Ultrametrics Meet Fine-Grained Complexity

Karthik C. S.  
(New York University)

Joint work with

Vincent Cohen-Addad  
(Google)

Guillaume Lagarde  
(LaBRI)
How to Visualize all this data?
How to Visualize all this data?
Approaches

- Principal Component Analysis
Approaches

- Principal Component Analysis
- Dendrogram / Hierarchical Clustering
(Γ, Δ) is a metric space

- Δ(a, b) ≤ Δ(a, c) + Δ(b, c)
(\Gamma, \Delta) \text{ is a metric space}

- \ \Delta(a, b) \leq \Delta(a, c) + \Delta(b, c)

\textbf{Ultrametric:} \ \forall a, b, c \in \Gamma,

\[ \Delta(a, b) \leq \max\{\Delta(a, c), \Delta(b, c)\} \]
(\(\Gamma, \Delta\)) is a metric space

- \(\Delta(a,b) \leq \Delta(a,c) + \Delta(b,c)\)

**Ultrametric:** \(\forall a, b, c \in \Gamma,\)

\[\Delta(a, b) \leq \max\{\Delta(a, c), \Delta(b, c)\}\]

**Cool Property:** \(\forall a, b, c \in \Gamma,\)

\[\Delta(a, b) = \Delta(a, c) \text{ or } \Delta(a, c) = \Delta(b, c) \text{ or } \Delta(a, b) = \Delta(b, c)\]
Example

\[ \Delta: \text{Leaves} \rightarrow \mathbb{R}^+ \]

\[ \Delta(x, y) = w(LCA(x, y)) \]

\( w \) is non-increasing from root
Example

Arises in:

- Evolutionary Biology
- Hierarchical Clustering

\[ \Delta : \text{Leaves} \rightarrow \mathbb{R}^+ \]
\[ \Delta(x, y) = \omega(LCA(x, y)) \]
\( \omega \) is non-increasing from root
- **Topology**: Discrete metric
- **Number Theory**: $p$-adic numbers
- **Graph Theory**: Minmax paths
Example

- **Topology**: Discrete metric
- **Number Theory**: $p$-adic numbers
- **Graph Theory**: Minmax paths

$$\Delta : V \to \mathbb{R}^+$$

$$\Delta(x, y) = \min_{\text{paths } p \text{ from } x \to y} \max_{e \in p} w(e)$$

$$\Delta(c, e) = 1.5$$
Focus on Embedding
Focus on Embedding

Embedding from Ultrametric
Focus on **Embedding**

Embedding *from* Ultrametric

- Not today
Focus on Embedding

Embedding from Ultrametric
  • Not today

Embedding to Ultrametric
Focus on **Embedding**

Embedding **from** Ultrametric
- Not today

Embedding **to** Ultrametric

\[ \tau : X \to L, \ \forall x, y \in X, \]
\[ \|x - y\|_p \leq \Delta(\tau(x), \tau(y)) = w(LCA(\tau(x), \tau(y))) \leq \rho_{\text{OPT}} \cdot \|x - y\|_p \]
Motivation: Data Visualization

\[ \{a, b, c, d, e\} \subseteq \mathbb{R}^{100} \]
Motivation: Data Visualization

\[
\{a, b, c, d, e\} \subseteq \mathbb{R}^{100}
\]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>1.5</td>
<td>1</td>
<td>0</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>1</td>
<td>1.5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>e</td>
<td>1</td>
<td>2</td>
<td>1.5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Motivation: Data Visualization

\( \{a, b, c, d, e\} \subseteq \mathbb{R}^{100} \)
Replace edge weights with min-max path weights
Results

Theorem (Farach–Kannan–Warnow’95)
Given the distance matrix of $n$ points, the optimal ultrametric embedding can be computed in time $O(n^2)$. 

Theorem (Cohen-Addad–K–Lagarde)
⊚ Assuming SETH, no approximate embedding in $O(1.99^\ell)$ time from $\ell_\infty$-metric.
⊚ Assuming non-standard hypothesis, no approximate embedding in $O(1+\epsilon^\ell)$ time from Euclidean metric.
⊚ For any $\ell \geq 1$, $5$-approximate embedding in time $O(1+\epsilon^\ell)$ for Euclidean metric.

Performs Well in Experiments!
Results

Theorem (Farach–Kannan–Warnow’95)
Given the distance matrix of \( n \) points, the optimal ultrametric embedding can be computed in time \( O(n^2) \).

Theorem (Cohen-Addad–K–Lagarde)

\( \circledast \) Assuming SETH, no 1.5 approximate embedding in \( n^{1.99} \) time from \( l_{\infty} \)-metric.
### Results

<table>
<thead>
<tr>
<th>Theorem (Farach–Kannan–Warnow’95)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given the distance matrix of $n$ points, the optimal ultrametric embedding can be computed in time $O(n^2)$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Cohen-Addad–K–Lagarde)</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊢ Assuming SETH, no $1.5$ approximate embedding in $n^{1.99}$ time from $l_{\infty}$-metric.</td>
</tr>
<tr>
<td>⊢ Assuming non-standard hypothesis, no $1.001$ approximate in $n^{1+o(1)}$ time from Euclidean metric.</td>
</tr>
</tbody>
</table>
Results

Theorem (Farach–Kannan–Warnow’95)
Given the distance matrix of $n$ points, the optimal ultrametric embedding can be computed in time $O(n^2)$.

Theorem (Cohen-Addad–K–Lagarde)

- Assuming SETH, no 1.5 approximate embedding in $n^{1.99}$ time from $\ell_\infty$-metric.
- Assuming non-standard hypothesis, no 1.001 approximate in $n^{1+o(1)}$ time from Euclidean metric.
- For any $\gamma \geq 1$, $5\gamma$ approximate embedding in time $O(n^{1+\frac{1}{\gamma^2}})$ for Euclidean metric.
Results

Theorem (Farach–Kannan–Warnow’95)

Given the distance matrix of $n$ points, the optimal ultrametric embedding can be computed in time $O(n^2)$.

Theorem (Cohen-Addad–K–Lagarde)

- Assuming SETH, no $1.5$ approximate embedding in $n^{1.99}$ time from $\ell_\infty$-metric.
- Assuming non-standard hypothesis, no $1.001$ approximate in $n^{1+o(1)}$ time from Euclidean metric.
- For any $\gamma \geq 1$, $5\gamma$ approximate embedding in time $O(n^{1+\frac{1}{\gamma^2}})$ for Euclidean metric. Performs Well in Experiments!
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

   \[
P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\max}(i, j) = w(e)\}
   \]

   \[
   CW(e) = \max_{(i,j) \in P(e)} \|v_i - v_j\|_p
   \]
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

   $$P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\text{max}}(i, j) = w(e)\}$$

   $$CW(e) = \max_{(i,j) \in P(e)} ||v_i - v_j||_p$$

3. Build **ultrametric tree**:
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^\text{ULT}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

   $$ P(e) = \{ (i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\text{max}}(i, j) = w(e) \} $$
   $$ CW(e) = \max_{(i, j) \in P(e)} \|v_i - v_j\|_p $$

3. Build **ultrametric tree**:
   - Leaves are $V$
   - **Root** is $e \in E(T^G)$ of max weight
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

   \[
P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\max}(i, j) = w(e)\}
   \]

   \[
   CW(e) = \max_{(i, j) \in P(e)} \|v_i - v_j\|_p
   \]

3. Build **ultrametric tree**:
   - **Leaves** are $V$
   - **Root** is $e \in E(T^G)$ of max weight
   - **Recursively** build both children components of root
Farach–Kannan–Warnow’95: Algorithm

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

$$P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\text{max}}(i, j) = w(e)\}$$

$$\text{CW}(e) = \max_{(i,j) \in P(e)} \|v_i - v_j\|_p$$

3. Build **ultrametric tree**:

   - **Leaves** are $V$
   - **Root** is $e \in E(T^G)$ of max weight
   - **Recursively** build both children components of root
   - **Weight** of internal node $e$ is $\text{CW}(e)$
Cut weights: Illustration
Cut weights: Illustration
Cut weights: Illustration

$L(a,b)$

$R(a,b)$
Cut weights: Illustration

\[ L(a,b) \]

\[ R(a,b) \]
Cut weights: Illustration
Cut weights: Illustration

![Diagram showing a graph with nodes a and b, and the sets L(a,b) and R(a,b) highlighted.](image)
Cut weights: Illustration

\[ \text{L(a,b)} \]

\[ \text{R(a,b)} \]
Farach–Kannan–Warnow’95: Algorithm

Input: An edge-weighted clique $G$

Output: An ultrametric tree $T^{\text{ULT}}$

1. Compute Minimum Spanning Tree $T^G$

2. Compute cut weights of $T^G$, i.e., $\forall e \in E(T^G)$:

$$P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\text{max}}(i, j) = w(e)\}$$

$$CW(e) = \max_{(i, j) \in P(e)} \|v_i - v_j\|_p$$

3. Build ultrametric tree:
**Farach–Kannan–Warnow’95: Algorithm**

**Input:** An edge-weighted clique $G$

**Output:** An ultrametric tree $T^{\text{ULT}}$

1. Compute **Minimum Spanning Tree** $T^G$

2. Compute **cut weights** of $T^G$, i.e., $\forall e \in E(T^G)$:

   $P(e) = \{(i, j) \in V \times V \mid e \in \text{Path}_{T^G}(i, j), \Delta_{\text{max}}(i, j) = w(e)\}$

   $$CW(e) = \max_{(i,j) \in P(e)} \|v_i - v_j\|_p$$

3. Build **ultrametric tree**:
   - **Leaves** are $V$
   - **Root** is $e \in E(T^G)$ of max weight
   - **Recursively** build both children components of root
   - **Weight** of internal node $e$ is $CW(e)$
Approximation Variants

- $\gamma$-approximate MST $T$
○ \( \gamma \)-approximate MST \( T \)

\[ \forall e \in G \setminus T, \ w(e) \geq \frac{1}{\gamma} \cdot \max_{e' \in C^T_e} w(e'), \]

\( C^T_e \) is cycle induced by adding \( e \) to \( T \).
Approximation Variants

- $\gamma$-approximate MST $T$

$$\forall e \in G \setminus T, \ w(e) \geq \frac{1}{\gamma} \cdot \max_{e' \in C^T_e} w(e'),$$

$C^T_e$ is cycle induced by adding $e$ to $T$.

- $\beta$-approximate Cut Weights ACW:
Approximation Variants

- **γ-approximate MST** $T$

  \[ \forall e \in G \setminus T, \ w(e) \geq \frac{1}{\gamma} \cdot \max_{e' \in C^T_e} w(e'), \]

  $C^T_e$ is **cycle** induced by adding $e$ to $T$.

- **β-approximate Cut Weights** $ACW$:

  \[ \forall e \in T, \ CW(e) \leq ACW(e) \leq \beta \cdot CW(e). \]
Our Approximation Algorithm

APPROX-BUF: an approximation algorithm for BUF∞

1. Compute a γ-approximate MST TG over the complete graph G
2. Compute a β-estimate of the cut weights of the edges in TG
3. Compute the ultrametric tree using TG and β-estimates

→ This gives a γ · β-approximation
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$

→ This gives a $\gamma \cdot \beta$-approximation
1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T_G$ over the complete graph $G$

$w(\_\_\_) \geq 1/\gamma \cdot \max(\_\_\_)$

1. Compute a $\gamma$-approximate MST $T_G$ over the complete graph $G$
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates

$\rightarrow$ This gives a $\gamma \cdot \beta$-approximation
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates

$w(\ldots) \geq 1/\gamma \cdot \max(\ldots)$
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates

This gives a $\gamma \cdot \beta$-approximation.
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates
Our Approximation Algorithm

1. Compute a \( \gamma \)-approximate MST \( T^G \) over the complete graph \( G \)
2. Compute a \( \beta \)-estimate of the cut weights of the edges in \( T^G \)
3. Compute the ultrametric tree using \( T^G \) and \( \beta \)-estimates
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G$
3. Compute the ultrametric tree using $T^G$ and $\beta$-estimates

→ This gives a $\gamma \cdot \beta$-approximation
Proof Ideas

○ $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
Proof Ideas

- $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
  - If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  - If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
\(\gamma\)-approximate MST \(T^G\) is an exact MST for \(G'\):

- If \((v_i, v_j) \in T^G\) then \(w'(v_i, v_j) = w(v_i, v_j)\)
- If \((v_i, v_j) \notin T^G\) then \(w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)\)
- Distances in \(G\) distorted in \(G'\) by at most \(\gamma\) factor
Proof Ideas

○ $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
  ○ If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  ○ If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
  ○ Distances in $G$ distorted in $G'$ by at most $\gamma$ factor

○ Next if we use exact CW (i.e., $\beta = 1$) of $T^G$: 
Proof Ideas

- $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
  - If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  - If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
  - Distances in $G$ distorted in $G'$ by at most $\gamma$ factor

- Next if we use exact CW (i.e., $\beta = 1$) of $T^G$:
  - We obtain optimal ultrametric embedding $\Delta'$ of $G'$
Proof Ideas

- $\gamma$-approximate MST $T^G$ is an **exact** MST for $G'$:
  - If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  - If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
  - Distances in $G$ distorted in $G'$ by at most $\gamma$ factor

- Next if we use **exact** CW (i.e., $\beta = 1$) of $T^G$:
  - We obtain **optimal** ultrametric embedding $\Delta'$ of $G'$
  - Distortion of embedding $G'$ is **less** than Distortion of embedding $G$
Proof Ideas

- $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
  - If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  - If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
  - Distances in $G$ distorted in $G'$ by at most $\gamma$ factor

- Next if we use exact CW (i.e., $\beta = 1$) of $T^G$:
  - We obtain optimal ultrametric embedding $\Delta'$ of $G'$
  - Distortion of embedding $G'$ is less than Distortion of embedding $G$

- Using $\beta$-approximate CW of $T^G$ we obtain ultrametric embedding $\Delta$ of $G'$
Proof Ideas

- $\gamma$-approximate MST $T^G$ is an exact MST for $G'$:
  - If $(v_i, v_j) \in T^G$ then $w'(v_i, v_j) = w(v_i, v_j)$
  - If $(v_i, v_j) \notin T^G$ then $w'(v_i, v_j) = \gamma \cdot w(v_i, v_j)$
  - Distances in $G$ distorted in $G'$ by at most $\gamma$ factor

- Next if we use exact CW (i.e., $\beta = 1$) of $T^G$:
  - We obtain optimal ultrametric embedding $\Delta'$ of $G'$
  - Distortion of embedding $G'$ is less than Distortion of embedding $G$

- Using $\beta$-approximate CW of $T^G$ we obtain ultrametric embedding $\Delta$ of $G'$

$$\forall v_i, v_j \in V, \Delta'(v_i, v_j) \leq \Delta(v_i, v_j) \leq \beta \cdot \Delta'(v_i, v_j)$$
For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos ‘13 in time $O(nd + n^{1+O(1/\gamma^2)})$
For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos’13 in time $O(nd + n^{1+O(1/\gamma^2)})$

$\beta = 5$-estimate using a variant of union-find data structure
A $\gamma$-spanner of $S \in \mathbb{R}^d$ is $G(S, E)$:

$$\forall u, v \in S, \|u - v\|_2 \leq \Delta^G(u, v) \leq \gamma \cdot \|u - v\|_2$$
Implementation: Approximate Euclidean MST

- A $\gamma$-spanner of $S \in \mathbb{R}^d$ is $G(S, E)$:

$$\forall u, v \in S, \|u - v\|_2 \leq \Delta^G(u, v) \leq \gamma \cdot \|u - v\|_2$$

- For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos' 13 in time $O(nd + n^{1+O(1/\gamma^2)})$
Implementation: Approximate Euclidean MST

○ A $\gamma$-spanner of $S \in \mathbb{R}^d$ is $G(S, E)$:

$$\forall u, v \in S, \|u - v\|_2 \leq \Delta^G(u, v) \leq \gamma \cdot \|u - v\|_2$$

○ For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos’13 in time $O(nd + n^{1+O(1/\gamma^2)})$
  ○ Uses LSH of Andoni-Indyk’06
A $\gamma$-spanner of $S \in \mathbb{R}^d$ is $G(S, E)$:

$$\forall u, v \in S, \|u - v\|_2 \leq \Delta^G(u, v) \leq \gamma \cdot \|u - v\|_2$$

For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos'13 in time $O(nd + n^{1+O(1/\gamma^2)})$

- Uses LSH of Andoni-Indyk'06

Kruskal Algorithm on $G$ gives $\gamma$-approximate MST $T$ of $G$

$$\forall e \in G \setminus T, w(e) \geq \frac{1}{\gamma} \cdot \max_{e' \in C_e^T} w(e').$$
Compute 5-estimate of CW of edges of $T$ in $O(nd + n \log n)$ time
Implementation: Approximate Cut Weights

- Compute $5$-estimate of CW of edges of $T$ in $O(nd + n \log n)$ time
- Maintain union-find data structure over vertices of $T$ such that for each equivalence class $C$
  - store $r_C$: a special vertex
  - store $m_C$: max distance from $r_C$ inside $C$
Implementation: Approximate Cut Weights

- Compute a 5-estimate of CW of edges of $T$ in $O(nd + n \log n)$ time.
- Maintain a union-find data structure over vertices of $T$ such that for each equivalence class $C$
  - store $r_C$: a special vertex
  - store $m_C$: max distance from $r_C$ inside $C$
- Consider edges of $T$ in increasing order.
Implementation: Approximate Cut Weights

- Compute 5-estimate of CW of edges of $T$ in $O(nd + n \log n)$ time
- Maintain union-find data structure over vertices of $T$ such that for each equivalence class $C$
  - store $r_C$: a special vertex
  - store $m_C$: max distance from $r_C$ inside $C$
- Consider edges of $T$ in increasing order
- For $e = (x, y) \in C \times D$
  
  $$5 \cdot \max(d(r_C, r_D), m_C - d(r_C, r_D), m_D - d(r_C, r_D))$$

  is a 5 approximation of $CW(e)$. 
Experiments

- Data set:
  - DIABETES (768 samples, 8 features)
  - MICE (1080 samples, 77 features)
  - PENDIGITS (10992 samples, 16 features)
Experiments

- Data set:
  - DIABETES (768 samples, 8 features)
  - MICE (1080 samples, 77 features)
  - PENDIGITS (10992 samples, 16 features)

- Compare with Scikit-learn implementation:
  - Single linkage
  - Complete linkage
  - Average linkage
  - Ward method
Experiments

- **Data set:**
  - DIABETES (768 samples, 8 features)
  - MICE (1080 samples, 77 features)
  - PENDIGITS (10992 samples, 16 features)

- **Compare with Scikit-learn implementation:**
  - Single linkage
  - Complete linkage
  - Average linkage
  - Ward method

- **Our algorithms:**
  - ApproxULT (approximate MST)
  - ApproxAccULT (exact MST)
### Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>DIABETES</th>
<th>MICE</th>
<th>PENDIGITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>11.1</td>
<td>9.7</td>
<td>27.5</td>
</tr>
<tr>
<td>Complete</td>
<td>18.5</td>
<td>11.8</td>
<td>33.8</td>
</tr>
<tr>
<td>Single</td>
<td>6.0</td>
<td>4.9</td>
<td>14.0</td>
</tr>
<tr>
<td>Ward</td>
<td>61.0</td>
<td>59.3</td>
<td>433.8</td>
</tr>
<tr>
<td>ApproxULT</td>
<td>41.0</td>
<td>51.2</td>
<td>109.8</td>
</tr>
<tr>
<td>ApproxAccULT</td>
<td>9.6</td>
<td>9.4</td>
<td>37.2</td>
</tr>
<tr>
<td>Farach et al.</td>
<td>6.0</td>
<td>4.9</td>
<td>13.9</td>
</tr>
</tbody>
</table>
Experiments

Single linkage is almost optimal

<table>
<thead>
<tr>
<th></th>
<th>DIABETES</th>
<th>MICE</th>
<th>PENDIGITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>11.1</td>
<td>9.7</td>
<td>27.5</td>
</tr>
<tr>
<td>Complete</td>
<td>18.5</td>
<td>11.8</td>
<td>33.8</td>
</tr>
<tr>
<td>Single</td>
<td>6.0</td>
<td>4.9</td>
<td>14.0</td>
</tr>
<tr>
<td>Ward</td>
<td>61.0</td>
<td>59.3</td>
<td>433.8</td>
</tr>
<tr>
<td>ApproxULT</td>
<td>41.0</td>
<td>51.2</td>
<td>109.8</td>
</tr>
<tr>
<td>ApproxAccULT</td>
<td>9.6</td>
<td>9.4</td>
<td>37.2</td>
</tr>
<tr>
<td>Farach et al.</td>
<td>6.0</td>
<td>4.9</td>
<td>13.9</td>
</tr>
</tbody>
</table>
Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>DIABETES</th>
<th>MICE</th>
<th>PENDIGITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>11.1</td>
<td>9.7</td>
<td>27.5</td>
</tr>
<tr>
<td>Complete</td>
<td>18.5</td>
<td>11.8</td>
<td>33.8</td>
</tr>
<tr>
<td>Single</td>
<td>6.0</td>
<td>4.9</td>
<td>14.0</td>
</tr>
<tr>
<td>Ward</td>
<td>61.0</td>
<td>59.3</td>
<td>433.8</td>
</tr>
<tr>
<td>ApproxULT</td>
<td>41.0</td>
<td>51.2</td>
<td>109.8</td>
</tr>
<tr>
<td>ApproxAccULT</td>
<td>9.6</td>
<td>9.4</td>
<td>37.2</td>
</tr>
<tr>
<td>Farach et al.</td>
<td>6.0</td>
<td>4.9</td>
<td>13.9</td>
</tr>
</tbody>
</table>

- Single linkage is almost optimal
- Room for improvement: Good approximate MST
Experiments

- Order of magnitude faster
- [GitHub link]
- Helped also by Rémi de Verclos
Experiments

Order of magnitude faster

Helped also by Rémi de Verclos

average  complete  single  ward  approxU LT

DIABETES  MICE  PENDIGITS
Experiments

Order of magnitude faster

https://github.com/guillaume-lagarde/fast-ultrametrics

Helped also by Rémi de Verclos
Assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time, given $X \in \mathbb{R}^{O_{\varepsilon}(\log n)}$ ($|X| = n$) in $\ell_\infty$-space can distinguish:

**YES**: $X$ can be embedded *isometrically* into an ultrametric.

**NO**: Distortion is at least $3/2$. 

Morally Equivalent to Search Version
Assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time, given $X \in \mathbb{R}^{O(\log n)} (|X| = n)$ in $\ell_\infty$-space can distinguish:

YES: $X$ can be embedded \textit{isometrically} into an ultrametric.
NO: Distortion is at least $3/2$.

Moreover, in both cases $\text{dist}(X) = \text{dist}(Y) = 2$ and $\text{dist}(X,Y) \in \{1,3\}$.
Hardness from SETH

Theorem (Cohen-Addad–K–Lagarde)
Assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time, given $X \in \mathbb{R}^{O_\varepsilon(\log n)}$ ($|X| = n$) in $\ell_\infty$-space can distinguish:

YES: $X$ can be embedded isometrically into an ultrametric.
NO: Distortion is at least $3/2$.

Theorem (David–K–Laekhanukit’19)
Assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time, given $A, B \in \mathbb{R}^{O_\varepsilon(\log n)}$ ($|A| = |B| = n$) can distinguish:

YES: $\exists (a, b) \in A \times B$ such that $\|a - b\|_\infty = 1$.
NO: $\forall (a, b) \in A \times B$ we have $\|a - b\|_\infty = 3$.

Moreover, in both cases $\text{dist}(A) = \text{dist}(B) = 2$ and $\text{dist}(A, B) \in \{1, 3\}$. 
Hardness from SETH: YES case

- **Input:** \( A, B \in \mathbb{R}^{O(\log n)} (|A| = |B| = n) \)

- **Promise:** \( \forall a, a' \in A \) and \( \forall b, b' \in B: \|a - a'\|_\infty = \|b - b'\|_\infty = 2 \)

- **Case Assumption:** \( \forall (a, b) \in A \times B \) we have \( \|a - b\|_\infty = 3 \)
Hardness from SETH: YES case

- **Input:** $A, B \in \mathbb{R}^{O(\log n)} (|A| = |B| = n)$

- **Promise:** $\forall a, a' \in A$ and $\forall b, b' \in B$: $\|a - a'\|_\infty = \|b - b'\|_\infty = 2$

- **Case Assumption:** $\forall (a, b) \in A \times B$ we have $\|a - b\|_\infty = 3$
Hardness from SETH: NO case

- **Input**: \( A, B \in \mathbb{R}^{O(\log n)} \) (\(|A| = |B| = n\))
- **Promise**: \( \forall a, a' \in A \) and \( \forall b, b' \in B \): \( \|a - a'\|_\infty = \|b - b'\|_\infty = 2 \)
- **Case Assumption**: \( \exists (a, b) \in A \times B \) such that \( \|a - b\|_\infty = 1 \)
Hardness from SETH: NO case

- **Input:** \( A, B \in \mathbb{R}^{O(\log n)} \) (\(|A| = |B| = n\))

- **Promise:** \( \forall a, a' \in A \) and \( \forall b, b' \in B : \|a - a'\|_\infty = \|b - b'\|_\infty = 2 \)

- **Case Assumption:** \( \exists (a, b) \in A \times B \) such that \( \|a - b\|_\infty = 1 \)

- Let \( S : \{a, a', b\} \) such that \( \|a - b\| = 1 \) and \( \|a' - b\| = 3 \)
Hardness from SETH: NO case

- **Input:** $A, B \in \mathbb{R}^{O(\log n)}$ ($|A| = |B| = n$)

- **Promise:** $\forall a, a' \in A$ and $\forall b, b' \in B$: $||a - a'||_{\infty} = ||b - b'||_{\infty} = 2$

- **Case Assumption:** $\exists (a, b) \in A \times B$ such that $||a - b||_{\infty} = 1$

- Let $S : \{a, a', b\}$ such that $||a - b|| = 1$ and $||a' - b|| = 3$

- Let $\tau : S \to L$ be ultrametric embedding and $\rho$ be distortion
Hardness from SETH: NO case

- **Input:** \( A, B \in \mathbb{R}^{O(\log n)} \) (\(|A| = |B| = n\))

- **Promise:** \( \forall a, a' \in A \text{ and } \forall b, b' \in B: \|a - a'\|_\infty = \|b - b'\|_\infty = 2 \)

- **Case Assumption:** \( \exists (a, b) \in A \times B \text{ such that } \|a - b\|_\infty = 1 \)

- Let \( S : \{a, a', b\} \) such that \( \|a - b\| = 1 \) and \( \|a' - b\| = 3 \)

- Let \( \tau : S \to L \) be ultrametric embedding and \( \rho \) be distortion

\[
3 = \|a' - b\|_\infty \leq \Delta(\tau(a'), \tau(b)) \\
\leq \max\{\Delta(\tau(a), \tau(b)), \Delta(\tau(a'), \tau(a))\} \\
\leq \max\{\rho \cdot \|a - b\|_\infty, \rho \cdot \|a' - a\|_\infty\} = 2\rho
\]
Colinearity Problem

- **YES** case: Input is $n$ points sampled from $B_d$. 

  - Sample $(0, 1, \ldots, 0)$ from $B_d$.
  - Pick distinct indices $8, 9, \ldots$ in $[n]$ at random.
  - Let $0_8, 9$ be the midpoint of $0_8$ and $0_9$.
  - Let $\tilde{0}_8$ be $(1 - \cdot)0_8 + \cdot 0_8, 9$.
  - Input is $(0, 1, \ldots, \tilde{0}_8, 0_8 + 1, \ldots, 0)$. 

Colinearity Problem

- **YES** case: Input is $n$ points sampled from $B_d$.
- **NO** case:
  - Sample $(a_1, \ldots, a_n)$ from $B_d$. 
  - Pick distinct indices $8, 9, \vdots$ in $[1]$ at random.
  - Let $0_8, 9$ be the midpoint of $0_8$ and $0_9$.
  - Let $\tilde{0}_8$ be $(1 - \cdot)0_8 + \cdot 0_8, 9$. 
  - Input is $(0_1, \ldots, \tilde{0}_8, \tilde{0}_8, 1, \ldots, 0_1)$. 


Colinearity Problem

- **YES** case: Input is $n$ points sampled from $\mathcal{B}_d$.

- **NO** case:
  - Sample $(a_1, \ldots, a_n)$ from $\mathcal{B}_d$.
  - Pick distinct indices $i, j, k$ in $[n]$ at random.
Colinearity Problem

- **YES** case: Input is \( n \) points sampled from \( B_d \).

- **NO** case:
  - Sample \((a_1, \ldots, a_n)\) from \( B_d \).
  - Pick distinct indices \( i, j, k \) in \([n]\) at random.
  - Let \( a_{i,j} \) be the midpoint of \( a_i \) and \( a_j \).
Colinearity Problem

○ **YES** case: Input is $n$ points sampled from $\mathcal{B}_d$.

○ **NO** case:
  - Sample $(a_1, \ldots, a_n)$ from $\mathcal{B}_d$.
  - Pick distinct indices $i, j, k$ in $[n]$ at random.
  - Let $a_{i,j}$ be the **midpoint** of $a_i$ and $a_j$.
  - Let $\tilde{a}_k$ be $(1 - \rho) \cdot a_k + \rho \cdot a_{i,j}$. 
Colinearity Problem

- **YES** case: Input is $n$ points sampled from $\mathcal{B}_d$.

- **NO** case:
  - Sample $(a_1, \ldots, a_n)$ from $\mathcal{B}_d$.
  - Pick **distinct** indices $i, j, k$ in $[n]$ at random.
  - Let $a_{i,j}$ be the **midpoint** of $a_i$ and $a_j$.
  - Let $\tilde{a}_k = (1 - \rho) \cdot a_k + \rho \cdot a_{i,j}$.
  - Input is $(a_1, \ldots, \tilde{a}_k, a_{k+1}, \ldots, a_n)$. 
Colinearity Hypothesis

Colinearity Hypothesis: There exists constants $\rho, \varepsilon > 0$ such that no randomized algorithm running in time $n^{1+\varepsilon}$ can distinguish the two cases for every $d \geq O_{\rho,\varepsilon}(\log n)$. 

- Worst Case variant is $\Theta$-hard for even $n = 2$.
- Related to Light bulb problem.

Theorem (Cohen-Addad–K–Lagarde): Assuming CH, there exists $\rho, \varepsilon > 0$, no randomized algorithm running in time $n^{1+\varepsilon}$ can distinguish the two cases for every $d \geq O_{\rho,\varepsilon}(\log n)$. 

- YES: Distortion is at most $1$.
- NO: Distortion is at least $1+2$. 

/s two/ \seven
Colinearity Hypothesis

- Colinearity Hypothesis: There exists constants $\rho, \varepsilon > 0$ such that no randomized algorithm running in time $n^{1+\varepsilon}$ can distinguish the two cases for every $d \geq O_{\rho,\varepsilon}(\log n)$.

- Worst Case variant is 3-SUM hard for even $d = 2$. 

Colinearity Hypothesis: There exists constants $\rho, \varepsilon > 0$ such that no randomized algorithm running in time $n^{1+\varepsilon}$ can distinguish the two cases for every $d \geq O_{\rho,\varepsilon}(\log n)$.

Worst Case variant is 3-SUM hard for even $d = 2$.

Related to Light bulb problem.
Colinearity Hypothesis

- **Colinearity Hypothesis**: There exists constants $\rho, \varepsilon > 0$ such that no randomized algorithm running in time $n^{1+\varepsilon}$ can distinguish the two cases for every $d \geq O_{\rho, \varepsilon} (\log n)$.

- **Worst Case** variant is 3-SUM hard for even $d = 2$.

- Related to Light bulb problem.

**Theorem (Cohen-Addad–K–Lagarde)**

Assuming CH, there exists $\varepsilon, \delta > 0$, no randomized algorithm running in $n^{1+\varepsilon}$ time, given $X \in \mathbb{R}^{O_{\varepsilon, \delta}(\log n)} (|X| = n)$ in Euclidean space can distinguish:

**YES**: Distortion is at most $1 + \delta$.

**NO**: Distortion is at least $1 + 2\delta$. 
Results

Theorem (Farach–Kannan–Warnow’95)
Given the distance matrix of \( n \) points, the optimal ultrametric embedding can be computed in time \( O(n^2) \).

Theorem (Cohen-Addad–K–Lagarde)
- Assuming SETH, no \( 1.5 \) approximate embedding in \( n^{1.99} \) time from \( l_\infty \)-metric.
- Assuming Colinearity Hypothesis, no \( 1.001 \) approximate in \( n^{1+o(1)} \) time from Euclidean metric.
- For any \( \gamma \geq 1 \), \( 5\gamma \) approximate embedding in time \( O(n^{1+\frac{1}{\gamma^2}}) \) for Euclidean metric.
Results

Theorem (Farach–Kannan–Warnow’95)
Given the distance matrix of $n$ points, the optimal ultrametric embedding can be computed in time $O(n^2)$.

Theorem (Cohen-Addad–K–Lagarde)

- Assuming SETH, no $1.5$ approximate embedding in $n^{1.99}$ time from $l_\infty$-metric.
- Assuming Colinearity Hypothesis, no $1.001$ approximate in $n^{1+o(1)}$ time from Euclidean metric.
- For any $\gamma \geq 1$, $5\gamma$ approximate embedding in time $O(n^{1+\frac{1}{\gamma^2}})$ for Euclidean metric.

Performs Well in Experiments!
Open Problems

Improved Approximation Factor?
Open Problems

Improved Approximation Factor?

Euclidean Inapproximability under SETH?
THANK YOU!