Ultrametrics Meet Fine-Grained Complexity

Karthik C. S.
(Tel Aviv University)

Joint work with

Vincent Cohen-Addad
(Google)

Guillaume Lagarde
(LaBRI)
\( (\Gamma, \Delta) \) is a metric space

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

○ $(\Gamma, \Delta)$ is a metric space
  ○ $\Delta(a, b) \leq \Delta(a, c) + \Delta(b, c)$

○ **Ultrametric:** $\forall a, b, c \in \Gamma,$

$$\Delta(a, b) \leq \max\{\Delta(a, c), \Delta(b, c)\}$$
(Γ, Δ) is a metric space

- Δ(a, b) ≤ Δ(a, c) + Δ(b, c)

Ultrametric: ∀a, b, c ∈ Γ,

\[ Δ(a, b) ≤ \max\{Δ(a, c), Δ(b, c)\} \]

Cool Property: ∀a, b, c ∈ Γ,

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

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

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

w is non-increasing from root
Example

Arises in:
- Evolutionary Biology
- Hierarchical Clustering

\[ \Delta : \text{Leaves} \to \mathbb{R}^+ \]
\[ \Delta(x, y) = w(\text{LCA}(x, y)) \]

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

- **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

\[
\Delta(x,y) = \min_{\text{paths } p \text{ e } P} \max_{x \leftrightarrow y} w(e)
\]

\[
\Delta(c,e) = 1.5
\]
Focus on Embedding
Focus on **Embedding**

**Embedding from Ultrametric**
Directions

- Focus on **Embedding**
- Embedding **from** Ultrametric
  - Not today
Focus on Embedding

Embedding from Ultrametric
  - Not today

Embedding to Ultrametric
Directions

Focus on **Embedding**

Embedding **from** Ultrametric

- Not today

Embedding **to** Ultrametric

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

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

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

\{a, b, c, d, e\} \subseteq \mathbb{R}^{100}
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 one approximate embedding in $2^{0.99}\ell_\infty$ time from $\ell_\infty$-metric.
⊚ Assuming non-standard hypothesis, no one/zero/one approximate in $1+\epsilon > (1+\epsilon)$ time from Euclidean metric.
⊚ For any $\epsilon \geq 1$, 5/2 approximate embedding in time $O(n^2)$ 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)

- Assuming SETH, no $1.5$ approximate embedding in $n^{1.99}$ time from $\ell_\infty$-metric.
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.
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 \( l_\infty \)-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 $T^{\text{ULT}}$
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., $\forall e \in E(T^G)$:

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

   $$C(e) = \max_{(i, j) \in P(e)} \|v_i - v_j\|_p$$
**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., $\forall e \in E(T^G)$:

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

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

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., $\forall e \in E(T^G)$:

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

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

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., $\forall e \in E(T^G)$:

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

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

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., $\forall e \in E(T^G)$:

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

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

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 $CW(e)$
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

L(a,b)

R(a,b)
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 $G$
Our Approximation Algorithm

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

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

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

\[ w(\ \ ) \geq 1/\gamma \cdot \max(\ \ ) \]
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 $T^G_G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G_G$

→ 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 $T^G$

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

This gives a $\gamma \cdot \beta$-approximation
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

$\to$ 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
Our Approximation Algorithm

1. Compute a $\gamma$-approximate MST $T^G_G$ 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 $T^G_G$ over the complete graph $G$
2. Compute a $\beta$-estimate of the cut weights of the edges in $T^G_G$
3. Compute the ultrametric tree using $T^G_G$ 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
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
For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulos in time $O(nd + n^{1+O(1/\gamma^2)})$
For any $\gamma \geq 1$, $\gamma$-spanner constructions of Har-Peled, Indyk, Sidiropoulous in time $O(nd + n^{1+O(1/\gamma^2)})$

$\beta = 5$-estimate using a variant of union-find data structure
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 $\frac{3}{2}$.
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 $3/2$.

Morally Equivalent to Search Version
Hardness from SETH

Theorem (Cohen-Addad–K–Lagarde)

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

YES: $X$ can be embedded \textit{isometrically} into an ultrametric.

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

Theorem (David–K–Laekhanukit’19)

Assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time, given $A, B \in \mathbb{R}^{O(\log n)}$ ($|A| = |B| = n$) 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 $\text{dist}(A, B) \in \{1, 3\}$.
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

ён

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 $\|a - b\|_{\infty} = 1$
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:** $\forall a, a' \in A$ 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 $\tau : S \rightarrow L$ 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 $\mathcal{B}_d$. 

- **NO** case:
  1. Sample $(0, 1, \ldots, 0)$ from $\mathcal{B}_d$.
  2. Pick distinct indices $8, 9, \ldots$ in $[n]$ at random.
  3. Let $0^8_9$ be the midpoint of $0^8$ and $0^9$.
  4. Let $0^\sim_8 : 9$ be $(1 - \rho) \cdot 0^8 : 9 + \rho \cdot 0^8_9$.
  5. Input is $(0, 1, \ldots, 0^\sim_8 : 9, 0 : 9 + 1, \ldots, 0)$. 

\[ \text{one.taboldstyle/four.taboldstyle} \]
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 $8, 9, \ldots$ in $[d]$ at random.
  - Let $0_{8,9}$ be the midpoint of $0_8$ and $0_9$.
  - Let $\tilde{0}_{8,9}$ be $(1 - (\cdot)) \cdot 0_{8,9} + \cdot 0_{8,9}$.
  - Input is $(0_1, \ldots, \tilde{0}_{8,9}, 0_{8,9} + 1, \ldots, 0_{8,9})$. 
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 \( \mathcal{B}_d \).

- **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 \( a_j \).
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 $\tilde{a}_k$ be $(1 - \rho) \cdot a_k + \rho \cdot a_{i,j}$.
  - 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 $d \geq O_{\rho,\varepsilon}(\log n)$. 
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

- **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 $1 + 2\delta$. 
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?
Open Problems

Improved Approximation Factor?

Euclidean Inapproximability under SETH?

More Applications of Colinearity Hypothesis?
THANK YOU!