})$
3. Compute a cartesian tree $CT$ of $(T, ACW) \rightarrow \Delta$, $\alpha \cdot \gamma$ approx

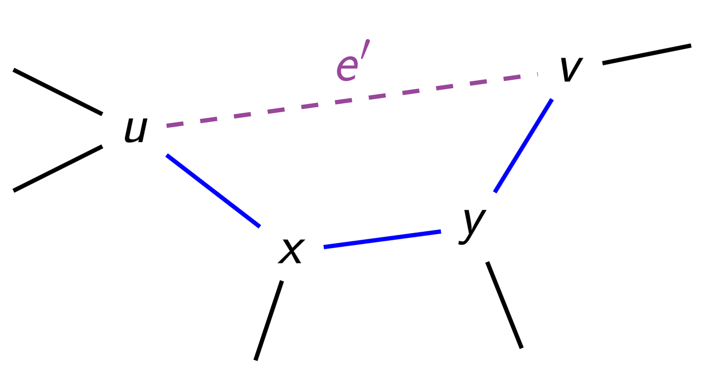
$\rightarrow$ for $\alpha = \gamma = \sqrt{c}$, $c$-approx. in $\mathcal{O}(n^{1+1/c})$

## $\gamma$-Kruskal Trees: Locally Approximate Minimum Spanning Trees

$\forall e$ on $u$-$v$ path,
MST: $w(e') \geq w(e)$

$\gamma$-KT: $w(e') \geq \frac{1}{\gamma} w(e)$

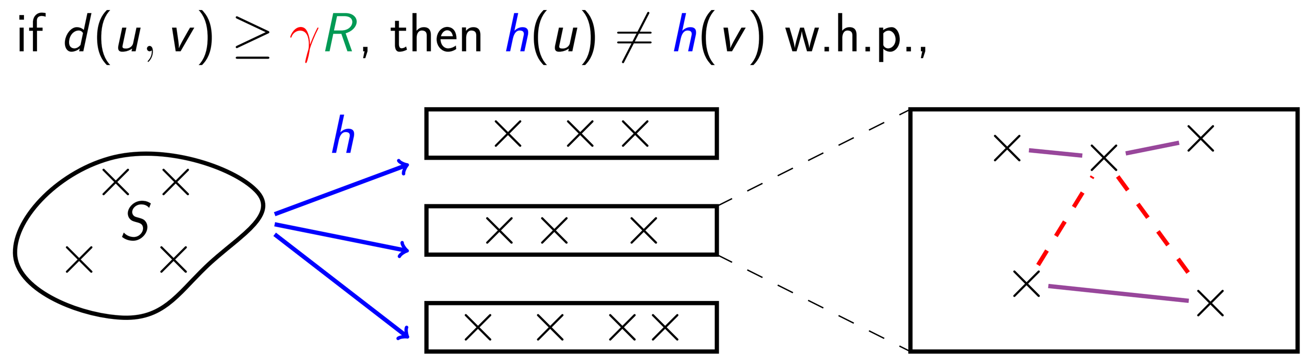
$\rightarrow$ Cannot take MST + long edge.




## $\gamma$-Kruskal Trees via Locality Sensitive Hashing

**Locality Sensitive Hashing**: hash function $h$

1. if $d(u,v) \leq R$, then $h(u) = h(v)$ w.h.p.,
2. if $d(u,v) \geq \gamma R$, then $h(u) \neq h(v)$ w.h.p.,




**Prop:** Takes $\mathcal{O}(n^{1+1/\gamma^2})$ time.
**Prop:** if $w(uv) \leq R$, then path with edges $w(e') \leq \gamma R$ between $u$ and $v$ in $E$.
**Algorithm for $\gamma$-KT:**

- Repeat for $R = d_{\min}, 2d_{\min}, 4d_{\min} \ldots, d_{\max}$
- take MST of union of all $E$'s.

## References