Minimum Robinson-Foulds Distance Supertree

Xilin Yu, Thien Le, Sarah Christensen, Erin Molloy, Tandy Warnow

June 3, 2019

1 Introduction

2 The Maximum Bipartition Support Supertree Problem

2.1 Terminology and Preliminary

Throughout the paper, we consider only unrooted trees. For any tree T, let $V(T)$, $E(T)$, and $L(T)$ denote the vertex set, the edge set, and the leaf set of T, respectively. For any $v \in V(T)$, let $N_T(v)$ A tree is *fully resolved* if every non-leaf node has degree 3. Let $\mathcal{T}_S$ denote the set of all fully resolved trees on leaf set S. In any tree T, each edge e induces a bipartition $\pi_e := A|B$ of the leaf set, where A and B are the leaves in the two components of $T - e$, respectively. A bipartition $A|B$ is non-trivial if both sides have size at least 2. For a tree T, $C(T) := \{\pi_e \mid e \in E(T)\}$ denotes the set of all bipartitions of T. For a fully resolved tree with n leaves, $C(T)$ contains $2n - 3$ bipartitions, exactly $n - 3$ of which are non-trivial. A tree T' is a *refinement* of T if T can be obtained from T' by contracting a set of edges. Equivalently, T' is a refinement of T if and only if $C(T) \subseteq C(T')$.

Two bipartitions π_1 and π_2 of the same leaf set are *compatible* if and only if there exists a tree T such that $\pi_1, \pi_2 \in C(T)$. The following theorem and corollary give other categorizations of compatibility.

Theorem 1 (Theorem 2.20 of [1])**.** *A pair of bipartitions $A|B$ and $A'|B'$ of the same set is compatible if and only if at least one of the four pairwise intersections $A \cap A'$, $A \cap B'$, $B \cap A'$, $B \cap B'$ is empty.*

Corollary 1. *A pair of bipartitions $A|B$ and $A'|B'$ of the same set is compatible if and only if one side of $A|B$ is a subset of one side of $A'|B'$.*

A tree T restricted to a subset R of its leaf set, denoted $T|_R$, is the minimal subtree of T spanning R with nodes of degree two suppressed. A bipartition $\pi = A|B$ restricted to a subset $R \subseteq A \cup B$ is $\pi|_R = A \cap R|B \cap R$. We have the following intuitive lemma with its proof in the appendix.

Lemma 1. *Let T be a tree with leaf set S and let $\pi = A|B \in C(T)$ be a bipartition induced by $e \in E(T)$. Let $R \subseteq S$.*

1. *If $R \cap A \neq \emptyset$ and $R \cap B \neq \emptyset$, then for any $\pi' \in C(T|_R)$ induced by $e' \in E(T|_R)$, $\pi|_R = \pi'$ if and only if $e \in P(e')$.*

Definition 1. *For two trees T, T' with the same leaf set, the bipartition support of them is $bisup(T, T') := |C(T) \cap C(T')|$.*

Bipartition support measures the similarity between the topology of the trees.

2.2 Problem Statement

Let T_1 and T_2 be two fully resolved trees on leaf sets S_1 and S_2, respectively, such that $X := S_1 \cap S_2 \neq \emptyset$. Let $S := S_1 \cup S_2$. The Maximum Bipartition Support Supertree problem, abbreviated Max-Bisup-Supertree, finds a fully resolved supertree T^* on leaf set S that maximizes the sum of the bipartition support of T^* with respect to T_1 and T_2. That is,

$$
\begin{aligned}
T^* &= \underset{T \in \mathcal{T}_S}{\operatorname{argmax}}\, bisup(T|_{S_1}, T_1) + bisup(T|_{S_2}, T_2) \\
&= \underset{T \in \mathcal{T}_S}{\operatorname{argmax}}\, |C(T|_{S_1}) \cap C(T_1)| + |C(T|_{S_2}) \cap C(T_2)|.
\end{aligned}
$$

We call $bisup(T|_{S_1}, T_1) + bisup(T|_{S_2}, T_2)$ the support score of T when T_1 and T_2 are clear from context.

2.3 Algorithm

We first set up the notations for the algorithm and the analysis. Let T_1, T_2, S_1, S_2, and X be defined as from the problem statement. Let $T_1|_X$ and $T_2|_X$ be the backbone trees of T_1 and T_2, respectively. Let Π be the set of bipartitions of X. Let Triv and NonTriv denotes the set of trivial and non-trivial bipartitions in $C(T_1|_X) \cup C(T_2|_X)$. For each $e \in E(T_i|_X)$, $i \in \{1, 2\}$, let $P(e)$ denote the path in T_i from which e is obtained by suppressing all degree-two nodes. Let $w(e)$ be the number of edges on $P(e)$.

We define a weight function $w : \Pi \to \mathbb{N}_{\geq 0}$ such that for any bipartition π of X, $w(\pi) = w(e_1) + w(e_2)$, where e_i induces π in $T_i|_X$ for $i \in \{1, 2\}$. If for any $i \in \{1, 2\}$, no e_i exists that induces π in $T_i|_X$, then we use $w(e_i) = 0$.

For each $i \in \{1, 2\}$ and each $e \in E(T_i|_X)$, let $\text{In}(e)$ be the set of internal nodes of $P(e)$. For each $v \in \text{In}(e)$, let $L(v)$ be the set of leaves in $S_i \backslash X$ whose connecting path to the backbone tree $T_i|_X$ goes through v and let $T(v)$ be the minimal subtree spanning $L(v)$ in T_i. We say $T(v)$ is an extra subtree attached to v. Consider $T(v)$ rooted at the node u which is the neighbor of v in $T(v)$. Let $\mathcal{T}(e) := \{T(v) \mid v \in \text{In}(e)\}$. Then $\mathcal{T}(e)$ is the set of extra subtrees attached to internal nodes of $P(e)$ in T_i. We note that $|\mathcal{T}(e)| = |\mathrm{In}(e)| = w(e) - 1$. For any bipartition $\pi \in C(T_1|_X) \cup C(T_2|_X)$, we denote $\mathcal{T}(\pi) := \mathcal{T}(e_1) \cup \mathcal{T}(e_2)$, where e_i is the edge that induces π in $T_i|_X$ for $i \in \{1, 2\}$ if $\pi \in C(T_i|_X)$. Let $\mathrm{Extra}(T_i) := \bigcup_{e \in E(T_i|_X)} \mathcal{T}(e)$. Then $\mathrm{Extra} := \mathrm{Extra}(T_1) \cup \mathrm{Extra}(T_2)$ denotes the set of all extra subtrees in T_1 and T_2. figure to help

Algorithm 1 Max-BiSup Supertree

Input: two fully resolved trees T_1, T_2 with leaf sets S_1 and S_2 where $S_1 \cap S_2 = X \neq \emptyset$

Output: a fully resolved supertree T on leaf set $S = S_1 \cup S_2$ that maximizes the support score

compute $C(T_1|_X)$ and $C(T_2|_X)$
for each $\pi \in C(T_1|_X) \cup C(T_2|_X)$ **do**
    compute $\mathcal{T}(\pi)$ and $w(\pi)$
construct T by having a star of leaf set X with center vertex $\hat{v}$ and connecting the root of each $t \in \mathrm{Extra}$ to $\hat{v}$
for each $\pi \in \mathrm{Triv}$ **do**
    $T \leftarrow \text{Refine-Triv}(T, \pi, \mathcal{T}(\pi))$
construct the incompatibility graph $G = (V_1 \cup V_2, E)$, where $V_1 = C(T_1|_X) - C(T_2|_X)$ and $V_2 = C(T_2|_X) - C(T_1|_X)$, and $E = \{(\pi, \pi') \mid \pi \in V_1, \pi' \in V_2, \pi \text{ is not compatible with } \pi'\}$
compute the maximum weight independent set I in G with weight w
let $H(\hat{v}) = \mathrm{NonTriv} \cap (C(T_1|_X) \cup C(T_2|_X))$
let $R(\hat{v}) = \emptyset$
for each $\pi \in \mathrm{NonTriv} \cap (C(T_1|_X) \cup C(T_2|_X))$ **do**
    $sv(\pi) = \hat{v}$
    add the root of each $t \in \mathcal{T}(\pi)$ to $R(v)$
for each $\pi \in \mathrm{NonTriv} \cap (I \cup (C(T_1|_X) \cap C(T_2|_X)))$ **do**
    $T \leftarrow \text{Refine}(T, \pi, H, sv)$
refine T arbitrarily at polytomies until it is fully resolved
return T

Algorithm 2 Refine-Triv

A Proofs from Section 2

Proof of Lemma 1

Proof. Let T_R be the minimal subtree of T that spans R. It follows that the leaf set of T_R is R and $T|_R$ is obtained from T_R by suppressing all degree-two nodes. Let $\pi' = A'|B'$. By definition of e inducing $\pi = A|B$, the vertices of A are all disconnected from vertices of B in $T - e$. If $R \cap A \neq \emptyset$ and $R \cap B \neq \emptyset$, then e is

Algorithm 3 Refine

Input: two trees T_1, T_2 with leaf sets S_1 and S_2 where $S_1 \cap S_2 = X \neq \emptyset$, an unrooted tree T on leaf set $S = S_1 \cup S_2$, a bipartition $\pi = A|B$ of X, a dictionary H, a dictionary sv

Output: an tree T' which is a refinement of T such that $\pi \in C(T'|_X)$

$v \leftarrow sv(\pi)$
compute $N_A := \{u \in N_T(v) \mid \exists a \in A \text{ such that } u \text{ can reach } a \text{ in } T - v\}$ and $N_B := \{u \in N_T(v) \mid \exists b \in B \text{ such that } u \text{ can reach } b \text{ in } T - v\}$.
$V(T) \leftarrow V(T) \cup \{v_a, v_b\}$, $E(T) \leftarrow E(T) \cup \{(v_a, v_b)\}$
$H(v_a) \leftarrow \emptyset, H(v_b) \leftarrow \emptyset$
for each $u \in N_A \cup N_B$ **do**
    if $u \in N_A$ **then** connect u to v_a
    else connect u to v_b
detach all extra subtrees in $\mathcal{T}(\pi)$ from v and attach them onto (v_a, v_b) such that the subtrees from $\mathcal{T}(e_1)$ and subtrees from $\mathcal{T}(e_2)$ are side by side and each group respects the ordering of subtrees in T_i
for each bipartition $\pi' = A'|B' \in H(v)$ such that $\pi' \neq \pi$ **do**
    detach all extra subtrees in $\mathcal{T}(\pi')$ from v
    if $A' \subseteq A$ or $B' \subseteq A$ **then**
        $sv(\pi') = v_a$ and $H(v_a) \leftarrow H(v_a) + \pi'$
        attach all extra subtrees in $\mathcal{T}(\pi')$ to v_a
    else if $A' \subseteq B$ or $B' \subseteq B$ **then**
        $sv(\pi') = v_b$ and $H(v_b) \leftarrow H(v_b) + \pi'$
        attach all extra subtrees in $\mathcal{T}(\pi')$ to v_b
    else
        discard π' and attach all extra subtrees in $\mathcal{T}(\pi')$ to either v_a or v_b
for each remaining extra subtree attached to v **do**
    detach it from v and attach it to either v_a or v_b
delete v and incident edges from T
return the resulting tree T'

necessary to connect $R \cap A$ with $R \cap B$, and thus e must be in any tree spanning R and in particular $e \in E(T_R)$. Since T_R is a subgraph of T, the two components in $T_R - e$ are subgraphs of the two components in $T - e$. Thus, the leaves of the two components in $T_R - e$ are exactly $R \cap A$ and $R \cap B$. We also know that suppressing degree-two nodes does not change the connectivity between any leaves so the leaves of the two components in $T_R - P(e')$ (with vertices on the path also deleted) are the same as the leaves of the two components in $T|_R - e'$, which are A' and B'. If $e \in P(e')$, since all internal nodes of $P(e')$ have degree two with both incident edges on $P(e')$, there is no leaf which exists in any of the two components in $T_R - e$ but does not exists in the corresponding component in $T_R - P(e')$. Therefore, $\pi|_R = R \cap A | R \cap B = A'|B' = \pi'$. If $e \notin P(e')$, then since $e \in E(T_R)$, there must exists $e'' \in E(T|_R)$ such that $e'' \neq e'$ and $e \in P(e'')$. By the arguement above, $\pi|_R = \pi''$ where π'' is the bipartition induced by e'' in $T|_R$. Since $e'' \neq e'$, we know $\pi' \neq \pi''$ and thus $\pi|_R \neq \pi'$. This concludes our proof that $\pi|_R = \pi'$ if and only if $e \in P(e')$. $\square$

References

[1] Tandy Warnow. *Computational phylogenetics: an introduction to designing methods for phylogeny estimation.* Cambridge University Press, 2017.