以最小ultrametric size 決定樹根之快速演算法
A fast algorithm for rooting a tree to minimize the ultrametric size

吳邦一(Bang Ye Wu)
樹德科技大學資訊工程系 Email:bangye@mail.stu.edu.tw

Abstract

For a given unrooted tree and observed distances among the species, we developed a fast algorithm for rooting the tree such that the size of the rooted ultrametric tree is minimum. The time complexity of the algorithm is $O(n^2)$, while a naive algorithm will take $O(n^3)$ time.
Keywords: algorithms, computational biology, ultrametric trees.

1 Introduction

Trees are used to represent evolutionary relationship and to guide the alignment of multiple sequences. The leaves of the tree represent the species and the internal nodes are the inferred ancestors. For constructing trees from observed distances, there are many different models which motivate algorithmic problems. However, most of the optimization problems of evolutionary tree construction have been shown to be NP-hard. Heuristic algorithms and computer softwares were developed to build rooted or unrooted trees by observed distances among the species. For example, PHYLIP [3] is one of the popular software packages, which contains several methods for building trees.

To guide the alignment of sequences, such as in the computer software CLUSTAL W[4], a tree should be rooted. An unrooted tree may be rooted at any edge. Trees obtained by rooting the same unrooted tree at different edges represents different grouping orders, and therefore should be considered as different. Usually the root of a tree may be determined by outgroups. We investigated how to determine the root by the distances.

The mathematical model we used is the *minimum ultrametric tree* [2]. An ultrametric tree is a rooted tree in which every internal node has the same path length to all the leaves in its subtree. The size of a tree is the sum of the length of all edges. For given observed distances among species, we hope to find the ultrametric tree with minimum size subject to that, for each pair of species, the distance on the tree is no less than the given one. To construct the minimum ultrametric tree for given distances had been shown to be NP-hard, and therefore it is very unlikely to find the optimal tree in reasonable time [2]. In [5], a branch and bound algorithm was developed to solve the problem for moderate number, about 20, of species.

The problem considered in this paper is much easier. In addition to the observed distances, an unrooted tree topology is also given. The goal is to root the tree at an edge and to give the length of each edge such that the rooted tree is an ultrametric tree and its size is minimum among all possible roots. It will be referred as the optimal root in the remaining of this paper. The optimal root may be not unique, and our goal is to find one of them.

To determine the optimal root, we may try every edge of the tree. Once the tree is rooted at an edge, the minimum ultrametric size with respect to the fixed topology can be computed in $O(n^2)$ time by an algorithm developed in [5], where n is the number of species. Consequently the optimal root can be determined in $O(n^3)$ time since there are only $O(n)$ edges in a tree with n leaves. In this paper, we present an $O(n^2)$ time algorithm for the problem.

2 Preliminaries

In this paper, by $T = (V, E)$ we denote an unweighted tree with vertex set V and edge set E. A tree with an edge weight function w is denoted by $T = (V, E, w)$. Let n denote the number of species. All the elements in a matrix and the weights on edges of a graph are assumed to be nonnegative. We first give some definitions as follows:

Definition 1 : A *distance matrix* of n species is a symmetric $n \times n$ matrix M such that $M[i, j] \geq 0$ for all $0 \leq i, j \leq n$, and $M[i, i] = 0$ for all $0 \leq i \leq n$.

Definition 2 : An $n \times n$ metric M is an ultrametric if and only if
$M[i, j] \leq \max\{M[i, k], M[j, k]\}$ for all $1 \leq i, j, k \leq n$. [1]

Definition 3: Let $T = (V, E, w)$ be an edge weighted tree and $u, v \in V$. The path length from u to v is denoted by $d_T(u, v)$. The size of T is defined by $w(T) = \sum_{e \in E} w(e)$.

Definition 4: Let T be a rooted tree and r be any node of T. we use T_r to denote the subtree rooted at r, and $L(T)$ to denote the leaf set of T.

Definition 5 : An *ultrametric tree* T of $\{1..n\}$ is a rooted and edge-weighted binary tree with $L(T) = \{1..n\}$ and root r such that $d_T(u, r) = d_T(v, r)$ for all $u, v \in L(T)$.

A rooted tree is binary if every internal node has exactly two children. An unrooted binary tree is a tree in which the degree of every internal node is exactly three. We consider only binary tree since any nonbinary tree can be easily transformed into a binary tree without changing the distances between leaves.

Let T be an ultrametric tree with root r. It is easy to see that for any internal node v, T_v is an ultrametric tree of $L(T_v)$. It should be noted that an $n \times n$ metric is ultrametric if and only if there is an ultrametric tree T of $\{1..n\}$ such that $d_T(i, j) = M[i, j]$ for all $1 \leq i, j \leq n$ [1]. By the definition of an ultrametric tree, the distances from an internal node r to all the leaves in T_r are the same. Therefore we can define the height of a node as follows:

Definition 6: Let $T = (V, E, w)$ be an ultrametric tree. For any $r \in V$, The height of r is the distance from r to any leaf in the subtree T_r.

The minimum ultrametric tree of a distance matrix was defined in [2].

Definition 7: For an n by n distance matrix M, an ultrametric tree T is an ultrametric tree of M if $L(T) = \{1..n\}$ and $d_T(i, j) \geq M[i, j]$ for all $1 \leq i, j \leq n$. T is the minimum ultrametric tree of M if the tree size is minimum among all ultrametric trees of M.

The next definition and two lemmas were shown in [5].

Definition 8: *Min Ultrametric Tree with a given Topology* (MUTT) problem:
Given a distance matrix M and a unweighted rooted tree $P = (V, E)$ with $L(P) = \{1..n\}$, the MUTT problem is to find a nonnegative edge weight function w of P such that $T = (V, E, w)$ is the minimum ultrametric tree of M.

Lemma 1: A tree T is a minimum ultrametric tree with respect to the fixed topology and distance matrix M if and only if the height of each internal node r is exactly $\max\{M[u, v]/2 \mid u, v \in L(T_r)\}$. [5]

Lemma 2: The MUTT problem, as well as the heights of all nodes of the minimum tree, can be computed in $O(n^2)$ time. [5]

The problem to be solved in this paper is formally defined in the following:

Definition 9: Given any distance matrix M and a unweighted unrooted tree $P = (V, E)$ with $L(P) = \{1..n\}$, the RMUT problem is to root P at one of its edges and to find a nonnegative edge weight function w for the resulted tree T such that T is an ultrametric tree of M and $w(T)$ is minimum among all possible roots and edge weight functions.

3 The algorithm

As mentioned in Section 1, the RMUT problem can be solved in $O(n^3)$ time. We shall reduce the time complexity to $O(n^2)$ in this section. The next property is helpful for improving the time efficiency.

Lemma 3: Let M be the distance matrix and $M[u, v]$ be maximal among all observed distances. The tree can be rooted optimally at some edge of the path between u and v on the tree.

Proof: Let T and r be an optimal tree and an optimal root of the RMUT problem respectively. By Lemma 1, the height of r is $M[u,v]/2$ since $M[u,v]$ is maximal. Also we have $d_T(u,v) = M[u,v]$. Therefore there is an internal node r_1 of the path between u and v on T, whose height is exactly $M[u,v]/2$. In the case that $r_1 \neq r$, since the heights of r and r_1 are the same, we may reroot T at r_1 and the size of the tree remains minimal. □

By the above lemma, the trees rooted at one of the edges of the path are candidates of the solution. However, the number of edges of the path may be up to $O(n)$. Computing all of the candidates individually takes also $O(n^3)$ time in worst case. The idea is to compute all the candidates in two passes. Let $M[u,v]$ be a maximal element of M and $(u = x_0, x_1, x_2, ..., x_k = v)$ be the path from u to v on T. For each vertex x_i, we first compute $f_1(i)$ as the minimum size of the subtree rooted at x_i if the optimal root is between x_i and v. Then we compute $f_2(i)$ as the minimum size of the subtree rooted at x_i if the optimal root is between x_i and u. Finally the minimum size of the whole tree rooted at edge (x_i, x_{i+1}) can be found by $f_1(i)$ and $f_2(i+1)$. The time complexity is reduced because the values $f_1(i)$ for all $0 \leq i \leq k$ can be computed in one pass. Similarly every value $f_2(i)$ can be found in the second pass. Our algorithm is listed below and illustrated in Figure 1:

Algorithm RootMUT
Input:A unweighted unrooted tree $T = (\{1..n\}, E)$ and a distance matrix M.
Output:A rooted tree with edge weights.
1: Find u,v such that $M[u,v]$ is a maximal element of M.
2: Find $(u = x_0, x_1, x_2, ..., x_k = v)$ which is the path from u to v on T.
3: Root T at edge (x_{k-1}, v). For every i, compute $f_1(i)$ to be the minimum size of the subtree rooted at x_i and $h_1(i)$ to be the height of x_i.
4: Root T at edge (u, x_1) . For every i, compute $f_2(i)$ to be the minimum size of the subtree rooted at x_i and $h_2(i)$ to be the height of x_i.
5: For every i, compute $f_1(i) + f_2(i+1) + M[u,v] - h_1(i) - h_2(i+1)$, which is the minimum size of the whole tree rooted at edge (x_i, x_{i+1}). Then find the optimal root by choosing the minimum.
6: Output the tree with the optimal root.

Theorem 4 : The algorithm RootMUT finds the optimal root for the RMUT problem in $O(n^2)$ time.

Proof: Apparently Step 1 takes $O(n^2)$ time and Step 2, 5, 6 take $O(n)$ time. By Lemma 2, Step 3 and 4 can be done in $O(n^2)$ time. Therefore the time complexity of the algorithm is $O(n^2)$. For the correctness of the algorithm, we shall show that $f_1(i)$ is the minimum size of the subtree rooted at x_i in the case that the optimal root is between x_i and v. Let e_1, e_2 be two edges of the path between x_i and v. For the two trees resulted by rooting T at e_1 and e_2 respectively, the leaf sets of the subtrees rooted at x_i are the same. By Lemma 1, the subtree rooted of x_i has the same minimum size once the root is between x_i and v. Therefore, in the case that the optimal root is between x_i and v, the minimum size of the subtree rooted at x_i is correctly given by $f_1(i)$. The correctness of $f_2(i)$ can be shown similarly. Let r be the root. The minimum size of the tree rooted at edge (x_i, x_{i+1}) is $f_1(i) + f_2(i+1) + w(r, x_i) + w(r, x_{i+1})$, in which $w(r, x_i) = M[u,v]/2 - h_1(i)$ and $w(r, x_{i+1}) = M[u,v]/2 - h_2(i+1)$ since the height of r is $M[u,v]/2$. □

4 Concluding remarks

It is interesting how to compute the minimum additive tree size of a given tree topology, instead of the restriction to ultrametric. It is obviously that such a problem can be solved by linear programming. But the algorithmic approach is still open. For the RMUT problem discussed in this paper, a C program based on algorithm RootMUT was written and ported on a PC running MS-DOS. The program, as well as some explanation and a sample input, are free and available at URL `http://www.personal.stu.edu.tw/~bangye/mutroot.htm`.

Acknowledgements

The work was partially supported by grant NSC 89-2218-E-366-003 from the National Science Council.

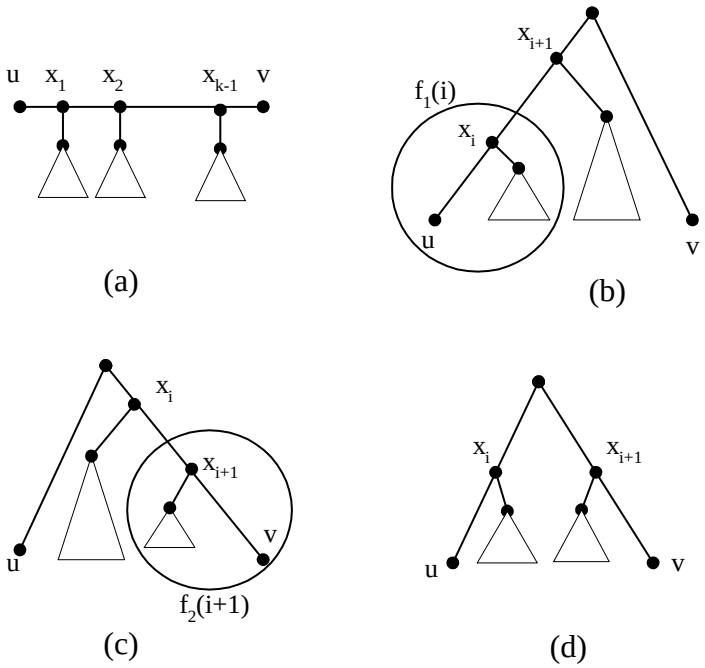

Figure 1: **(a):** Find the path between u and v on the tree. **(b):** Root the tree at the edge incident to v and compute $f_1(i)$, $h_1(i)$. **(c):** Root the tree at the edge incident to u and compute $f_2(i)$, $h_2(i)$. **(d):** The minimum size for rooting at edge (x_i, x_{i+1}) can be computed by $f_1(i)$, $f_2(i+1)$, $h_1(i)$ and $h_2(i+1)$.

References

[1] H.J. Bandelt, Recognition of tree metrics, *SIAM Journal on Discrete Mathematics.*, 3(1), 1–6, 1990.

[2] M. Farach, S. Kannan and T. Warnow, A robust model for finding optimal evolutionary trees, *Algorithmica*, 13, 155–179, 1995.

[3] J/ Felsenstein, PHYLIP — Phylogeny Inference Package (Version 3.2), *Cladistics*, 5, 164–166, 1989.

[4] J.D. Thompson, D.G. Higgins, and T.J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Research*, 22, 4673-4680, 1994

[5] B.Y. Wu, K.M. Chao and C.Y. Tang, Approximation and exact algorithms for constructing minimum ultrametric trees from distance matrices, *Journal of Combinatorial Optimization*, 3, 199–211, 1999.