Approximation Algorithms for Closest Metric Problems

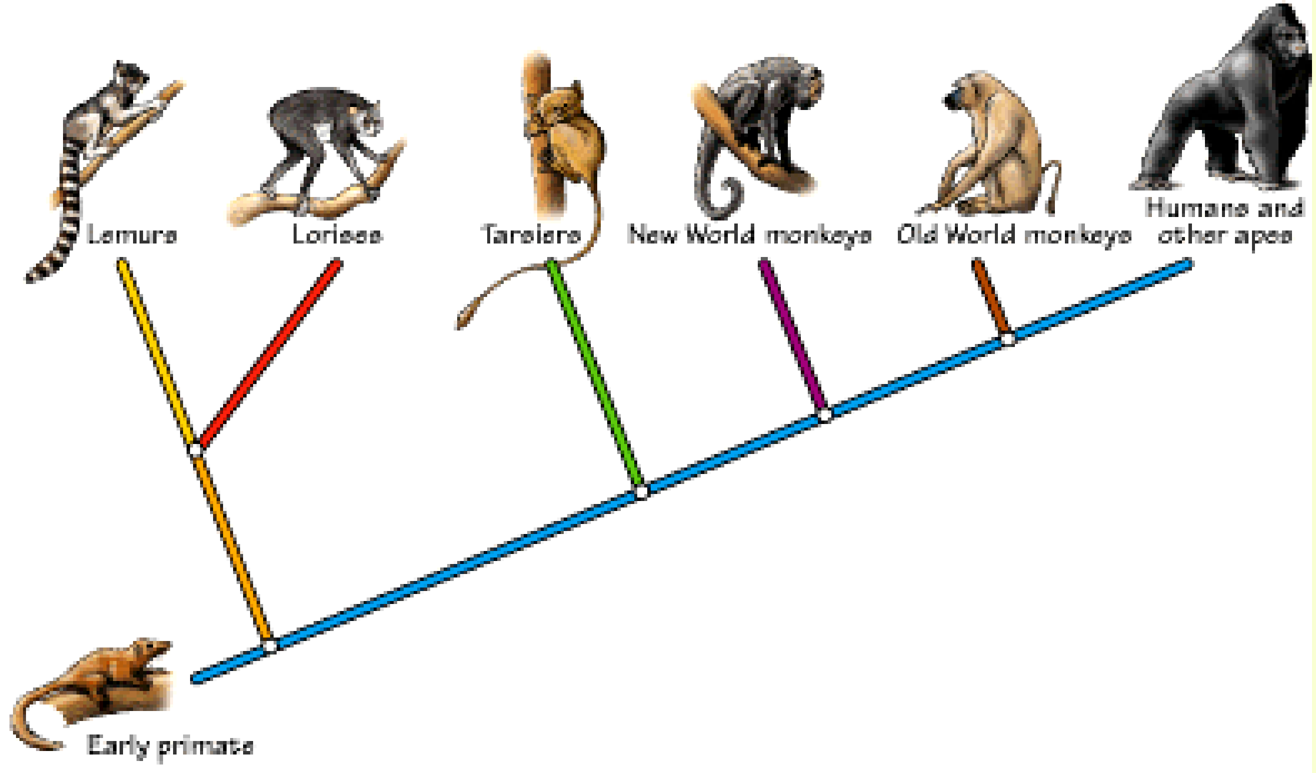
Kedar Dhamdhere

Outline of the talk

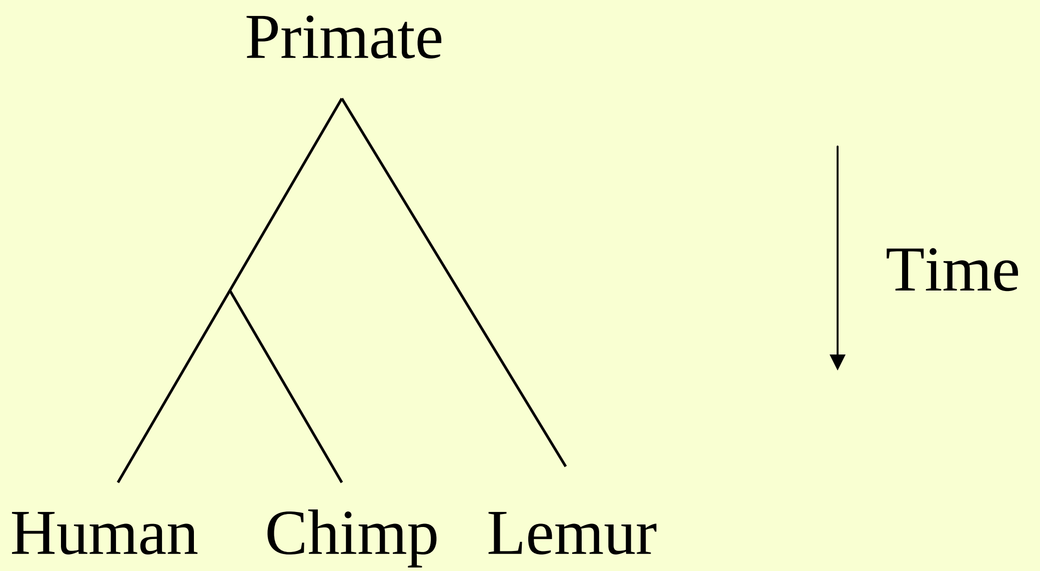
- Motivation
  - Evolutionary trees
- Problem definition & previous work
- Our results
- Conclusion

Motivation

Evolutionary tree



Evolutionary trees



All species evolved from one ancestor (root of the tree).

Length of the edges proportional to amount of time passed.

Finding evolutionary tree

- In practice, evolutionary time can be estimated using DNA sequences.
  - We get a table of pairwise distances.

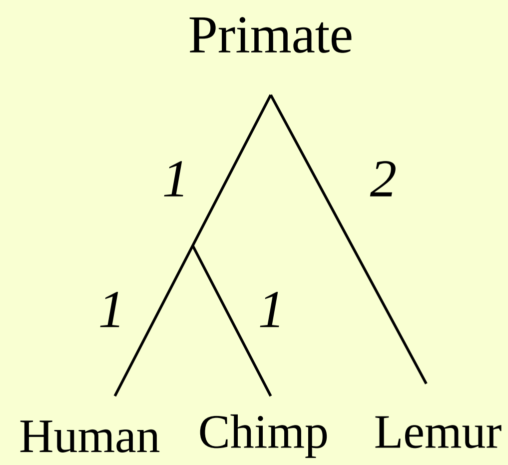
	Human	Chimp	Lemur
Human	0	2	4
Chimp	2	0	4
Lemur	4	4	0

Finding evolutionary tree

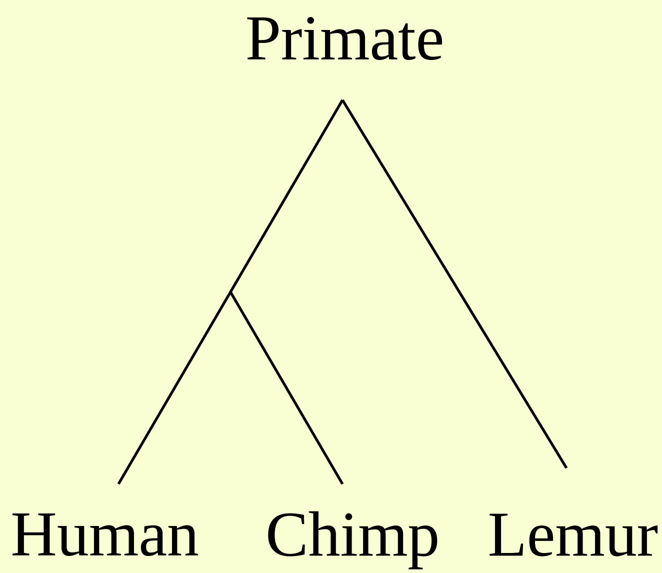
Input:
Distance matrix

	Human	Chimp	Lemur
Human	0	2	4
Chimp	2	0	4
Lemur	4	4	0

Output:
Evolutionary tree



Tree metric



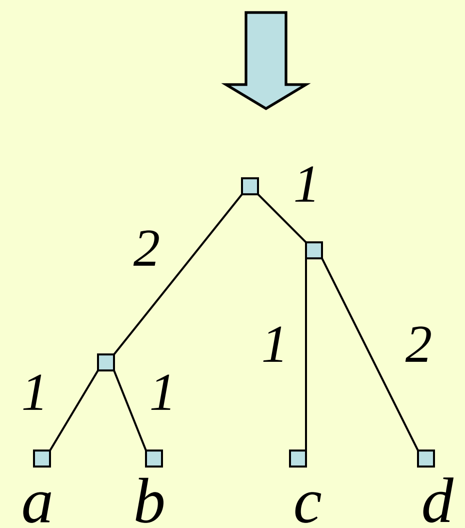
$dist_T(u,v)$ = length of the (unique) shortest path in the tree

Note: $dist_T(u,v) \leq dist_T(u,w) + dist_T(w,v)$

Fitting tree to input

Given $n \times n$ matrix D representing distances

$$
\begin{array}{c|cccc}
 & a & b & c & d \\
\hline
a & 0 & 2 & 5 & 6 \\
b & 2 & 0 & 5 & 6 \\
c & 5 & 5 & 0 & 3 \\
d & 6 & 6 & 3 & 0
\end{array}
$$



Find a tree T:

$$dist_T(i, j) = D[i, j]$$

Fitting tree to input

[Waterman-Smith-Singh-Beyer '77] $O(n^2)$-time algorithm to find a tree that fits the input data

In practice, no tree fits the data exactly

Find the *closest* tree metric

Outline of the talk

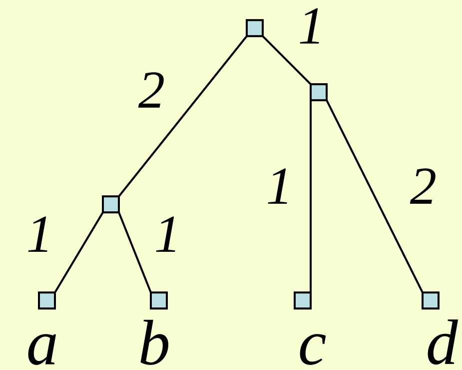
- Motivation
  - Evolutionary trees
- Problem definition & previous work
  - A special case – line metric
- Results
- Conclusion

Closest tree metric

Given $n \times n$ matrix D representing distances

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 2 & 5 & 6 \\ b & 2 & 0 & 5 & 6 \\ c & 5 & 5 & 0 & 3 \\ d & 6 & 6 & 3 & 0 \end{array}$$

Find a tree T closest to the input D



Closest tree metric

- What does closest mean?
  - Let $T_{n \times n}$ be the matrix of distances in the output tree.
  - L_p norm: $L_p(T, D) = (\sum_{i,j} |T[i,j] - D[i,j]|^p)^{1/p}$

    Important cases:

    - $p = 2$: sum of squared errors
    - $p = 1$: total error
    - $p = \infty$: $\max_{i,j} \{ |T[i,j] - D[i,j]| \}$

Previous work

- *[Day ’87], [Wareham ’93]* NP-hardness
- *[Farach-Kannan-Warnow ’93]* Polynomial time algorithm for a special case (*ultrametric)*
- *[Saitu-Nei ’87], [Felsenstein ’93], [Olsen et al ’94] , [Swofford ’98]* Hill-climbing heuristics
- *[Dress-Kruger ’87], [Strimmer-Haesler ’96], [Huson-Nettles-Warnow ’99]* Divide & conquer
- *[Lundy ’85], [Baker ’97], [Salter-Pearl ’00]* Simulated Annealing
- *[Yang-Rannala ’97], [Mau-Newton-Larget ’99], [Li-Pearl-Doss ’00]* Monte Carlo Markov Chain

Approximation algorithms

- An *approximation algorithm* for an NP-hard problem finds a near optimal solution quickly
  - Runs in polynomial time
  - Has a performance guarantee on quality of solution

- Performance Ratio: Worst-case performance ratio ρ of an approximation algorithm A for a minimization problem

$$= \max_{\text{input } I} \frac{\text{Value of solution}_A(I)}{\text{Value of optimal solution}(I)}$$

Previous work

- *[Agrawala-Bafna-Farach-Narayanan-Patterson-Thorup '95]* 3-approximation for finding closest tree under L_∞ norm

- Open: Approximate the closest tree metric under L_1 norm

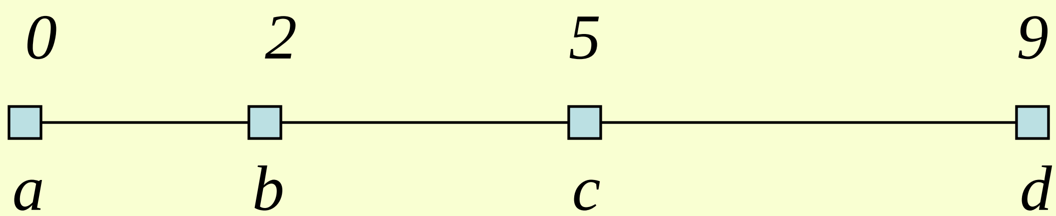
Previous work

- *[Agrawala-Bafna-Farach-Narayanan-Patterson-Thorup '95]*
  3-approximation for finding closest tree under L_∞ norm

- Open: Approximate the closest tree metric under L_1 norm

- Special Case: Find closest line metric under L_1 norm

Line metric



- $dist(x,y) = |x - y|$

- e.g. $dist(b,d) = 7$
  $dist(a,c) = 5$

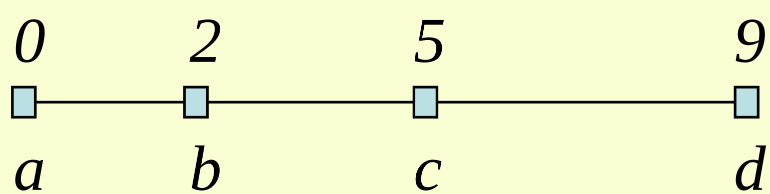
Closest line metric

Given $n \times n$ matrix D representing distances

$$
\begin{array}{c} \\ a \\ b \\ c \\ d \end{array}
\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 2 & 4 & 9 \\ 2 & 0 & 4 & 6 \\ 4 & 4 & 0 & 5 \\ 9 & 6 & 5 & 0 \end{pmatrix} \end{array}
$$

Convert to distances in line: $A_{n \times n}$

Minimize: $L_p(D,A)$



Previous work

[Håstad-Ivansson-Lagergren 98]

2-approximation for closest line metric under L_∞ norm

- Application to physical mapping of chromosomes
- Better approximation (e.g. 2-δ) is unlikely

Closest line metric (L_1)

Given $n \times n$ matrix D representing distances

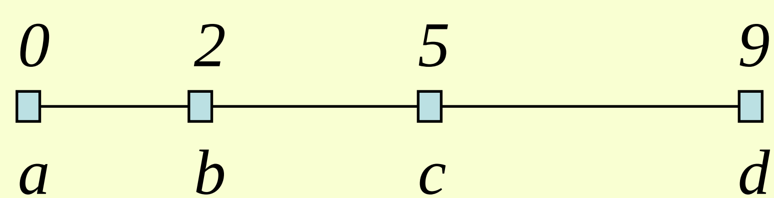
$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 2 & 4 & 9 \\ 2 & 0 & 4 & 6 \\ 4 & 4 & 0 & 5 \\ 9 & 6 & 5 & 0 \end{pmatrix} \end{array}$$

Convert to distances in line:

$A_{n \times n}$

Minimize:

$$L_1(A,D) = \sum_{i,j} |D(i,j) - A(i,j)|$$



This example: $L_1(A,D) = 8$

Closest line metric

Our results:

$O(\log n)$-approximation algorithm for closest line metric under L_1 norm

$O(\sqrt{\log n})$-approximation for sum of squared errors (L_2 norm) using same technique

$\vdots$

$O(\log^{1/p} n)$-approximation for L_p norm

Approximation for closest line metric

- Modify optimal solution to make it simpler (v-fixed)
  - Distances of all vertices from v are same as those in the input
  - Best v-fixed solution at most 3 times worse

- Approximate best v-fixed solution
  - Use multi-cut algorithm as a subroutine to get $O(\log n)$ approximation ratio

Open Questions

- Can we improve approximation: $O(\log n)$ to $O(1)$?
  - Replace multi-cut subroutine by something else?
- Approximation for tree metrics under L_p norm?

Monitoring Web Information Sources

- Dynamic nature of web
  - 23% of all pages change every day
- Monitoring information sources
  - Commuter updates: traffic and weather conditions
  - Alerts on baseball scores, stock portfolios
- Scheduling problem
  - How to schedule the crawling of web sources?
  - Maximize "timeliness" & "completeness" of information

Joint work with Sandeep Pandey, Christopher Olston

Credits

Thanks to **ALADDIN** for funding this work!