Learning and Approximation Algorithms for problems motivated by Evolutionary Trees

by

Mary Elizabeth Cryan

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Computer Science

October 1999

Contents

Acknowledgments

The biggest acknowledgement is due to my supervisor, Dr Leslie Ann Goldberg. I am very grateful to Leslie for getting me started doing research and for collaborating with me on the research presented in this thesis. Leslie also carefully proofread the draft of this thesis and her comments led to many improvements in the presentation of the thesis. I am also grateful to my co-authors Paul W. Goldberg and Cynthia A. Phillips for the experience I had of working with them.

I really enjoyed the experience of studying for my PhD in the Algorithms and Computational Complexity group at Warwick. Thanks must go to the other PhD students of my group (Graham, Hesham and Jonathan) for being such nice officemates while I was writing this thesis. I would also like to thank Mike Paterson, Hristo Djidjev and Cenk Sahinalp, and all the other people that have passed through our group while I was here. All these people helped to provide a stimulating environment for research.

Also, I would like to thank my non-academic friends, especially my flatmates Dom and Ben, for listening to me talking about this thesis for so long. I am also very grateful to my parents for their support during these four years of further study.

Finally, I would like to thank the department of Computer Science at the University of Warwick for funding my PhD.

Declarations

This thesis is being submitted to the University of Warwick in fulfilment of the requirements of the degree of Doctor of Philosophy. No part of this thesis has been submitted in support of an application for any other degree or qualification at this institution or any other institution of learning. Some parts of the thesis have already appeared in jointly-authored papers. The research described in Chapter 2 is based on the results presented in:

Mary Cryan, Leslie Ann Goldberg and Paul W. Goldberg, "Evolutionary Trees can be Learned in Polynomial Time in the Two-State General Markov Model". In *Proceedings of the 38th Annual IEEE Symposium on the Foundations of Computer Science*, pages 436–445, 1998.

The research on labelling fixed-topologies presented in Sections 3.3 and 3.4 appeared in the following paper:

Mary Cryan, Leslie Ann Goldberg and Cynthia A. Phillips, "Approximation Algorithms for the Fixed-Topology Phylogenetic Number Problem". In *Algorithmica*, volume **25**(2), pages 311–329, (1999). (A preliminary version appeared in *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching*, pages 130–149, volume **1264**, Springer-Verlag, (1997).)

Abstract

In this thesis we consider some computational problems motivated by the biological problem of reconstructing evolutionary trees. In this thesis, we are concerned with the design and analysis of efficient algorithms for clearly defined combinatorial problems motived by this application area. We present results for two different kinds of problem.

Our first problem is motivated by models of evolution that describe the evolution of biological species in terms of a stochastic process that alters the DNA of species. The particular stochastic model that we considered is called the Two-State General Markov Model. In this model, an evolutionary tree can be associated with a distribution on the different "patterns" that may appear among the sequences for all the species in the evolutionary tree. Then the data for a collection of species whose evolutionary tree is unknown can be viewed as samples from this (unknown) distribution. An interesting problem asks whether we can use samples from an *unknown* evolutionary tree M to find another tree $\widehat{M}$ for those species, so that the distribution of $\widehat{M}$ is similar to that of M. This is essentially a PAC-learning problem ("Probably Approximately Correct") in the sense of Valiant [60] and Kearns et al. [46]. Our results show that evolutionary trees in the Two-State General Markov can be efficiently PAC-learned in the variation distance metric using a "reasonable" number of samples.

The two other problems that we consider are combinatorial problems that are also motivated by evolutionary tree construction. The input to each of these problems consists of a fixed tree topology whose leaves are bijectively labelled by the elements of a species set, as well as data for those species. Both problems involve labelling the internal nodes in the fixed topology in order to minimize some function on that tree (both functions that we consider are assumed to test the quality of the tree topology in some way). The two problems that we consider are known to be NP-hard. Our contribution is to present efficient approximation algorithms for both problems.

Chapter 1

Introduction

1.1 Overview

The subject of this thesis is finding efficient algorithms for some problems on trees. The first problem that we consider is a computational learning theory problem about *learning* classes of probabilistic distributions defined in terms of a tree topology. Another problem that we consider involves finding polynomial-time algorithms that label a known tree in order to approximate a minimal-cost labelling under two basic cost functions.

Although these problems involve quite different types of analysis, they are motivated by the same application, which is the field of *computational molecular biology*. In particular, the research presented in this thesis is motivated by the problem of constructing evolutionary trees for groups of related biological species. We are interested in clearly defined combinatorial problems that arise in connection with this application area; our interest lies in designing *efficient* algorithms to solve the problem. For the approximation algorithms that we develop, this means that the algorithm should run in time that is polynomial in the size of the input. Since a learning algorithm is given samples from the distribution as part of its input, we will also insist that only a *reasonable* number of samples should be needed to learn a distribution.

The layout of this Chapter is influenced by the fact that the learning problem requires more explanation than the other problems. We begin in Section 1.2 by explaining the biological motivation for our research. Section 1.3 gives details of the learning problem, and Section 1.4 explains the layout of the thesis.

1.2 Biological Background

During the last ten years, a lot of research has been carried out on the development of computational techniques for molecular biology research. This research effort, which includes research into algorithmic and combinatorial problems as well as software development, is called *computational molecular biology*. A lot of the motivation for computational biology research has come from the Human Genome Project, which aims to sequence the DNA of humans and to use this information to understand genes and their functions. Karp [44] and Pevzner and Waterman [56] have written surveys on the computational problems motivated by the Human Genome Project. Another biology problem that has attracted the interest of researchers in the theoretical computer science community is the issue of using species data to infer evolutionary relationships among groups of biological species. This latter problem is the motivation for the research presented in this thesis.

We will always assume that the evolutionary history of a group of related biological species can be represented as a rooted tree (see Figure 1.1 for an example). Although disagreement about the true nature of evolution has generated much controversy, this is generally considered to be a reasonable assumption. The root of the tree represents the ancient species from which all the other species have evolved. Any internal node of the tree represents a speciation event which splits the original species at that node into two or more new species, depending on the number of outgoing edges from the internal node. The leaves of the tree are bijectively labelled by the group of species for which data is available.

Until recently, most biologists relied on heuristic methods for constructing evolutionary trees. Many of these do not even provide performance guarantees for the quality of the solution returned. One group of methods that do have performance guarantees are the Maximum Likelihood Estimation (MLE) methods [27, 38]: these methods construct the tree that is *most likely* to have produced the data, given some assumptions about the evolutionary process. There are certainly many open computational problems motivated by evolutionary tree construction. The type of computational problem that is important depends on two factors: the type of species data that is available; and whether or not there is a model for the evolutionary process.

1.2.1 Data

The data used by procedures for inferring evolutionary trees may be either morphological or biomolecular in origin.

Morphological data

This is data that has been obtained by observing the species and classifying them by a number of different traits or characteristics. For example, one characteristic for the birds of prey (see Figure 1.1) is the "inside-egg colour" characteristic, which classifies the birds according to the tint of the inside of their eggs. This would be represented by a *characteristic function*, which is also called a *character*, from the birds of prey to the set of possible *charac-*

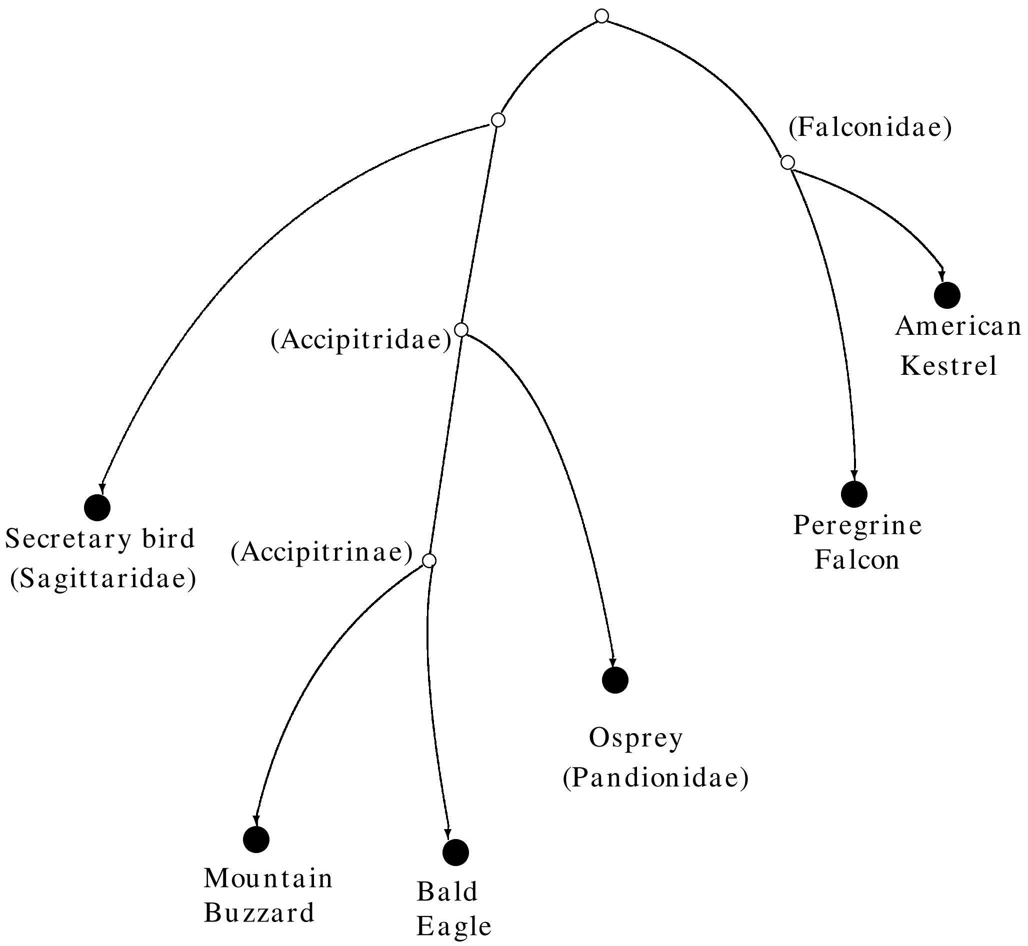


Figure 1.1: Here is an evolutionary tree for some of the birds of prey, based on the DNA-based taxonomic classification of birds described by Monroe and Sibley [53]. There are almost 300 different birds of prey, but we only show a few of them here. The root of this tree represents a hypothetical "least common ancestor" for all these birds. There are over 100 birds in the Accipitrinae subfamily (only the Bald eagle and the Mountain Buzzard are shown here). The Osprey is the only bird in the Pandionidae subfamily (see Monroe and Sibley [53]).

ter states $\{greenish, reddish - yellow, white, \ldots\}$. For this example, the value of $inside - egg(x)$ would be $greenish$ for every bird x in the Accipitridae family. The data for a group of species will always consist of a collection of different characteristics. The significance of morphological data rests on the assumption that any particular character state is unlikely to have evolved very often and that therefore a group of species that share a character state have probably evolved together.

The problem of constructing evolutionary trees also arises in relation to the evolution of natural languages. Characteristic functions can be defined by making observations about the words used in different languages for a single semantic concept. Some of the research described in Chapter 3 is related to the problem of finding evolutionary trees for natural languages. Another situation where "species" may be described by characteristic functions arises when an ancient manuscript has been copied out over a long period of time, and errors in the copying process have given rise to different versions of the manuscript; in this case, scholars are interested in using the different versions of the manuscript to find the original manuscript. An article in "The Times" by Hawkes [37] reports on a successful computer-aided reconstruction of the evolution of Chaucer's "Canterbury Tales".

Biomolecular data

Many different types of data collected from the cells of organisms contain valuable information about the evolutionary relationships among groups of species. The chromosomal DNA within a cell contains the hereditary information that is passed on to the descendants of the organism, so it is generally assumed that the DNA fragments of a group of species carry information about the evolutionary history of those species. However, because DNA controls the production of proteins within cells, some by-products of the manufacturing process may also carry information about the evolutionary process. The next paragraph sketches a few details about the protein manufacture process, and further details can be found in an article by Brown [10].

Every cell contains a certain number of chromosomes, and each of these contains a double-stranded DNA sequence. Chromosomal DNA contains the genetic information that organisms inherit from their ancestors; the number of chromosomes in a cell depends on the species. Human cells have 23 pairs of chromosomes, and every pair of these chromosomes contains one chromosome from each parent. In its natural form, a DNA sequence folds into a three-dimensional structure, but we will think of it as a sequence over the alphabet $\{a, c, g, t\}$ of *nucleotides*. A *gene* is a fragment of the DNA sequence that codes for a specific protein; the protein is manufactured whenever the gene is expressed. During these periods, a copying process constructs a sequence over the nucleotide set $\{a, c, g, u\}$, called an RNA sequence, from the gene sequence. The RNA sequence is made by copying one nucleotide of the gene at a time, replacing occurrences of t with u. Then the RNA sequence fragment makes its way to the ribosome (the "protein factory") of the cell and is used as the code for putting together the *amino acids* that make up the protein. This translation of the RNA into a protein sequence is possible because every sequence of three nucleotides makes one of the 20 amino acids (there are only 20 because some amino acids have alternative forms). This natural mapping from $\{a, c, g, u\}^3$ to the set of amino acids is called the *genetic code*.

It is known that two members of the same species have very similar DNA (see Felsenstein [28]), so we can use the chromosomal DNA of a particular organism as the data for its entire species. Also, because the RNA and protein sequences within a cell are derived from genes, these sequences may be used for reconstructing evolutionary relationships between species. Many scientists, including Jukes and Cantor [40], Kashyap and Subas [45], Kimura [49] and Waterman, Smith, Singh and Beyer [66], have been interested in reconstructing the evolutionary history of groups of proteins. Although there is interest in the evolution of proteins as a problem in its own right, further motivation is provided by the hope that the evolutionary history of a group of proteins found in different organisms may provide insight into the evolution of those organisms (see Jukes and Cantor [40]).

In Subsection 1.2.2, we will see that most models of evolution describe the evolutionary process as a stochastic process on the sequence data of a species. The data generated by such a process will consist of *aligned* sequence fragments for all the species. In this context, alignment is defined in terms of the sequence for the ancestral species at the root of the unknown evolutionary tree. The fragments for a group of species are aligned if they have all evolved from the same fragment of the sequence at the root. Finding sequence fragments with a common evolutionary history is a difficult problem in its own right (see Karp [44]), but we will assume that aligned sequence data is available. This is a common assumption: most early research (for example, Cavender [12]) and also most recent research (for example, Farach and Kannan [25]) depends on this assumption.

1.2.2 Models and Methods

Before asking how we can reconstruct evolutionary trees from species data, we need to define some formal relationship between the true evolutionary tree and the species data. Once we have a definition, many questions can be asked: Given data for a group of species, is it possible to determine the true evolutionary tree for those species? If not, what related information can we obtain? The question that is most important to us is whether or not we can design efficient algorithms for solving problems that interest us.

Since the 1960s, systematic biologists have come up with different ways of formally relating species data and evolutionary trees. It is most common to provide a *model* for the evolutionary process, which describes the type of processes that are assumed to be responsible for introducing changes in species data, and therefore are ultimately responsible for generating data for the leaf species of the tree. An alternative approach sidesteps the problem of modelling evolution, and instead assumes that evolutionary changes are very unusual. Under this assumption, the data provided for a group of species implicitly defines some optimal tree(s) that may correspond to the evolutionary tree for those species.

Models

Most models of evolution for biological species are motivated by the general belief that evolution proceeds as a *stochastic process* that randomly induces changes in the DNA and related data of a species. These models usually represent each of the leaf species and ancestral species by a sequence over a finite alphabet. The elements of this alphabet are often called *states*. There are also some stochastic models, such as the model presented by Cavender [12], that view evolution as a process that induces changes in morphological characteristics.

The point of modelling evolution is to formalise the assumptions that are made about the evolutionary process. It is always assumed that evolution proceeds independently along different lineages of the tree, so we only need to decide the type of stochastic process that acts on a single edge. An early model for these edge processes was developed in 1969 by Jukes and Cantor [40]. Jukes and Cantor were interested in modelling the evolution of groups of proteins, and represented proteins by their RNA sequences. The main feature of Jukes and Cantor's model [40] is that every edge process in a tree is assumed to be an *independently* and *identically* distributed (*iid*) stochastic process that induces *mutations* in RNA sequences:

Mutations A transition at a sequence position occurs when the nucleotide at that position is replaced by a different nucleotide from $\{a, c, g, u\}$. The process in the edge $(v \to w)$ is mutational if the sequence at v is transformed into the sequence at w by a number of nucleotide changes.

Jukes and Cantor point out that the true evolutionary process induces deletions and insertions in DNA and RNA sequences, as well as mutations. In the stochastic models of evolution that we consider in this thesis, sequences are only modified by simple mutational processes. This is true of the Jukes-Cantor model, and of all the models of evolution that will be discussed in this section and Section 2.1. (In Chapter 3 we consider a combinatorial problem in which species data is altered by deletions and insertions. However, we do not assume any model of evolution for this particular problem).

Independent The probability that a transition takes place at a specific position (or *site*) in the RNA sequence should not depend on the actual transitions that are taking place elsewhere in the sequence.

Identical Finally, we assume that although the stochastic process at a site proceeds independently of the process at other sites, the *probability* of a transition from the nucleotide a to the nucleotide g is the same at every site. This is also true for the other eleven possible transitions.

Most stochastic models of evolution that have been developed make these three

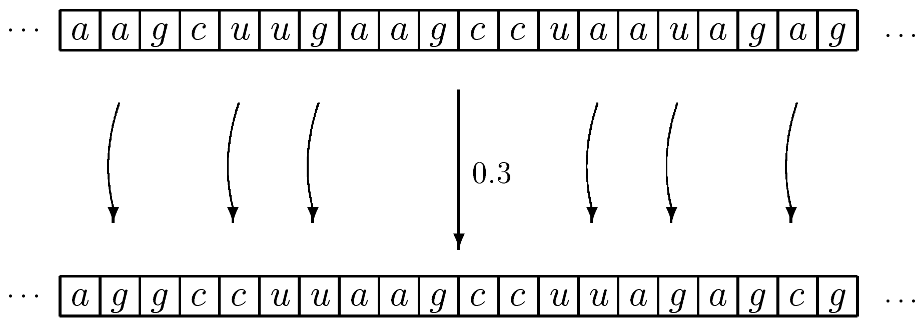


Figure 1.2: The Jukes-Cantor process on a single edge with probability 0.3.

assumptions, although the sequences may be DNA sequences, protein sequences or even sequences of morphological characteristics. For example, the models defined in the papers by Cavender [12], Farris [26], Felsenstein [27, 28], Hendy, Penny and Steel [38], Kashyap and Subas [45], Kimura [49], Neyman [55] and Steel [58] all assume that edge processes are iid mutational processes.

Any independently and identically distributed process on an edge can be described by specifying, for every pair of states i and i', the probability that an i-state in the ancestral sequence becomes i' in the lower sequence. In the **Jukes-Cantor model** [40], a tree only has one transition probability per edge: on any edge of the tree, the probability of the transition $i \to i'$ is the same for every pair of states with $i \neq i'$. Therefore, if there are j distinct states, every edge can be represented by a single probability p such that the probability of *any* mutation is $p/(j-1)$, and the probability that a state does not change is $1-p$. Figure 1.2 shows how the Jukes-Cantor process for $p = 0.3$ might behave on an RNA sequence fragment of length 20. Remember that there are four RNA states. To describe the Jukes-Cantor process for $p = 0.3$, imagine that we have four different "four-sided" coins, the a-coin, the c-coin, the g-coin and the u-coin. Each of these coins has its sides labelled by the four states a,

c, g, and u. The a-coin is biased towards a so that a flip of the a-coin brings up a with probability 0.7, while c, g and u each have probability 0.1. The other coins are also defined by probabilities 0.7 and 0.1, though the c-coin is biased towards c, the g-coin is biased towards g, and the u-coin is biased towards u. To determine the state of a site in the descendant sequence, we perform the following experiment: choose the biased four-sided coin that corresponds to the original RNA state for this site. Then flip this coin and let the state of the site in the descendant sequence be the result of the flip. For example, if we consider the two leftmost sites shown in Figure 1.2, the states of these sites in the descendant sequence would have been determined by two independent flips of the a-coin. We can see that the result of the first flip was another a, and that the flip for the second site produced a g.

The models of Cavender [12], Farris [26] and Neyman [55] are essentially the same as the Jukes-Cantor model, though Farris and Neyman consider sequences over a general alphabet, and Cavender's model is defined for sequences of binary morphological characters. The Two-State version of the Jukes-Cantor process is usually called the Cavender-Farris model or the Neyman model in the literature. Most of the other models of evolution do not insist that all mutations are equally likely, because it is generally accepted that this is not true for real DNA and RNA sequences (see Jukes-Cantor [40], Swofford et al. [59]). The most general iid model of all is the General Markov Model, which is due to Steel [58], and this model is described in more detail in Section 1.3.

The advantage of having a model of the evolutionary process is that when we are given a group of sequence fragments that have evolved in this model, we know the structure of the process that has generated the fragments. In Section 1.3, we will show that when we are given an evolutionary tree from an iid model, and we also know the transition properties for the edges of this tree, we can calculate the probability of seeing the different "patterns" that appear among the sequences at the leaves. Intuition tells us that when we are given sequence fragments from the leaf species of an *unknown* evolutionary tree, the observed probability of these patterns among the sequence fragments might allow us to make inferences about the underlying evolutionary tree. In Chapter 2 we will see that the exact topology cannot always be reconstructed from the sequences at the leaves; instead, we will consider a related learning problem, which is described in Section 1.3.

Before the recent interest in these problems, most of the procedures available for constructing evolutionary trees were heuristic methods without performance guarantees. There were some exceptions. For example, Maximum Likelihood Estimation procedures were developed which take a collection of sequence fragments as input and construct the *most likely* tree (for a predetermined iid model) to have generated that data (see Felsenstein [27]). The method is consistent: as the sequence fragments provided for a group of species become longer, the most likely tree will converge to a single topology. Recently there has been further research on the problem of learning evolutionary trees in iid models, and this will be discussed in Section 2.1.

Minimizing changes

The true nature of evolution is still not completely understood. It is especially difficult to explain changes in morphology in terms of a stochastic model: although morphological changes probably occur as a result of a substantial number of mutations in the DNA of a species, it doesn't seem likely that a single morphological change should correspond to a fixed number of nucleotide mutations in DNA. Therefore, when the data available on a group of species consists of observations about the morphology of those species, systematic biologists do not interpret the species data in terms of a model.

In this situation, biologists usually approach the problem of constructing evolutionary trees by making the assumption that the true evolutionary tree is a tree that minimizes the number of evolutionary changes *in some sense*. A hypothetical evolutionary tree consists of a tree topology whose leaves are bijectively labelled by the species of interest; also, when the species data is a set of characters, each character c must be extended so that $c(v)$ is also defined for every internal node v. An evolutionary change for character c occurs along an edge if the endpoints of any edge in the tree have different states under c.

The strongest assumption that can be made about the evolution of a group of species is that their evolutionary tree should form a *perfect phylogeny*: this is an evolutionary tree in which every character state forms a single connected component in the tree. One problem that has been widely studied is the computational complexity of determining, for a group of species and a collection of characteristic functions, whether or not a perfect phylogeny exists for that data. This problem has been studied by Gusfield [35], Kannan and Warnow [42, 41], Bodlaender et al [7], Steel [57], and Agarwala and Fernández-Baca [3]. The existence of a perfect phylogeny is based on a very strong assumption about the evolutionary process, so it makes sense to consider problems that place slightly weaker constraints on the evolutionary tree. Goldberg, Goldberg, Phillips, Sweedyk and Warnow [32] introduced the concept of *phylogenetic number*: the phylogenetic number of a labelled evolutionary tree is the minimum number ℓ such that every state of every character forms at most ℓ connected components in the tree. Goldberg et al [32] presented results on the problem of finding a tree with the minimum phylogenetic number, and these results are discussed in Section 3.2. An alternative generalisation of the perfect phylogeny problem is to ask for a tree with the minimum number of evolutionary changes, summed over all characters and all character states. Research on this problem, which is known as the *parsimony* problem, has appeared in papers by Graham and Foulds [33], Day [20], Day, Johnson and Sankoff [21], Kou, Markowsky and Berman [50] and many others.

The use of parsimony or phylogenetic number to evaluate hypothetical evolutionary trees can only be justified if the characteristic functions for the group of species are defined in terms of morphological traits that have evolved infrequently. The use of parsimony to infer evolutionary hypotheses from biomolecular sequences has been criticized by Felsenstein [27, 28]. This criticism is certainly justified, because mutations in DNA occur frequently during the evolutionary process. However, when the only data available on a group of species is morphological data, parsimony or phylogenetic number may be useful measures for distinguishing between potential trees.

In Chapter 3, we consider the problem of labelling the internal nodes of a *known topology* to approximate the minimum phylogenetic number for that tree, and extend our results to a fixed-topology problem related to the parsimony problem.

Additive metrics

In Chapter 2 we will show that stochastic models of evolution are related to special metrics with a treelike structure called additive metrics:

Definition 1.1 *A function* $d : S \times S \to \mathbb{R}$ *is an* additive metric *if there is some tree* T *with strictly positive edge weights whose leaves are bijectively labelled by the elements of* S*, and* d *is the distance function on the leaves of* T*.*

For any additive metric, the weighted tree corresponding to this metric is unique and can be constructed in polynomial time (see Buneman [11]). Algorithms for learning evolutionary trees in certain iid models sometimes involve taking a distance function that is not an additive metric and finding an additive metric that is *close* to the distance function (for example, see Farach and Kannan[25]). Closeness may be measured using one of the traditional norms $\ell_1, \ell_2, \ldots, \ell_\infty$, and it has already been shown that the problem of finding the closest additive metric to a distance function is NP-hard for the ℓ_∞ norm (see Agarwala, Bafna, Farach, Paterson and Thorup [2]). Agarwala et al. also gave a 3-approximation algorithm for this problem.

1.3 Learning in the General Markov Model

1.3.1 The Model

In the j-State General Markov Model each species in the evolutionary tree is identified by a sequence over the alphabet $\{0, \ldots, j-1\}$, and the edge processes are all independently and identically distributed mutational processes. From our discussion on page 9, it is obvious that the stochastic process along any edge e can be written as a $j \times j$ matrix M_e of probabilities, where $M_e[i, i']$ is the probability that an i-state at the upper endpoint is an i'-state at the lower endpoint (for any row i of M_e, the sum of the probabilities in row i equals 1). This sort of matrix is called a *stochastic transition matrix*. The sequence at the root is specified by j probabilities $\rho_0, \ldots, \rho_{j-1}$, where ρ_i is the probability that a randomly chosen position of the root sequence has state i.

Definition 1.2 (The General Markov Model) *A* j-State Markov Evolutionary Tree *consists of a tree topology T with n leaves and a distinguished root ρ. The distribution on the state set $\{0, \ldots, j-1\}$ at the root is specified by j parameters $\rho_0, \ldots, \rho_{j-1}$ which sum to* 1. *Every edge is directed away from the root,* *and each directed edge e is labelled by a stochastic transition matrix M_e over the state set $\{0, \ldots, j-1\}$.*

From now on we will use the symbol M to represent an arbitrary Markov Evolutionary Tree (MET). The General Markov Model is almost identical to the model presented in Steel's 1994 paper [58]. Steel enforces two extra constraints (i) $\rho_i > 0$ for every state i; and (ii) $0 < |\det M_e| < 1$ for every edge e.

For every j-State Markov Evolutionary Tree with n leaves, we can define a probabilistic experiment that generates a string from $\{0, \ldots, j-1\}^n$. This experiment is described as a *broadcast* in the paper by Farach and Kannan [25]. Assume some fixed ordering on the leaves of the tree. At the beginning of the experiment, a single state is randomly generated at the root according to the distribution $\rho_0, \ldots, \rho_{j-1}$, and this state is propagated down the edges of the tree towards the leaves. When a state "travels" down an edge, it undergoes a probabilistic transition according to the transition matrix for that edge. On arriving at an internal node, the state is duplicated for each outgoing edge, and the states proceed independently down these edges. The result of the experiment is the ordered concatenation of the states that arrive at the leaves of the tree (this string is one of the "patterns" referred to in the discussion on page 12). Therefore every MET M generates a distribution on $\{0, \ldots, j-1\}^n$. Often we will also use M to denote the distribution generated by the MET M. We will use $\Pr(s)$ to denote the probability that $s \in \{0, \ldots, j-1\}^n$ is generated by a broadcast on M. In situations where we are discussing the relationships between different METs, we will use $\Pr[M](s)$ to indicate the probability of s in the distribution of M.

Now let m be the number of nodes in T and define a fixed ordering on all of the nodes of T. If we let the result of a broadcast on M be the ordered concatenation of the states that arrive at all nodes in the tree, rather than just at the leaves, we obtain a string from $\{0,\ldots,j-1\}^m$. Therefore, a MET also defines a distribution on $\{0,\ldots,j-1\}^m$. When we talk about "the distribution generated by a MET", we will mean the distribution on $\{0,\ldots,j-1\}^n$, unless we state otherwise.

In terms of the motivation for iid models of evolution, our model assumes that the sequence at the root is the concatenation of states from $\{0,\ldots,j-1\}$, where each state is chosen independently and at random, according to the distribution $\rho_0,\ldots,\rho_{j-1}$. Then a single probabilistic experiment on a Markov Evolutionary Tree is equivalent to choosing a random position in the root sequence and concatenating the states that we find at that position for every leaf of the tree. When we are given a set of aligned sequences of length k for each of the leaf species of the tree, this is the same as taking k samples from the distribution generated by the tree. From this point we will forget about the biological motivation for the General Markov Model, and simply assume that we can take samples from a tree.

In this thesis we will be interested in learning in the **Two-State General Markov Model**. We will write the transition matrix for an edge e as

$$M_e = \begin{bmatrix} 1-e_0 & e_0 \\ e_1 & 1-e_1 \end{bmatrix}$$

e_0 is the probability that a 0 state changes to 1 as it travels down e; similarly, e_1 is the probability of observing a flip from a 1-state on e. In Figure 1.3 we show the result of four independent probabilistic experiments (broadcasts) on a particular Two-State MET. The **Cavender-Farris-Neyman model** is the restriction of this model satisfying:

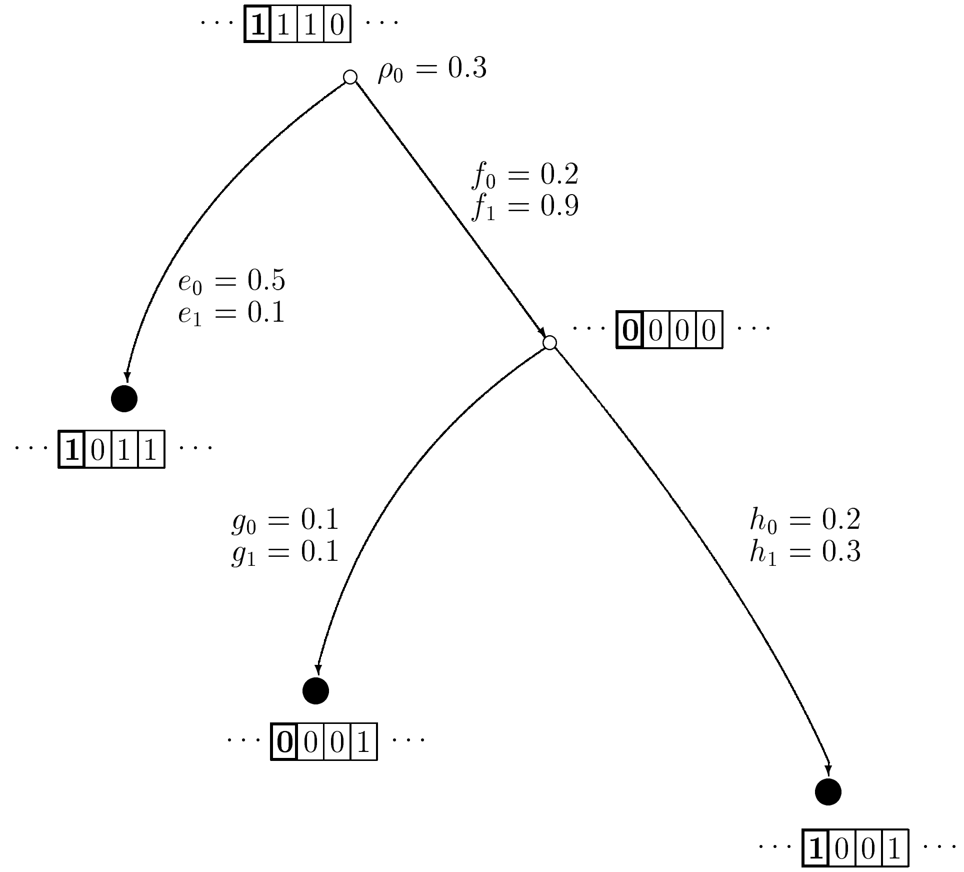


Figure 1.3: The same portion of the sequence is shown at each node, and each separate "column" corresponds to a single probabilistic experiment on the tree. As in Figure 1.2, the experiment corresponding to a single column can be performed by flipping biased coins as the state travels down the edges of the tree. Since there are only two states, we only need to use ordinary two-sided (biased) coins. Assuming wlog that the leaves are ordered from left to right, the picture shows the generation of four independent samples: 101, 000, 100 and 111.

(cfn-i) $e_0 = e_1$ for every edge e;

(cfn-ii) for every e, this common probability, often written as p_e, is less than $1/2$.

The j-State version of the Cavender-Farris-Neyman model is the **Jukes-Cantor model**:

(jc-i) $M_e[i, i']$ has the same value for every pair of states i, i' such that $i \neq i'$;

(jc-ii) for every e, this common probability can be written as $p_e/(j-1)$, for some probability p_e less than $(j-1)/j$.

1.3.2 Learning Problems for Evolutionary Trees

From now on we will assume that we have a source of independent samples from an unknown j-State MET, and we will consider the problem of using these samples to estimate (i) the distribution of the MET; and (ii) the topology of the MET. The problem that interests us most can be informally described as:

> Is there an efficient algorithm that can use a "*reasonable*" number of samples to learn an hypothesis j-State MET M' in polynomial time such that the distribution generated by M' is probably "*approximately correct*" when compared to the distribution of the original j-State MET?

This sort of problem is a computational learning theory problem. In particular, the type of learning algorithm described above is called an efficient PAC-learning algorithm, because of the "Probably Approximately Correct" (PAC) condition. The original definition of PAC-learning was given by Valiant [60], who was interested in learning classes of boolean functions.

PAC-learnability

In this subsubsection we will discuss the origins of PAC-learning, and on page 24 we will discuss PAC-learning for Markov Evolutionary Trees. The books by Anthony and Biggs [5] and Kearns and Vazirani [47] provide more thorough introductions to computational learning theory.

This first person to formally define PAC-learning was Valiant, who was interested in learning some classes of propositional formulas [60] and some restricted classes of first-order formulas [61]. A typical PAC-learning problem (C, H) is described in terms of a *concept class* C and an *hypothesis class* H. The *concept class* is some collection of objects that we are interested in learning, and the *hypothesis class* contains the objects that may be output by a learning algorithm.

Most work that has been carried out on PAC-learning, including Valiant's early research, is concerned with *supervised learning*. When learning is supervised, it is assumed that every concept $c \in C$ is associated with a set of labelled *examples*. The input to a learning algorithm consists of labelled examples of some unknown concept c from the concept class, which is called the *target*. The only assumption that is made about these examples is that they are generated according to *some unknown* distribution D on the examples for the concept c. The output of a learning algorithm is some object h from the hypothesis class, which is called the *hypothesis* or the *estimate*.

To help explain what sort of object can serve as a labelled example, consider the concept class of propositional formulas on n logical variables (as in the first paper by Valiant [60]). A single example of a propositional formula c is some satisfying assignment for c. In this case, all of the examples have the label "yes". If we also gave examples of assignments that do not satisfy c, then we would need to label each example with "yes" or "no".

Informally, an *efficient PAC-learning algorithm* for the learning problem (C, H) is defined as an efficient algorithm that uses a "reasonable" number of examples of any $c \in C$, and a "reasonable" amount of time, to output an hypothesis h that is probably "approximately correct" for c. We will not define "approximately correct" for supervised learning, but only note that the approximate correctness of h for c is measured with respect to the unknown distribution which generated the examples. For formal definitions of supervised PAC-learning, see either of the books by Anthony and Biggs [5] and Kearns and Vazirani [47].

Our problem of learning in the General Markov Model assumes that the input to our learning algorithm is a collection of samples from the distribution generated by some j-State MET M. These samples can be regarded as examples of M, but they are unlabelled examples: although some strings from $\{0, \ldots, j-1\}^n$ may be more likely to be generated by M than other strings, it is quite likely that each string from $\{0, \ldots, j-1\}^n$ will have some non-zero probability of being generated. Also, the hypothesis that we want to construct is not a function or a geometric concept (whose examples are generated according to some unknown distribution); we are interested in learning the actual distribution that generated the samples. This is *unsupervised* learning.

The question of learning discrete distributions has already been considered by Kearns, Mansour, Ron, Rubinfeld, Schapire and Sellie [46]. Kearns et al. were most interested in efficiently learning classes of discrete distributions over the set $\{0, 1\}^n$ which can be *generated* by certain classes of computational circuits. Their definition of PAC-learning was inspired by the original definition of Valiant [60]: a problem is described by a concept class of discrete distributions over $\{0, 1\}^n$ and an hypothesis class. They defined two types of PAC-learning for discrete distributions, *generation* and *evaluation*.

The two definitions of PAC-learning of discrete distributions given by Kearns et al. are characterized by the type of hypothesis class that they use, although both hypothesis classes are defined in terms of simple computational circuits. To explain further, let $p(n)$ and $r(n)$ be polynomial functions of n and consider any simple computational circuit G with n outputs, of size $p(n)$ and with $r(n)$ input bits. If the $r(n)$ inputs to the circuit are uniform random bits, then the circuit generates some distribution on $\{0,1\}^n$, which we will denote by D_G. The problem of *generation* considered by Kearns et al. is the problem of learning with an hypothesis class of these generating circuits, for some polynomials $p(n)$ and $r(n)$. Alternatively, a deterministic computational circuit E which has n input bits and is of size $p(n)$ defines a distribution on $\{0,1\}^n$ if for every $s \in \{0,1\}^n$, the output of E is a bit string representing a rational number in $[0,1]$ and the sum of these outputs over all $s \in \{0,1\}^n$ equals 1. Let this distribution on $\{0,1\}^n$ be denoted by D_E. For the problem of *evaluation*, the hypothesis class is the class of these deterministic computational circuits, for some polynomial $p(n)$.

Before we present the definition of PAC-learning of Kearns et al., it is helpful to discuss how "probably", "approximately correct" and "reasonable" are interpreted. First of all, note that because the input to a learning algorithm consists of samples from a distribution, there is some probability that the samples are a bad representation of the true distribution. For this reason, the input to a PAC-learning algorithm includes a probability δ (with $\delta \in (0,1)$) that is the maximum probability with which the learning algorithm should fail to return a "approximately correct" hypothesis. Also, we do not necessarily expect to learn the *exact* distribution with high probability, so we only require an hypothesis that is approximately correct. Kearns et al. measured the difference between the concept distribution and the hypothesis distribution using the Kullback-

Liebler distance measure: Let D_1 and D_2 be two distributions over the same finite sample space Ω. The Kullback-Liebler distance, or KL-distance, from D_1 to D_2 is defined as

$$\mathrm{KL}(D_1, D_2) = \sum_{s \in \Omega} D_1(s) \log(D_1(s)/D_2(s))$$

The input to the algorithm will include a parameter which is denoted by ϵ (for $\epsilon > 0$), and which represents the maximum value allowed for the KL-distance from the original distribution to the hypothesis distribution. The number of samples that is deemed to be reasonable is some sufficiently large polynomial function of n, $1/\epsilon$ and $1/\delta$. The algorithm should also run in time that is bounded by a polynomial function of n, $1/\epsilon$ and $1/\delta$. It is important to point out that the idea of using varying parameters to control the quality of an hypothesis and the maximum failure probability is originally due to Valiant [60]. In Valiant's original paper, only one parameter was used (it was assumed that $\delta = \epsilon$). However, as more papers on PAC-learning were published, it became common to use separate δ and ϵ parameters, and this is the definition used in the books by Anthony and Biggs [5][1] and Kearns and Vazirani [47].

Let $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ denote some sufficiently large polynomial in n, $1/\epsilon$ and $1/\delta$.

Definition 1.3 (Kearns et al.) *Let D_n be a class of distributions over the set $\{0,1\}^n$. Then D_n is* efficiently PAC-learnable with a generator *if there is an algorithm A such that for any $D \in D_n$, every $\epsilon > 0$ and every $\delta \in (0,1)$, the algorithm runs in time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ *(counting one time step for each sample taken from D) and returns an hypothesis G such that* $\mathrm{KL}(D, D_G) \leq \epsilon$ *with probability at least* $1 - \delta$.

[1]The PAC-learning definitions of Anthony and Biggs [5] only allow the number of examples to depend on $\log(1/\delta)$. See the paper by Haussler et al. [19] for details of why this distinction is usually unimportant.

The definition of Kearns et al. for efficient PAC-learning with an evaluator is obtained by replacing the word "generator" with "evaluator" and the symbol "G" with "E" in the definition above.

PAC-learning Evolutionary Trees

Finally, we will define the PAC-learning problem that we will consider in Chapter 2. We are interested in the PAC-learnability of the class of j-State General Markov Model with n leaves. The results in Chapter 2 hold for the Two-State General Markov Model.

Our learning problem, which we introduced informally on page 19, is similar to the problem of learning discrete distributions with a generator considered by Kearns et al. For our problem, the concept class is the class of j-State METs, and the hypothesis class is also the class of j-State METs. Since j-State METs are just slightly more general than the generating circuits of Kearns et al. (they allow arbitrarily biased coin flips rather than uniform coin flips, and have j states), our problem is essentially a problem of learning with a generator.

One difference between our definition of PAC-learning and the definition of Kearns et al is that we will usually measure "approximate correctness" by the variation distance metric, defined as follows: Let D_1 and D_2 be two distributions over a finite sample space Ω. The variation distance[2] between D_1 and D_2 is defined as

$$\mathrm{V}(D_1, D_2) = \sum_{s \in \Omega} |D_1(s) - D_2(s)|$$

For any space Ω, variation distance forms a metric on distributions over Ω.

Here is the reason that variation distance is a nice distance measure between distributions: Imagine that D_1 and D_2 are two distributions such that

[2] In the paper [15] we used the notation var(. , .) to denote variation distance. I am using V(. , .) here instead because var(. , .) is usually used for variance in probability textbooks.

$\mathrm{V}(D_1, D_2) \leq \varepsilon$ for a small value ε. Suppose that we are given k samples from one of these distributions and that we are asked to guess the origin of the samples. We can analyse the success of any guessing procedure by considering the variation distance between the two k-sample distributions induced by taking k independent samples from D_1 and D_2 respectively. Farach and Kannan [25] showed that whenever two distributions are within variation distance ε, the variation distance between their k-sample distributions is at most $k\varepsilon$. They then show that any procedure that uses k samples to guess between D_1 and D_2 will succeed with probability at most $(1 + k\varepsilon)/2$. To distinguish between D_1 and D_2 with high probability, the number of samples that we must be given depends at least linearly on $1/\mathrm{V}(D_1, D_2)$.

It is easy to construct pairs of METs that generate distributions that are different, but are arbitrarily close in variation distance; an arbitrarily high number of samples are needed to distinguish between such pairs. This is why we do not consider the problem of *exactly* learning the distribution of METs.

Definition 1.4 *Let $j \geq 2$ be fixed. We say that an algorithm efficiently* PAC-learns *the class of j-State METs* in the variation distance measure *if for any n-leaf MET M and any $\epsilon, \delta \in (0, 1)$, the algorithm takes* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ *time (including the time to take samples from the distribution of M) and constructs another j-State MET $\widehat{M}$ such that with probability at least $1 - \delta$,*

$$\mathrm{V}(M, \widehat{M}) \leq \epsilon$$

We also consider the learning of Two-State METs in the KL-distance measure that was used by Kearns et al. By replacing $\mathrm{V}(M, \widehat{M})$ by $\mathrm{KL}(M, \widehat{M})$ in the definition above, we obtain a definition for efficient PAC-learning in the KL-distance measure. It is known that any distribution that can be learned in KL-distance can be also be learned in variation distance. This is because for any two distributions D_1 and D_2 over the same sample space Ω, $\mathrm{V}(D_1, D_2) \leq \sqrt{(2\ln 2)\mathrm{KL}(D_1, D_2)}$. This last result was proven independently by Csiszár [18], Kullback [51] and Kemperman [48], and is presented in the book by Cover and Thomas [13]. In Section 2.6, we show that the converse is true, if the hypothesis class is allowed to be general enough.

Finally, we note that results on learning classes of evolutionary trees have been presented in papers by Farach and Kannan [25] and by Ambainis, Desper, Farach and Kannan [4]. This research will be discussed in detail in Chapter 2, but for now we will simply point out that they also consider the problem of PAC-learning in the variation distance metric in the sense of our definition above.

Note: Another problem that has been studied is the problem of using samples from the distribution of a MET to find, with high probability, the topology of the original tree. However, there are evolutionary trees in the j-State General Markov Model that have *different topologies* but which generate *identical distributions* along the leaves of the tree. This was originally pointed out by Steel [58] and is part of the reason that we have considered the problem of learning the distribution. Research on the problem of learning the topology of a MET, for classes of METs that do allow the topology to be reconstructed from the distribution, will be discussed in Chapter 2. However in Chapter 4 we will see that even when the topology of a Two-State MET can be reconstructed from its distribution, it is not possible to reconstruct the topology using poly$(n, 1/\epsilon, 1/\delta)$ samples. The number of samples needed to reconstruct the topology will also depend on the transition probabilities of the MET.

1.4 Layout of the thesis

Chapter 2 contains our main result, which is a polynomial-time PAC-learning algorithm for the class of Two-State Markov Evolutionary Trees:

Theorem 1.1 *Let δ and ϵ be any positive constants. If our algorithm is given* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ *samples from any Two-State MET M with any n-leaf topology T, then with probability at least $1-\delta$, the Two-State MET $\widehat{M}$ constructed by the algorithm satisfies* $\mathrm{V}(M, \widehat{M}) \leq \epsilon$.

At certain points in Chapter 2 we will state that an equation can be proved by "algebraic manipulation". These equations were verified using Mathematica, though any other package for performing symbolic manipulation would be just as useful.

Chapter 3 contains approximation algorithms for both the fixed-topology phylogenetic number problem and a related fixed-topology problem.

Chapter 2

Learning Two-State Markov Evolutionary Trees

2.1 Previous research

In this section we will survey previous research on learning evolutionary trees. We will also discuss the connection between our work and a problem of Kearns et al. [46]. This section is also important because it contains some definitions that we will use later in the chapter.

2.1.1 The General Idea

All of the previous research on learning either the distribution or the topology of evolutionary trees takes a similar approach to the problem: the algorithms use samples from the distribution to estimate some additive metric on the unknown tree (See Definition 1.1 of Subsection 1.2.2). This subsection is devoted to giving some intuition about why learning in iid models of evolution often involves the estimation of an additive metric. It is possible (with hindsight) to consider previous research on these problems in a common framework. This includes the research of Steel [58], Farach and Kannan [25], Ambainis, Desper, Farach and Kannan [4], Erdös, Steel, Székely and Warnow [23, 24] and Csűrös and Kao [16, 17]. It is important to note that the research in these papers was not carried out with reference to this framework, and that this general discussion has been included in order to make Subsections 2.1.2 and 2.1.3 easier to understand. Subsections 2.1.2 and 2.1.3 contain the details of previous research on learning the distribution and on learning the topology.

Now consider the exact distribution generated on the leaves of a tree in some j-State iid model of evolution. The following definitions will be useful:

Definition 2.1 *For any leaf x,* $\Pr(x = i)$ *is the probability that x is labelled by i in a sample from this distribution. This is an abuse of notation, as we are using x as the name of the random variable as well as the name of the leaf. For any pair of leaves x and y,* $\Pr(xy = ii')$ *is the probability that i labels x and i' labels y.* $\mathrm{F}(x, y)$ *is the* joint distribution matrix *of the labels on x and y (*$\mathrm{F}(x, y)[i, i'] = \Pr(xy = ii')$ *for all pairs of states i and i').*

Remember that the broadcasting process described on page 16 also defines an extended distribution if we concatenate the states at all nodes of the tree. When we write $\Pr(u = i)$ and $\Pr(uv = ii')$ for internal nodes u and v, these probabilities are defined in terms of the extended distribution.

Steel's paper

The first paper to relate additive metrics to the distribution of trees in iid models was the 1994 paper by Steel [58]. Steel's main contribution was to show that the topology of any j-State MET that satisfies restrictions (i) and (ii) from page 16 can be reconstructed from the exact distribution generated by that tree. To show this, Steel first defined a function Λ on the edges of a MET, where $\Lambda(e)$

is defined as

$$\Lambda(e) = \begin{cases} |\det M_e|\sqrt{\prod_{i=0}^{j-1} \Pr(u=i)}, & \text{if } v \text{ is a leaf} \\ |\det M_e|\sqrt{\prod_{i=0}^{j-1} \Pr(u=i)/\prod_{i=0}^{j-1} \Pr(v=i)}, & \text{otherwise.} \end{cases} \tag{2.1}$$

for any $e = (u \to v)$. These multiplicative weights are well-defined because, under assumptions (i) and (ii), every node u has a non-zero probability of being labelled by i, for every state i [58]. The reason that these values are interesting is because

$$|\det \mathrm{F}(x,y)| = \prod_{e \in (x,y)} \Lambda(e), \tag{2.2}$$

where (x, y) is the path from x to y. Also, restrictions (i) and (ii) imply that $0 < |\Lambda(e)| < 1$ for every edge e [58], so $-\ln(\Lambda(e))$ and $-\ln(|\det F(x,y)|)$ are well-defined, and $-\ln(\Lambda(e))$ is strictly positive for every edge e. By Equation 2.2,

$$-\ln(|\det F(x,y)|) = \sum_{e \in (x,y)} -\ln(\Lambda(e)) \tag{2.3}$$

holds for every pair of leaves x and y, so $-\ln(|\det \mathrm{F}(x,y)|)$ forms an additive metric on the leaf set of the MET. Also, because $-\ln(\Lambda(e)) > 0$ for every edge e, the topology of the additive weighted tree that realizes this additive metric is the unrooted topology of the original MET. Therefore, given the exact values of $-\ln(|\det \mathrm{F}(x,y)|)$ for all pairs of leaves x and y, we can reconstruct the original topology in polynomial time by using one of the polynomial-time algorithms that reconstruct the weighted additive tree of an additive metric (see Buneman [11], Waterman et al [66], Bandelt and Dress [6]). Steel did not consider the problem of reconstructing the edge probabilities of the topology from the distribution of the MET.

Other models of evolution

Some models of evolution are simpler than Steel's model, and therefore the additive metric on the leaves is sometimes defined in terms of a simpler function than $-\ln(|\det \mathrm{F}(x, y)|)$. However, the idea is more or less the same. First $\mathrm{W}(e)$ is defined as a function of the transition matrix M_e and the probabilities $\Pr(u = i)$, for every edge $e = (u \to v)$. When the goal is to reconstruct the topology, this function should satisfy $0 < \mathrm{W}(e) < 1$; if the goal is to learn the distribution, $\mathrm{W}(e)$ may equal 1. In either case, there is a function $\mathrm{W}(x, y)$ defined in terms of $\mathrm{F}(x, y)$ such that

$$\mathrm{W}(x, y) = \prod_{e \in (x,y)} \mathrm{W}(e) \tag{2.4}$$

holds for every pair of leaves x and y. Just like Steel's function, $\mathrm{D}(x, y) = -\ln(\mathrm{W}(x, y))$ is well-defined, and forms an additive metric. If we label the edges of the original tree with the $\mathrm{D}(e)$ values, we can see that the unique additive weighted tree that realizes D on its leaves is the tree obtained by contracting any edges with $\mathrm{D}(e) = 0$. If the model of evolution ensures that all edges satisfy $0 < \mathrm{W}(e) < 1$, the topology can be reconstructed from the exact distribution of the tree.

In Subsection 2.1.2, we will see that in some models of evolution that have "one-dimensional" edges (such as the Cavender-Farris-Neyman model defined on page 19, the Jukes-Cantor model (see page 19) and related models (see Subsection 2.1.2), the additive weight on an edge of the tree determines the original parameter for that edge. It is usually the case that edges with $\mathrm{D}(e) = 0$ can be lost without affecting the distribution, and therefore a tree that generates the exact distribution can be constructed from the additive metric.

Inexact data

The problem that interests us is how we can use samples from a tree to PAC-learn the distribution, so we cannot assume that we know the exact values of the $\mathrm{D}(x, y)$ distances. The strategy adopted in previous papers is to take enough samples to obtain close estimates $\widehat{\mathrm{D}}(x, y)$ for the $\mathrm{D}(x, y)$ distances (Farach and Kannan [25], Ambainis et al. [4]), or a substantial set of these distances (Erdös et al. [23, 24], Csűrös and Kao [16, 17]). Then these distances are passed to some algorithm which proceeds to construct the topology (or an approximate topology) from these estimates. Farach and Kannan showed how to use the weights of this approximate topology to learn the distribution, for the Cavender-Farris-Neyman model of evolution. Ambainis et al. then extended this argument to show how to PAC-learn the distribution for "one-dimensional" j-State models of evolution.

It will be useful to distinguish between additive closeness and multiplicative closeness when we are talking about estimates:

Definition 2.2 *An estimate $\widehat{Q}$ of a quantity Q is within* additive error ε *of its true value if $|Q - \widehat{Q}| \leq \varepsilon$. $\widehat{Q}$ is within* multiplicative error ε *of its true value if $|Q - \widehat{Q}| \leq |Q|\varepsilon$.*

All the existing analyses of algorithms for these problems require that the estimates $\widehat{\mathrm{D}}(x, y)$ must lie within some additive error of their true values. For this part of the discussion, we will denote this additive error by ξ. When the goal is to construct the distribution, ξ is related to the desired variation distance ϵ (see Subsection 2.1.2), and when the goal is finding the topology, ξ is related to the edge weights of the tree (see Subsection 2.1.3). In the papers listed on the previous page, the estimation of $\mathrm{D}(x, y)$ has been achieved by first taking a number of samples from the distribution and letting the entries of the matrix $\widehat{\mathrm{F}}(x, y)$ be the observed probabilities of each possible labelling of xy. Then $\widehat{\mathrm{W}}(x, y)$ is defined as the value of the function W when it is applied to the matrix $\widehat{\mathrm{F}}(x, y)$, and $\widehat{\mathrm{D}}(x, y)$ is defined as $-\ln(\widehat{\mathrm{W}}(x, y))$.

Suppose that we want to satisfy

$$\mathrm{D}(x,y)-\xi \quad \leq \quad \widehat{\mathrm{D}}(x,y) \quad \leq \quad \mathrm{D}(x,y)+\xi \tag{2.5}$$

with high probability. We will first show that if condition 2.5 is to hold, then $\mathrm{W}(x,y)(1-2\xi) \leq \widehat{\mathrm{W}}(x,y) \leq \mathrm{W}(x,y)(1+2\xi)$ must also hold. Note that condition 2.5 is equivalent to the condition $\exp[\mathrm{D}(x,y)-\xi] \leq \exp[\widehat{\mathrm{D}}(x,y)] \leq \exp[\mathrm{D}(x,y)+\xi]$. Substituting $-\ln(\mathrm{W}(x,y))$ for $\mathrm{D}(x,y)$ and $-\ln(\widehat{\mathrm{W}}(x,y))$ for $\widehat{\mathrm{D}}(x,y)$, we find that condition 2.5 is equivalent to $\mathrm{W}(x,y)\exp[-\xi] \leq \widehat{\mathrm{W}}(x,y) \leq \mathrm{W}(x,y)\exp[\xi]$. Now assume that $\xi < 1/2$, and write out the Taylor series expansion of $\exp[\xi]$. Then condition 2.5 is equivalent to

$$\mathrm{W}(x,y)(1-\xi+\sum_{i=2}^{\infty}(-\xi)^i/i!) \quad \leq \quad \widehat{\mathrm{W}}(x,y) \quad \leq \quad \mathrm{W}(x,y)(1+\xi+\sum_{i=2}^{\infty}\xi^i/i!)$$

Since $\xi < 1/2$, $\sum_{i=2}^{\infty}(-\xi)^i/i! \geq 0$, and $\sum_{i=2}^{\infty}\xi^i/i! \leq \xi$ both hold. Therefore, if we have an estimate $\widehat{\mathrm{W}}(x,y)$ that satisfies condition 2.5, it also must satisfy $\mathrm{W}(x,y)(1-\xi) \leq \widehat{\mathrm{W}}(x,y) \leq \mathrm{W}(x,y)(1+2\xi)$, and therefore it must satisfy

$$\mathrm{W}(x,y)(1-2\xi) \quad \leq \quad \widehat{\mathrm{W}}(x,y) \quad \leq \quad \mathrm{W}(x,y)(1+2\xi).$$

The point of this rather long-winded argument is that to to estimate $\mathrm{D}(x,y)$ within additive error, we need to estimate $\mathrm{W}(x,y)$ within multiplicative error. This usually means that we need to estimate some entries of $\mathrm{F}(x,y)$ within additive error $O(\xi\mathrm{W}(x,y))$. It is difficult to justify this remark here, because we don't know exactly how $\mathrm{W}(x,y)$ depends on the entries in $\mathrm{F}(x,y)$. For an example, see the Cavender-Farris-Neyman weights defined in Equation 2.7 of Subsection 2.1.2; it is obvious that the remark is true for these weights.

The fact that it may be necessary to estimate random variables to within additive error $O(\xi\mathrm{W}(x,y))$ is important. Suppose X is a binary random variable and that we want to use samples of X to estimate $\Pr(X=1)$ within additive error $c\xi\mathrm{W}(x,y)$ for some constant c, with high probability. Let $\Pr(X=1)=p$ and let Y be another binary random variable such that $\Pr(Y = 1) = p + 2c\xi \mathrm{W}(x,y)$. Then, the number of samples needed to estimate $\Pr(X = 1)$ within additive error $c\xi \mathrm{W}(x,y)$ (with high probability) is at least as big as the number of samples needed to distinguish between X and Y (with high probability). The variation distance between X and Y is $4c\xi \mathrm{W}(x,y)$. Then by the argument of Farach and Kannan described on page 25, we need to take a number of samples proportional to $1/\xi \mathrm{W}(x,y)$, if we are to distinguish between X and Y (or to estimate $\Pr(X = 1)$ within additive error $c\xi \mathrm{W}(x,y)$) with high probability. For this reason, most previous research considers classes of evolutionary trees in which all of the $\mathrm{W}(x,y)$ multiplicative distances (or a substantial subset of them) are bounded from below.

On a positive note, Chernoff Bounds and related results provide upper bounds on the number of samples that suffice to reliably estimate a random variable. The probability $\Pr(xy = ii')$ that a pair of leaves are labelled ii' can be interpreted as a binomial random variable that succeeds with probability $\Pr(xy = ii')$.

Lemma 2.1 (Chernoff Bounds) *Let X be a random variable and suppose that $\Pr(X = 1) = p$ and that $X_1, \ldots, X_k$ are k independent samples of this variable. Let $\widehat{X} = (\sum_{l=1}^{k} X_l)/k$. Then, for any $\varepsilon > 0$,*

$$\Pr[|\widehat{X} - p| \geq \varepsilon] \leq 2\exp[-2\varepsilon^2 k]$$

A proof of this lemma can be found in a paper by McDiarmid [52].

2.1.2 Previous work on learning the distribution

Farach and Kannan [25] and Ambainis, Desper, Farach and Kannan [4] were interested in PAC-learning the distribution of Cavender-Farris-Neyman trees and the distribution of some other models of evolution with “one-dimensional” edges. These were the first results obtained on the PAC-learnability of evolutionary trees.

Cavender-Farris-Neyman trees

For the CFN model, Farach and Kannan defined the multiplicative weights for the edges of a tree by

$$\mathrm{W}(e) =_{def} (1 - 2p_e) \tag{2.6}$$

and showed that Equation 2.4 holds for

$$\mathrm{W}(x, y) =_{def} (1 - 2\,\mathrm{Pr}(x \neq y)). \tag{2.7}$$

By the condition (cfn-ii) for Cavender-Farris-Neyman trees listed on page 19, $0 < (1 - 2p_e) \leq 1$ holds, so $\mathrm{D}(x, y) = -\ln(1 - 2\,\mathrm{Pr}(x \neq y))$ is defined and forms an additive metric.

The main contribution of Farach and Kannan was to give an algorithm that learns Cavender-Farris-Neyman trees in the variation distance metric. Their algorithm motivated our discussion of "Inexact data" in Subsection 2.1.1 to a large extent; therefore, their algorithm fits into the framework described in Subsection 2.1.1. To explain further, suppose $\epsilon > 0$ is the maximum variation distance allowed between the original CFN tree and the hypothesis output by the algorithm. Farach and Kannan take enough samples from a CFN tree to ensure that, with high probability, every $\widehat{\mathrm{D}}(x, y)$ is within additive error $\epsilon/(12n)$ of the true value. Their results are parametrized by a quantity denoted by α, where

$$\alpha =_{def} \min_{x,y}\{(1 - 2\,\mathrm{Pr}(x \neq y))/2\}$$

and the minimum is taken over all leaves x and y of the tree (Condition (cfn-ii) ensures that $\alpha > 0$ always holds). Note that by Equation 2.7, α is the minimum of $W(x, y)/2$, taken over all pairs of leaves. Their result was:

Theorem: (Farach and Kannan [25]) *Given a tree M with n leaves and some $\epsilon > 0$, the algorithm needs only*

$$\frac{(12n)^2 6 \ln n}{\alpha^2 \epsilon^2}$$

samples to construct M' such that $\mathrm{V}(M, M') \leq \epsilon$ with probability at least $1 - 1/n^2$.

The number of samples cited above is simply the number of samples used by Farach and Kannan to guarantee that with probability at least $1 - 1/n^2$, every estimate $\widehat{\mathrm{D}}(x, y)$ lies within additive error $\epsilon/(12n)$ of its true value.

Farach and Kannan described a very elegant algorithm for constructing an hypothesis CFN tree. This algorithm is based on a polynomial-time approximation algorithm of Agarwala, Bafna, Farach, Paterson and Thorup [2] for approximating distance functions by additive metrics. For any input distance function, the algorithm of Agarwala et al. constructs an additive metric d so that the ℓ_∞-distance between the input function and d is at most three times the best possible ℓ_∞-distance for the input. When the tree-fitting algorithm is given the estimate distance function $\widehat{\mathrm{D}}(x, y)$, then with probability at least $1 - 1/n^2$, the additive metric d that is returned satisfies $\ell_\infty(\widehat{\mathrm{D}}, d) \leq \epsilon/(4n)$, and by the triangle inequality, we find that $\ell_\infty(\mathrm{D}, d) \leq \epsilon/(3n)$. This is interesting because it relates the original additive metric to the additive metric returned by the tree-fitting algorithm.

Farach and Kannan then convert the additive tree returned by the Agarwala et al. algorithm into a CFN tree by defining $p'_e = (1 - \exp^{-d(e)})/2$ for every edge e of this new tree. This CFN tree is the hypothesis output by their learning algorithm. The proof that the hypothesis is close to the original tree in variation distance depends on the fact that $\ell_\infty(\mathrm{D}, d) \leq \epsilon/(3n)$. Usually the topology of the hypothesis tree will be quite similar to the topology of the original CFN tree. If e is an edge in the original CFN tree and e' is an edge of the hypothesis, these are considered to be the same edge if and only if the partition of the leaf set induced by e is the same as the partition induced by e'. Farach and Kannan show that if e is an edge in the original tree that does not match an edge in the hypothesis (or vice versa), then the probability p_e (or p'_e) is small, of order $O(\epsilon/n)$. They also show that if the edge e appears in both trees, then $|p_e - p'_e|$ is $O(\epsilon/n)$. These small differences in the topology and the edge parameters do not affect the distribution much, and the total variation distance between the original CFN tree and the hypothesis is at most ϵ (with probability $1 - 1/n^2$). It is straightforward to modify their argument to obtain an hypothesis that is close with probability $1 - \delta$, for varying δ.

Farach and Kannan also provided a information-theoretic lower bound for learning CFN trees. This bound was improved in the paper of Ambainis et al. [4], where it was shown that

Theorem: *Any algorithm needs $\Omega(n/\epsilon^2)$ samples to PAC-learn n-leaf Cavender-Farris-Neyman trees within variation distance ϵ with high probability.*

Ambainis et al. also showed that, for certain classes of Cavender-Farris-Neyman trees, $O(n/(\epsilon\alpha)^2)$ samples suffice to PAC-learn these trees.

Related models of evolution

Ambainis et al. [4] extended the results of Farach and Kannan to some j-State models of evolution which have "one-dimensional" transition matrices. We will explain their results by assuming that in this "one-dimensional" model, there is some parametrized $j \times j$ transition matrix $P(t)$ such that $P(t)$ is defined for every $t > 0$, and such that the transition matrices on the edges of a evolutionary tree are instances of this matrix. Ambainis et al. originally described their j-State model by defining these $P(t)$ matrices in terms of a continuous-time stochastic process acting along the edges of the evolutionary tree. We have omitted the details of these continuous-time processes here, but essentially they make the following assumptions:

1. $P(t_1 + t_2) = P(t_1)P(t_2)$ for every $t_1, t_2 > 0$; $P(0)$ is the identity matrix;
2. There is a *stationary distribution* π on $\{0, \ldots, j-1\}$: $\sum_{i=0}^{j-1} P(t)[i, i']\pi_i = \pi_{i'}$ for every state i' and every $t > 0$;
3. The transition matrices are *time-reversible*: $\pi_i P(t)[i, i'] = \pi_{i'} P(t)[i', i]$ for every pair of states i and i' and every $t > 0$;
4. $|\det P(t)| \neq 0$ for $t \geq 0$;
5. There is some *known* representation for $P(t)$. For any $\widehat{t}$ which lies within additive error $\varepsilon > 0$ of t, every entry of $P(\widehat{t})$ lies within additive error $c\varepsilon$ of the corresponding entry of $P(t)$, for some constant c.

Now consider a j-State Markov Evolutionary Tree such that every edge e is associated with some $t > 0$, and the transition matrix on e is equal to $P(t_e)$ for every edge e. Assume that the distribution on the states at the root of the tree is π. This is what we mean by a "one-dimensional" tree.

Ambainis et al. defined the multiplicative distance between any pair of leaves x and y as $|\det\, P(x \to y)|$. The matrix $P(x \to y)$ is defined by $P(x \to y)[i, i'] = \Pr(y = i' \mid x = i)$, the probability that when $x = i$, $y = i'$ also holds. Because the process is time-reversible,

$$P(x \to y) = \Pr(y \to x) = \prod_{e \in (x,y)} P(t_e)$$

holds. They then showed that $-\ln(|\det P(x \to y)|) = (-\ln|\det P(1)|)(\sum_{e\in(x,y)} t_e)$, and that

$$\mathrm{D}(x, y) =_{def} -\ln(|\det P(x \to y)|)$$

defines an additive metric on the leaves of the evolutionary tree. For this model, α is defined as $\min_{x,y}\{|\det\ P(x \to y)|\}$. Ambainis et al. showed that when the $\mathrm{D}(x, y)$ values are estimated closely, the algorithm of Farach and Kannan could be used to obtain an hypothesis where the edges are labelled with estimates $\widehat{\mathrm{D}}(e)$. They then showed that these can be used to obtain estimates of the original t_e values, and that estimates of the $P(t_e)$ matrices can be calculated so that the variation distance between the original tree and the hypothesis can be shown to be small.

Under these assumptions, Ambainis et al. proved an analogue of the theorem of Farach and Kannan stated on page 36 for their j-State model. They also showed that for certain classes of these one-dimensional j-State evolutionary trees, $O(n/(\epsilon\alpha)^2)$ samples suffice to construct an hypothesis that lies within variation distance ϵ of the original tree.

2.1.3 Previous work on finding the topology

Results by Erdös et al.

Erdös, Steel, Székely and Warnow [23, 24] considered the problem of reconstructing the *topology* of some restricted classes of Cavender-Farris-Neyman trees and j-State Markov Evolutionary Trees. Their aim was to find upper bounds on the number of samples needed to construct the unrooted topology, for classes of binary evolutionary trees (trees which only have nodes of degree 3) whose topology can be obtained from the distribution.

For the problem of learning Cavender-Farris-Neyman trees, the weights along the edges and between pairs of leaves are exactly the same as the weights in Subsection 2.1.2. It is assumed that $(1 - 2p_e)$ is strictly less than 1 for every edge of the tree. This assumption is necessary when we consider the problem of finding the topology: it is easy to show that if e is an edge in a CFN tree such that $p_e = 0$, then the CFN tree obtained by identifying the two endpoints of e generates *exactly* the same distribution as the original tree. It is also assumed that $(1 - 2p_e) \in [a, b]$ for some parameters $0 < a < b < 1$ and for every edge e in the tree, and the results are described in terms of these parameters.

Consider an arbitrary tree from this restricted CFN model. Since the tree is binary, removing any internal edge e from the tree creates four subtrees. Let x_1, x_2, x_3, x_4 be four leaves, one from each subtree, and assume that x_1 and x_2 are on the "same side" of e. Then, because the weights are additive, and because $\mathrm{D}(e) = -\ln(1 - 2p_e) \geq -\ln(b) > 0$,we can show that

$$\mathrm{D}(x_1, x_2) + \mathrm{D}(x_3, x_4) < \mathrm{D}(x_1, x_3) + \mathrm{D}(x_2, x_4) = \mathrm{D}(x_1, x_4) + \mathrm{D}(x_2, x_3)$$

(This is the well-known four point condition of Buneman [11]) Also, there is a difference of at least $-\ln(b)$ between $\mathrm{D}(x_1, x_2) + \mathrm{D}(x_3, x_4)$ and the bigger values. Then, if we could estimate the D function on pairs of leaves from $\{x_1, x_2, x_3, x_4\}$ so that the estimates were within additive error $-\ln b/4$ of their true values, this would allow us to locate the central edge for x_1, x_2, x_3, x_4. However, estimating $\mathrm{D}(x, y)$ within additive error means taking $O(1/(1 - 2\Pr(x \neq y))^2)$ samples, so it makes sense to choose the four leaves for e to make the multiplicative weights as large as possible. The key idea of the algorithm of Erdös et al. is to choose x_1, x_2, x_3, x_4 so that the path from e to each of these leaves has as few edges as possible. A group of four leaves that satisfies this constraint for an edge e is called a *short quartet*. The *depth* of e, denoted by $\mathrm{depth}(e)$ is the number of edges in the longest path from e to a leaf of any short quartet for e. For any restricted CFN tree M with the topology T, they define $\mathrm{depth}(M) = \max_{e \in E(T)} \mathrm{depth}(e)$.

Erdös et al. gave an algorithm that runs in $O(n^4 \log n)$ time [24], and constructs the true topology of an evolutionary tree from a partial set of estimated distances $\widehat{\mathrm{D}}$, as long as this set includes estimates for the distances of the short quartets in the tree. To analyse the number of samples needed to construct the topology, they show that the value of $(1 - 2\Pr(x \neq y))$ for any pair of leaves x and y in a short quartet is at least $a^{2\mathrm{depth}(e)+3}$. Since $\mathrm{depth}(e)$ is never more than $\log(n)$, it is only necessary to estimate $\Pr(x \neq y)$ within multiplicative error for paths that contain about $2\log n$ edges. The main result proved by Erdös et al. [23] was:

Theorem: (Erdös et al. [23, 24]) Under the assumption that $0 < a \leq (1 - 2p_e) \leq b < 1$ for every edge e in a CFN tree M

$$O\left(\frac{\log n}{(1-\sqrt{b})^2 a^{4\mathrm{depth}(M)}}\right)$$

samples suffice to reconstruct the topology of the tree with high probability.

In the worst case, $\mathrm{depth}(M)$ is $\log(n)$ and then the number of samples needed to construct the topology is $O\left(\log n/((1-\sqrt{b})^2 n^{4\log a})\right)$, which is polynomial for fixed a and b. Erdös et al. also gave results on the expected number of samples needed if the tree topology is randomly chosen from the uniform distribution on binary trees or from the Yule-Harding distribution [23, 24].

The research of Erdös et al. is described in terms of the additive metric D, so we could derive the same result for the restricted General Markov Model of Steel [58] under the assumption that $\Lambda(e) \in [a, b]$ for every edge e. Erdös et al. extended their result to the restricted General Markov Model in a slightly different way. In their second paper, they defined the distance between a pair of leaves to be the Steel weight $\mathrm{D}(x, y) = -\ln(|\det \mathrm{F}(x, y)|)$. However, they made the assumption that $|\det M_e| \in [a, b]$, and also that $\prod_{i=0}^{j-1}\Pr(u = i)$ is bounded from below for every node u. They then showed that if their algorithm is given

$$O\left(\frac{\log n}{(1-b)^2 a^{4depth(M)}}\right)$$

samples from any MET M satisfying these constraints, it finds the true topology with high probability.

Results by Csűrös and Kao

Further research on finding the topology in restricted models of evolution was recently presented in a paper by Csűrös and Kao [16] and an unpublished manuscript [17] by the same authors. Csűrös and Kao considered both the restricted Jukes-Cantor model and the restricted General Markov Model. Their approach is similar to that of Erdös et al, because their algorithm only needs accurate estimates for some of the additive $\mathrm{D}(x, y)$ distances in order to construct the topology. Their main contribution was to give an algorithm that constructs the topology in $O(n^2)$ time from these estimates, with high probability.

The first paper of Csűrös and Kao [16] presents the $O(n^2)$ algorithm and describes how to reconstruct trees in the j-State Jukes-Cantor model. In this paper, Csűrös and Kao define the multiplicative distance on an edge as

$$\mathrm{W}(e) = 1 - p_e(j/(j-1))$$

and the multiplicative distance between two leaves as

$$\mathrm{W}(x, y) = 1 - \Pr(x \neq y)(j/(j-1)).$$

Suppose we have two constants $0 < a \leq b < 1$. Csűrös and Kao showed that for any Jukes-Cantor tree M with topology T such that $\mathrm{W}(e) \in [a, b]$ for every edge in T, with probability at least $1 - \delta$, their algorithm reconstructs the topology using

$$O\left(\frac{\log(n/\delta)}{a^{4d(M)+8}(1-b)^2}\right)$$

samples (this function also includes some extra terms related to the distribution of the tree). We will not define $d(M)$ here, but will simply note that it is very closely related to the depth(M) parameter of Erdös et al: for any tree M, $\mathrm{depth}(M) - 1 \leq d(M) \leq \mathrm{depth}(M)$.

Their second manuscript [17] extends the analysis of their algorithm to trees in the j-State General Markov Model that satisfy Steel's two restrictions. They use a new additive metric for j-State Markov Evolutionary Trees. As in Subsection 2.1.2, $P(x \to y)$ is defined by $P(x \to y)[i, i'] = \Pr(y = i' \mid x = i)$. In the General Markov Model, $P(x \to y)$ and $P(y \to x)$ are not necessarily the same matrix. Csűrös and Kao defined the multiplicative weight between a pair of leaves as

$$\mathrm{W}(x, y) =_{def} \sqrt{|(\det P(x \to y))(\det P(y \to x))|}$$

and showed that this is a multiplicative measure on the edges of a tree and that the corresponding edge weights $\mathrm{W}(e)$ lie in $(0, 1)$. They then extended their earlier result to the j-State General Markov Model, assuming that $\mathrm{W}(e) \in [a, b]$ for some $0 < a \leq b < 1$. They also showed that in the General Markov Model

$$|\det \mathrm{F}(x, y)|^2 = \left(\prod_{i=0}^{j-1} \Pr(x = i) \prod_{i=0}^{j-1} \Pr(y = i) \right) |(\det P(x \to y))(\det P(y \to x))|,$$

holds for every pair of leaves x and y. In any j-State model, $\prod_{i=0}^{j-1} \Pr(x = i) \leq (1/j)^j$ for any leaf x. Therefore $\sqrt{|(\det P(x \to y))(\det P(y \to x))|}$ is greater than $|\det \mathrm{F}(x, y)|$. The reason they make this point is because this is an indication that it may be possible to estimate their $\mathrm{W}(x, y)$ function within multiplicative error with fewer samples than are needed to estimate $|\det \mathrm{F}(x, y)|$ (Steels's original function) within multiplicative error.

2.1.4 Relation to Hamming Balls

The problem of PAC-learning the distribution of 2-State Markov Evolutionary Trees (and j-State METs) is also related to a problem from the paper of Kearns, Mansour, Rubinfeld, Ron, Schapire and Sellie [46]. A *Hamming Ball distribution* over $\{0,1\}^n$ is defined in terms of a binary string c of length n called the *centre* and a *corruption probability* p: one sample from this Hamming Ball distribution is obtained by taking the centre c and independently flipping each bit of c with probability p. A *linear mixture* of j Hamming Balls is defined by j Hamming Balls and by j probabilities $\rho_1, \ldots, \rho_j$ that sum to 1. A single sample from the linear mixture is obtained by choosing i from $\{1, \ldots, j\}$ according to the distribution $\{\rho_1, \ldots, \rho_j\}$, and then generating one sample from the ith Hamming Ball distribution. Kearns et al. considered the problem of learning linear mixtures of Hamming Ball distributions.

Kearns et al. gave a PAC-learning algorithm for learning linear mixtures of Hamming Balls, as long as they all have the same corruption probability p.

There is a very natural generalisation of the general problem of learning linear mixtures of Hamming Balls. A *product distribution* over binary strings of length n is defined in terms of n parameters $\lambda_1, \ldots, \lambda_n$: one sample from the product distribution is generated by independently setting the ith bit to 1 with probability λ_i (and otherwise setting it to 0), and concatenating all of these bits together. A linear mixture of product distributions is defined in the obvious way from j probabilities $\rho_1, \ldots, \rho_j$ that sum to 1, and j product distributions.

The problem of learning a linear mixture of j product distributions is a special case of the problem of learning j-State Markov Evolutionary trees. For any linear mixture of j distributions, let $\rho_1, \ldots, \rho_j$ be the j probabilities for choosing between product distributions, and for every i in the state set $\{1, \ldots, j\}$, let $\lambda_{i,1}, \ldots, \lambda_{i,n}$ be the probabilities for the ith product distribution. The linear mixtures distribution can be generated by an n-leaf j-State MET with the star topology whose parameters are defined in the following way: the distribution on the state set $\{0, \ldots, j-1\}$ at the root is specified by $\rho_1, \ldots, \rho_j$ (ρ_i is the probability that the state $i-1$ is generated). For every edge $e_l = (\rho \to l)$, and every $i \in \{0, \ldots, j-1\}$, the transition matrix is:

$$M_e[i, i'] = \begin{cases} 1 - \lambda_{i+1,\, l} & \text{for } i' = 0 \\ \lambda_{i+1,\, l} & \text{for } i' = 1 \\ 0 & \text{for all } 2 \leq i' \leq j-1 \end{cases}$$

This means that our results on learning Two-State METs, described later in this chapter, imply the PAC-learnability of all linear mixtures of two product distributions. Also, the problem of learning j-State METs generalizes the problem of learning linear mixtures of j product distributions.

We also note that very recently the problem of learning linear mixtures of two product distributions was considered independently by Freund and Mansour [30], who showed that a very different algorithm learns linear mixtures of two product distributions from $\text{poly}(n, 1/\epsilon, \log(1/\delta))$ samples. They did not consider the problem of learning mixtures of more than two product distributions.

2.2 Sketch of our algorithm

In this chapter we will present an algorithm that PAC-learns the class of Two-State Markov Evolutionary Trees (METs) in the sense of Definition 1.4. The input to the learning algorithm is parametrized by the number of leaves n of the MET M, by the error tolerance $\epsilon > 0$ and the failure probability $\delta > 0$. Our description of the algorithm will also depend on five related "epsilons":

$$\epsilon_1 = \frac{\epsilon}{4(4n+1)},\ \epsilon_2 = \frac{\epsilon}{8n^2},\ \epsilon_3 = \frac{\epsilon_1}{4n},\ \epsilon_4 = \left(\frac{\epsilon_2}{2}\right)^3 \frac{\epsilon_3}{2^7} \text{ and } \epsilon_5 = \left(\frac{\epsilon_2}{2}\right)^2 \frac{\epsilon_4}{4}$$

The largest of these "epsilons" is ϵ_5, which is $\epsilon^6/(2^{33}n^{11}(4n+1))$.

Two important statistical concepts for the Two-State General Markov Model are correlation and covariance. Correlation is a statistical concept defined in terms of variance and covariance. For any discrete random variable X, the *variance* $\text{var}(X)$ of X is defined as $\mathbf{E}[(X - \mathbf{E}[X])^2]$, where $\mathbf{E}$ denotes the expectation of an expression (see Grimmett and Stirzaker [34]). For any two discrete random variables X and Y, the *covariance* $\text{cov}(X, Y)$ is defined as

$$\text{cov}(X, Y) = \Pr(XY = 11) - \Pr(X = 1)\Pr(Y = 1) \tag{2.8}$$

Then the *correlation* of X and Y is defined as

$$\frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}}$$

To a large extent, our algorithm is described in terms of the *covariances* between pairs of leaves; we will write $\text{cov}(x, y)$ to denote the covariance between the labels generated on leaves x and y. It is easy to show by algebraic manipulation that $\text{cov}(x, y) = \det F(x, y)$ holds in the Two-State General Markov Model. In the rest of this chapter, we will work in the multiplicative domain rather than the additive domain. Our algorithm has three phases.

Partition the set of leaves

Throughout this thesis, we use the word "sample" to mean a single string from $\{0, 1\}^n$ generated by the broadcasting process on the unknown MET. In the first phase of our algorithm, we take a number of samples from the MET M and calculate the observed covariance between every pair of leaves by defining

$$\widehat{\text{cov}}(x, y) = \widehat{\Pr}(xy = 11) - \widehat{\Pr}(x = 1)\widehat{\Pr}(y = 1)$$

where $\widehat{\Pr}(xy = 11)$ is the observed probability that a sample from the distribution labels both x and y with state 1, and $\widehat{\Pr}(x = 1)$ and $\widehat{\Pr}(y = 1)$ have similar meanings (In general, the "hatted" version $\widehat{Q}$ of a quantity Q will be used to denote an estimate constructed from a number of samples from the distribution). We will ensure that we take enough samples from M to ensure that with probability at least $1 - \delta/2$, every estimated covariance is within additive error ϵ_4 of its true value. Chernoff bounds allow us to bound the number of samples needed to achieve this result.

Lemma 2.2 *Given k samples from the distribution of M, where*

$$k \geq \frac{8 \ln(4n^2/\delta)}{\epsilon_4^2}$$

we can estimate the covariances for all pairs of leaves in M such that, with probability at least $1-\delta/2$, every estimated covariance is within additive error ϵ_4.

Proof: If we estimate each of $\Pr(xy = 11)$, $\Pr(x = 1)$ and $\Pr(y = 1)$ within additive error $\epsilon_4/4$ then the estimated covariances will lie within additive error ϵ_4 of their true values. There are $n(n-1)/2$ different $\Pr(xy = 11)$ variables in the tree and n different $\Pr(x = 1)$ variables, so we need to bound at most n^2 variables in total. Let X denote any of these random variables. By Lemma 2.1, if we use k variables to obtain the estimate $\widehat{X}$, then $\widehat{X}$ fails to be within additive error $\epsilon_4/4$ with probability at most $2\exp[-2\epsilon_4^2 k/4^2]$. The value of $2\exp[-2\epsilon_4^2 k/4^2]$ is at most $2\exp[-\ln(4n^2/\delta)]$, which is at most $\delta/(2n^2)$. Summing this error over all variables, the probability that even one of the variables fails to achieve the correct error bound is at most $\delta/2$. $\square$

Before we explain how we partition the leaf set of M, we need to explain the concept of a *leaf connectivity graph* for a MET. Assume that we have an estimate for the covariance between every pair of leaves (like the estimates constructed in Lemma 2.2, for example). A *leaf connectivity graph* for the threshold c (for some $c \in (0, 1/4)$) is a graph whose nodes are the leaves of the MET. A pair of leaves x and y are connected by a "positive" edge if $\widehat{\mathrm{cov}}(x, y) \geq c$ and by a "negative" edge if $\widehat{\mathrm{cov}}(x, y) \leq -c$. If neither of these two conditions hold, there is no edge between x and y. We say that a subset C of the leaves of M is a *related set* if when we ignore the signs on the edges, C forms a connected set in the graph. We say that a subset C of the leaves of M is a *maximal related set* if when we ignore the signs on the edges, C forms a *maximal* connected component in the graph.

In Lemma 2.2 we showed how to construct estimates for the covariances between pairs of leaves of M, such that all these estimates lie within additive error ϵ_4 of their true values with probability at least $1 - \delta/2$. Our algorithm partitions the set of leaves of M by using these estimates to construct a leaf connectivity graph for the threshold $\epsilon_2/2$. Then the maximal related sets of this graph form a partition of the set of leaves of M, and each maximal related set of leaves induces a subMET of M. In Section 2.6, we will show that we can closely approximate the distribution of the original Two-State MET M by approximating the distribution of each of these maximal related sets closely, and joining these subMETs by "cut edges" or "product edges".

Approximate the correct topology of each maximal related set

For every maximal related set C, the induced topology on the set of leaves in C (with any degree 2 nodes contracted) is denoted by $T(C)$. For each maximal related set, we construct an approximation to this induced topology. It is not possible to guarantee to construct the *exact* topology (Steel [58] has shown that if $\Lambda(e) = 1$ for one of the Λ-edge weights defined in Subsection 2.1.1, then the location of e cannot be determined from the distribution of the MET). In Section 2.4, we present a polynomial-time algorithm that constructs an *approximate* topology using the estimates of the covariances between pairs of leaves in C.

Definition 2.3 *Let $d \in (0, 1/2)$ and let T be the topology of a Two-State MET. We say the topology $\widehat{T}$ is a d-*contraction *of T if $\widehat{T}$ can be obtained from T by contracting some internal edges of T, and if every edge e that is contracted in $\widehat{T}$ satisfies $\Lambda(e) \geq 1 - d$.*

In Section 2.4, we will prove that if our algorithm is given estimates of inter-leaf covariances for C that lie within additive error ϵ_4 of their true values, and C is a maximal related set for the threshold $(\epsilon_2/2)$, then the algorithm constructs an ϵ_3-contraction $\widehat{T}(C)$ of $T(C)$. The construction of the topology is carried out by looking at triples of leaves. In one sense, our approach is similar to previous algorithms for estimating the topology of stochastic evolutionary trees: when we use tests to determine an approximate location for a leaf, we usually need to have estimates of the covariances that lie within *multiplicative error* of their true values. However, we are able to show that we only need to perform tests on triples of leaves with inter-leaf covariances whose absolute values are $\Omega((\epsilon_2)^3)$, and for these covariances, our estimates do lie within multiplicative error of their true values.

Find an approximate Two-State MET for each maximal related set

Once we have obtained an ϵ_3-contraction of $T(C)$ for each maximal related set C, we then construct a Two-State MET $\widehat{M}(C)$ on the topology $\widehat{T}(C)$ such that the distribution generated by $\widehat{M}(C)$ is close to $M(C)$. In Section 2.3, we will show that for any Two-State MET M, there is at least one alternative Two-State MET M' that generates the same distribution as M. However, we will also show that when we make certain assumptions about the type of labelling that we will construct, there is only one labelling that satisfies these conditions and generates the same distribution as the original MET. We also derive quadratic equations which allow us to recover the transition probabilities and the root probability of this labelling, using the exact distribution on triples of leaves.

In Section 2.5 we show that if we are given estimates for the distribution on every three leaves x, y and z in $M(C)$, such that each estimated probability $\widehat{\Pr}(xyz = i_1 i_2 i_3)$ (for i_1, i_2 and i_3 from $\{0,1\}$) satisfies

$$\widehat{\Pr}(xyz = i_1 i_2 i_3) \in \left[\Pr(xyz = i_1 i_2 i_3) - \frac{\epsilon_5}{32}, \Pr(xyz = i_1 i_2 i_3) + \frac{\epsilon_5}{32}\right]$$

then we can obtain a labelling $\widehat{M}(C)$ on $\widehat{T}(C)$ such that for *some* MET $M'(C)$ that generates the distribution of $M(C)$, every parameter of $\widehat{M}(C)$ is within additive error ϵ_1 of its value in $M'(C)$.

There are $n(n-1)(n-2)/6$ different triples x, y, z in any tree with n leaves, and there are 8 different values for $i_1 i_2 i_3$, so therefore we have at most $2n^3$ variables in total. The argument of Lemma 2.2 can be adapted to show that if we take

$$k \quad \geq \quad \frac{(32)^2 \ln(8n^3/\delta)}{2\epsilon_5^2}$$

samples from the distribution of the original MET M, then with probability at least $(1 - \delta/2)$, all of our estimates will lie within additive error $\epsilon_5/32$ of their real values.

Putting it together

The hypothesis returned by the algorithm is a Two-State MET $\widehat{M}$ that generates the product of the $\widehat{M}(C)$ subMETs. This product is defined in a similar way to the product distributions of Subsection 2.1.4, but the components of this product are Two-State METs rather than binary random variables. In

Section 2.6, we show that if the covariance estimates given to the topology construction algorithm are accurate to within additive error ϵ_4, and if each approximate MET $\widehat{M}(C)$ has parameters within additive error ϵ_1 of some labelling that generates the original MET on C, then the product of the $\widehat{M}(C)$ METs lies within variation distance ϵ of the original MET. Thus, we prove Theorem 1.1.

The next section of this chapter gives details about the exact distribution of a Two-State MET. Section 2.4 gives an algorithm that constructs a ϵ_3-contraction of the topology of a maximal related set (or even just a related set), given covariance estimates within additive error ϵ_4 of their true values. In fact, we prove a more general result, which is stated at the beginning of Section 2.4. Section 2.5 shows that if we are given estimates of the distribution for every triplet in C, and these estimates are as close as described above, we can construct $\widehat{M}(C)$ that is close to some MET that generates $M(C)$. Finally, Section 2.6 proves Theorem 1.1.

2.3 Some results about the exact distribution

2.3.1 Basic details

In this section we present some useful equations and results about the exact distribution of a Two-State Markov Evolutionary Tree. Throughout the thesis, we refer to a hypothetical Two-State MET as M, we denote the topology of the tree as T, and the number of leaves of the tree by n. When it is clear which MET is being considered, we will use $\Pr(x = 1)$ to denote the probability of the event that leaf x is labelled by 1, and define the probability of other events in a similar way. In situations where we need to distinguish between distributions, we will use $\Pr_{[M]}$ to indicate that we are talking about the distribution of M.

Steel has already defined a multiplicative weighting for Two-State METs that satisfy the extra conditions (i) $\rho_0 \in (0,1)$; and (ii) For every edge e, $|\det M_e| \in (0,1)$. We extend this weighting to the edges of any Two-State MET by defining

$$\Lambda(e) = \begin{cases} |1-e_0-e_1|\sqrt{\Pr(u=0)\Pr(u=1)}, & \\ \qquad \text{if } v \text{ is a leaf} & \\ |1-e_0-e_1|\sqrt{(\Pr(u=0)\Pr(u=1))/(\Pr(v=0)\Pr(v=1))}, & \\ \qquad \text{if } v \text{ is not a leaf, and } \Pr(v=0)\in(0,1). & \\ 0, & \\ \qquad \text{otherwise.} & \end{cases} \tag{2.9}$$

Note that $\det M_e = (1-e_0-e_1)$ for any edge e of a Two-State MET. It is easy to check that for any edge e for which $\Pr(u=0), \Pr(v=0) \in (0,1)$ holds (which is guaranteed in Steel's model, but not in the Two-State General Markov Model), this definition gives the same value for $\Lambda(e)$ as Steel's original weights. In Lemma 2.5 we will show that Equation 2.2, which states that

$$|\mathrm{cov}(x,y)| = \prod_{e\in(x,y)} \Lambda(e),$$

also holds when $\Lambda(e)$ is defined by Equation 2.9. We need a couple of observations first:

Observation 2.3 *If $e=(u\to v)$ is an edge such that $\Lambda(e)=0$, then the joint distribution on its two endpoints is a product distribution. Also, whenever $\Lambda(e)=0$, either $\Pr(u=0)\in\{0,1\}$ or $1-e_0-e_1=0$ holds.*

Proof: We show the second part first. Suppose that $\Lambda(e)$ equals 0 but that $\Pr(u=0)\in(0,1)$. We know that $\Pr(v=0)=\Pr(u=0)(1-e_0)+\Pr(u=1)e_1$, so under our assumptions, $\Pr(v=0)=0$ if and only if $e_0=1$ and $e_1=0$; then $1-e_0-e_1=0$. The same result holds if $\Pr(v=1)=0$.

It is easy to see that the joint distribution on u and v is a product distribution when $\Pr(u=0)=0$ or $\Pr(u=1)=0$ holds. Otherwise, we know that $1 - e_0 - e_1 = 0$, and the distribution on the nodes is a product distribution where $\Pr(u = 1)$ is the probability that node u is labelled 1 and e_0 is the probability that node v is labelled 1. $\square$

Observation 2.4 *Let M be a Two-State MET with topology T and let v be a degree-2 node whose adjacent edges are $e = (u \to v)$ and $f = (v \to w)$. Define T' from T by replacing the path from u to w with a single edge g, and define M' on this topology by setting $M_g = M_e * M_f$, and labelling all the other edges with their original transition matrices. Then M' generates the same distribution as M. Note that $g_0 = f_0 + e_0(1 - f_0 - f_1)$ and $g_1 = f_1 + e_1(1 - f_0 - f_1)$.*

Now we will show that the extended $\Lambda(e)$ weights multiply to give the covariance between a pair of leaves. Consider the joint distribution on two leaves x and y of a MET, and suppose that u is the node on the path (x, y) that is closest to the root ρ. We will let x_0 and x_1 denote the transition probabilities for the path from u to x (regarding $(u \to x)$ as a single edge) and y_0 and y_1 be the probabilities for the path from u to y.

Lemma 2.5 *Let x and y be two leaves of a Two-State Markov Evolutionary Tree Then the two following equalities hold*

$$\mathrm{cov}(x, y) = \Pr(u = 0)\Pr(u = 1)(1 - x_0 - x_1)(1 - y_0 - y_1) \quad (2.10)$$

$$|\mathrm{cov}(x, y)| = \prod_{e \in (x,y)} \Lambda(e) \quad (2.11)$$

where $\Lambda(e)$ is defined by Equation 2.9.

Proof: Equation 2.10 can be proved by algebraic manipulation of Equation 2.8. Then, if every node v on the path between x and y satisfies $\Pr(v = 0) \in (0, 1)$, Equation 2.11 is obvious. If some nodes on the path have $\Pr(v = 0)\Pr(v = 1) = 0$, applying Observation 2.3 inductively shows that either $\Pr(u =$

0) $\Pr(u = 1) = 0$ or some edge e on the path satisfies $1 - e_0 - e_1 = 0$. In the first case, Equation 2.10 shows that $\mathrm{cov}(x, y) = 0$. Otherwise, suppose wlog that f is an edge on the path $(u \to x)$ such that $1 - f_0 - f_1 = 0$. This is the value of $\det M_f$, and $(1 - x_0 - x_1)$ is the determinant of $\prod_{e \in (u \to x)} M_e$. By the multiplicative property of determinants, $(1 - x_0 - x_1)$ must equal 0, and therefore $|\mathrm{cov}(x, y)| = 0$. $\square$

Another interesting thing about the Two-State model is that whenever e is an internal edge and $\Lambda(e) \neq 0$, $\Lambda(e)$ is the absolute value of the *correlation* of the random variables for the endpoints of e. It is easy to check by algebraic manipulation that for any edge $e = (u \to v)$ in the tree,

$$\mathrm{cov}(u, v) = \Pr(u = 0) \Pr(u = 1)(1 - e_0 - e_1) \tag{2.12}$$

and for any node u in the tree, $\mathrm{var}(u) = \Pr(u = 0) \Pr(u = 1)$. Assuming that the probabilities for u and v both lie in $(0, 1)$, the absolute value of the correlation of u and v is $\Lambda(e)$.

Observation 2.6 *Let $e = (u \to v)$ be an edge in T. Then $\Lambda(e) \leq 1$, and if e is a leaf edge, $\Lambda(e) \leq 1/2$.*

Proof: When e is an internal edge and $\Lambda(e) \neq 0$, $\Lambda(e)$ is the absolute value of the correlation of u and v. It is well-known (see [34] for a proof) that the absolute value of the correlation of two random variables is at most 1. For a leaf edge, $\Lambda(e) = \sqrt{\Pr(u = 1) \Pr(u = 0)}|1 - e_0 - e_1|$, which is at most 1/2. $\square$

In Section 2.5 it will be important to relate the values of $\Lambda(e)$ and $|1 - e_0 - e_1|$ when we know that $\Lambda(e)$ is non-zero. The following Lemma will be useful.

Lemma 2.7 *Let e be an internal edge from u to v and suppose that $\Lambda(e) \geq 1 - d$, for some $0 \leq d < 1/2$. Suppose that $1 - e_0 - e_1 \geq 0$. Then $|1 - e_0 - e_1| \geq 1 - 2d$.*

Proof: Since $\Lambda(e) \neq 0$, it is obvious from the definition of $\Lambda(e)$ (see Equation 2.9) that $1 - e_0 - e_1 \neq 0$, $\Pr(u = 0) \in (0,1)$ and $\Pr(v = 0) \in (0,1)$. Algebraic manipulation shows that

$$\begin{aligned}\Pr(u = 1) &= \frac{\Pr(v = 1) - e_0}{1 - e_0 - e_1}\\ \Pr(u = 0) &= \frac{\Pr(v = 0) - e_1}{1 - e_0 - e_1}.\end{aligned}$$

Then by Equation 2.9,

$$\Lambda(e)^2 = \left(1 - \frac{e_0}{\Pr(v = 1)}\right)\left(1 - \frac{e_1}{\Pr(v = 0)}\right).$$

Then, since $1 - e_0 - e_1 > 0$, we have $\Pr(v = 1) > e_0$ and $\Pr(v = 0) > e_1$. Because $(1 - d)^2 \geq (1 - 2d)$, therefore $\Lambda(e)^2 \geq (1 - 2d)$. Then $\left(1 - \frac{e_0}{\Pr(v=1)}\right) \geq (1 - 2d)$ and $\left(1 - \frac{e_1}{\Pr(v=0)}\right) \geq (1 - 2d)$ hold. Hence $e_0 \leq 2d \Pr(v = 1)$ and $e_1 \leq 2d \Pr(v = 0)$ and therefore $|1 - e_0 - e_1| \geq 1 - 2d$. $\square$

2.3.2 Alternative METs

In the Two-State General Markov Model, there are different METs that give rise to exactly the same distribution on their leaves. In this section we present some results about these alternate METs. We will begin by making an observation that seems obvious, but which will be very useful in this subsection.

Observation 2.8 *Let M be a Two-State MET and u be any internal node of the topology of M. For any string $s \in \{0,1\}^n$, let s_1 denote the portion of the string on the leaves that lie in the subtree rooted at u, and let s_2 denote the portion of the string on the other leaves. Let $\Pr(s_1 \mid u = 0)$ and $\Pr(s_1 \mid u = 1)$ denote the conditional probabilities for s_1. Then*

$$\begin{aligned}\Pr(s) &= \Pr(u = 0)\Pr(s_1 \mid u = 0)\Pr(s_2 \mid u = 0) +\\ &\quad\ \Pr(u = 1)\Pr(s_1 \mid u = 1)\Pr(s_2 \mid u = 0)\end{aligned}$$

holds for any string $s \in \{0,1\}^n$.

Proof: Let $s \in \{0,1\}^n$, and let s_1 and s_2 be defined as described above. We will use $\Pr(s_1)$ to denote the probability that M generates s_1 on the leaves in the subtree rooted at u and $\Pr(s_2)$ to denote the probability that M generates s_2 on the other leaves. Finally, $\Pr(s_2 0)$ will denote the probability that a single broadcast generates s_2 on the appropriate leaves and 0 on the node u, and $\Pr(s_2 1)$ is defined similarly. Then

$$\Pr(s) \quad = \quad \Pr(s_2 0)\Pr(s_1 \mid u = 0) + \Pr(s_2 1)\Pr(s_1 \mid u = 1)$$

Also, $\Pr(s_2 0) = \Pr(u = 0)\Pr(s_2 \mid u = 0)$ and $\Pr(s_2 1) = Pr(u = 1)\Pr(s_2 \mid u = 1)$. Using these equations to substitute for $\Pr(s_2 0)$ and $\Pr(s_2 1)$, we obtain our result. $\square$

Observation 2.9 *Let M be a Two-State MET with topology T and consider any internal edge $e = (\rho \to v)$ from the root ρ. Let T' be a tree that is rooted at v and has the same* unrooted *topology as T. Then there is a Two-State MET M' defined on T' that generates the same distribution as the original MET. This is realised by letting all the edges except $e' = (v \to \rho)$ have their original transition probabilities, defining the distribution at the new root in terms of* $\Pr[M](v = 1)$, *and defining e'_0 and e'_1 as follows:*

If $\Pr[M](v = 1) = 0$, *define* $e'_0 = \rho_1$;

If $\Pr[M](v = 1) = 1$, *define* $e'_1 = \rho_0$;

Otherwise, set $e'_0 = \rho_1 e_1 / \Pr[M](v = 0)$ *and* $e'_1 = \rho_0 e_0 / \Pr[M](v = 1)$.

Corollary 2.10 *For any Two-State MET M and any internal node v of the topology of M, there is some MET whose topology is rooted at v and which generates the same distribution as M.*

Observation 2.11 *If MET M has the rooted topology T then there is a MET M' with the same rooted topology that generates the same distribution as M, such that every internal edge e satisfies $e_0 + e_1 \leq 1$.*

Proof of Observation 2.11: We say that the edge e is "good" if $e_0 + e_1 \leq 1$. Suppose that $e = (u \to v)$ is a non-good internal edge and that all the edges on the path from the root ρ to u are good edges. A relabelling of e and the outgoing edges from v that makes e good and that preserves the distribution generated by the MET can be defined as follows: define $e'_0 = 1 - e_0$ and $e'_1 = 1 - e_1$ and for every outgoing edge $f = (v \to w)$, set $f'_0 = 1 - f_1$ and $f'_1 = 1 - f_0$. It is easy to see that this labelling makes e a good edge, and therefore all the nodes on the path from ρ to v are good edges. The effect of the relabelling is to swap the columns of M_e, and interchange the meaning of "0" and "1" at v. Relabelling the outgoing edges swaps the rows of $M_{(v \to w)}$. If we think about the outgoing edges as defining two product distributions (as explained in Subsection 2.1.4), the effect of relabelling these edges is to ensure that each of the product distributions is chosen with the correct probability.

Formally, we will denote the new MET obtained by changing the values on e by M''. We need to show that M and M'' generate the same distribution. By Observation 2.8, we know that if s is any string in $\{0,1\}^n$ such that s_1 labels the leaves beneath u and s_2 labels the other leaves, then

$$\begin{aligned} \Pr[M](s) &= \Pr[M](u=0)\Pr[M](s_1 \mid u=0)\Pr[M](s_2 \mid u=0) + \\ &\quad \Pr[M](u=1)\Pr[M](s_1 \mid u=1)\Pr[M](s_2 \mid u=0) \end{aligned}$$

and

$$\begin{aligned} \Pr[M''](s) &= \Pr[M''](u=0)\Pr[M''](s_1 \mid u=0)\Pr[M''](s_2 \mid u=0) + \\ &\quad \Pr[M''](u=1)\Pr[M''](s_1 \mid u=1)\Pr[M''](s_2 \mid u=0) \end{aligned}$$

Therefore, to show that M and M'' generate the same distribution, we need to show (a) that $\Pr[M](u = 0) = \Pr[M''](u = 0)$; (b) that $\Pr[M](s_2 \mid u = 0) = \Pr[M''](s_2 \mid u = 0)$ and $\Pr[M](s_2 \mid u = 1) = \Pr[M''](s_2 \mid u = 1)$ hold for every labelling s_2 of the leaves outside the subtree rooted at u; (c) that $\Pr[M](s_1 \mid u = 0) = \Pr[M''](s_1 \mid u = 0)$ and $\Pr[M](s_1 \mid u = 1) = \Pr[M''](s_1 \mid u = 1)$ hold for every labelling s_1 of the leaves in the subtree rooted at u. Since M and M'' are identical except for some edge probabilities in the subtree rooted at u, therefore (a) and (b) automatically hold.

To show (c), first note that for any string s_1 that labels the leaves in the subtree rooted at u, we can divide s_1 into a collection of substrings as follows: let $s_1[v]$ be the portion of s_1 that lies on the leaves in the subtree rooted at v, and for every other child $v' \neq v$ of u, let $s_1[v']$ be the portion of s_1 that lies on the leaves in the subtree rooted at v'. Now

$$\Pr[M](s_1|u = 0) = \Pr[M](s_1[v]|u = 0) \prod_{v' \neq v} \Pr[M](s_1[v']|u = 0)$$

and also $\Pr[M''](s_1|u = 0) = \Pr[M''](s_1[v]|u = 0) \prod_{v' \neq v} \Pr[M''](s_1[v']|u = 0)$. We can write down similar equations for $\Pr[M](s_1|u = 1)$ and $\Pr[M''](s_1|u = 1)$ by changing $u = 0$ to $u = 1$ throughout these equations. Since the transition probabilities are the same in M and M'' except along $e = (u \to v)$ and in the subtree rooted at v, therefore $\Pr[M](s_1[v']|u = 0) = \Pr[M''](s_1[v']|u = 0)$ and $\Pr[M](s_1[v']|u = 1) = \Pr[M''](s_1[v']|u = 1)$ for every $v' \neq v$. So we only need to show that $\Pr[M](s_1[v]|u = 0) = \Pr[M''](s_1[v]|u = 0)$ and $\Pr[M](s_1[v]|u = 1) = \Pr[M''](s_1[v]|u = 1)$ for every labelling $s_1[v]$ of the leaves in the subtree rooted at v. In fact, we will simply show that for any binary labelling of the child nodes of v that is represented by the function ℓ, that $\Pr[M](\ell|u = 0) = \Pr[M''](\ell|u = 0)$ and $\Pr[M](\ell|u = 1) = \Pr[M''](\ell|u = 1)$. Since we have not changed any of the labels on the edges below the child nodes of v, this is enough to ensure that $\Pr[M](s_1[v]|u = 0) = \Pr[M''](s_1[v]|u = 0)$ and $\Pr[M](s_1[v]|u = 1) = \Pr[M''](s_1[v]|u = 1)$ for every labelling $s_1[v]$ of the leaves below v.

For any child w of the node v, let $M_{(v\to w)}$ denote the transition matrix for the edge $(v \to w)$. Then $\Pr[M](\ell \mid u = 0)$ is

$$e_0 * \prod_w M_{(v\to w)}[1, \ell(w)] + (1 - e_0) * \prod_w M_{(v\to w)}[0, \ell(w)]$$

The probability $\Pr[M''](\ell \mid u = 0)$ is

$$e'_0 * \prod_w M''_{(v\to w)}[1, \ell(w)] + (1 - e'_0) * \prod_w M''_{(v\to w)}[0, \ell(w)]$$

However, $e'_0 = (1 - e_0)$. Also, our redefinition of the transition probabilities for the outgoing edges has the effect of exchanging the *rows* of the matrix $M_{(v\to w)}$; therefore $M''_{(v\to w)}[0, i] = M_{(v\to w)}[1, i]$ holds and $\Pr[M](\ell \mid u = 0) = \Pr[M''](\ell \mid u = 0)$. In the same way, we can show that $\Pr[M](\ell \mid u = 1) = \Pr[M''](\ell \mid u = 1)$.

To make all of the internal nodes good, we carry out the following inductive procedure on the MET. If $e = (u \to v)$ is a non-good edge and every edge on the path from the root ρ to u is good, relabel e and v's outgoing edges to make e good. This procedure does not change any of the edges above e, so the rule can be applied inductively to find an MET M' with only good internal edges. $\square$

Observation 2.12 *The multiplicative weight $\Lambda(e)$ of any edge in a Two-State MET is unchanged by either the re-rooting process described in Observation 2.9 or the transformation to "good" edges described in Observation 2.11*

Proof: First consider the re-rooting of an MET described in Observation 2.9, and let M denote the original MET with root ρ and M' denote the new MET with root v. First suppose that $\Lambda(e) \neq 0$ in the original MET M. Then, by Equation 2.9, we know that $\rho_0 \in (0, 1)$, that $e_0 + e_1 \neq 1$ and that $\Pr[M](v = 0) \in$

$(0, 1)$. We will show that

$$\sqrt{\frac{\Pr[M](\rho=0)\Pr[M](\rho=1)}{\Pr[M](v=0)\Pr[M](v=1)}}(1-e_0-e_1)$$
$$= \sqrt{\frac{\Pr[M'](v=0)\Pr[M'](v=1)}{\Pr[M'](\rho=0)\Pr[M'](\rho=1)}}\left(1-\frac{\rho_1 e_1}{\Pr[M](v=0)}-\frac{\rho_0 e_0}{\Pr[M](v=1)}\right)$$

First note that by definition of $\Pr[M'](v=0)$ and e_0' and e_1', that $\Pr[M'](v=0) = \Pr[M](v=0)$ and $\Pr[M'](\rho=0) = \Pr[M](\rho=0) = \rho_0$. Therefore, from now on, we omit the $[M]$ or $[M']$ from our equations. Multiplying both sides of the equation above by $\sqrt{\Pr(v=0)\Pr(v=1)\Pr(\rho=0)\Pr(\rho=1)}$, our goal is to show that

$$\rho_0\rho_1(1-e_0-e_1) = \Pr(v=0)\Pr(v=1)\left(1-\frac{\rho_1 e_1}{\Pr(v=0)}-\frac{\rho_0 e_0}{\Pr(v=1)}\right)$$

If we multiply $\Pr(v=0)\Pr(v=1)$ into the right-hand side, we get

$$\begin{aligned}
\Pr(v=0)\Pr(v=1)-\rho_1 e_1\Pr(v=1)-\rho_0 e_0\Pr(v=0) &= \\
(\rho_0(1-e_0)+\rho_1 e_1)\Pr(v=1)-\rho_1 e_1\Pr(v=1)-\rho_0 e_0\Pr(v=0) &= \\
\rho_0(1-e_0)\Pr(v=1)-\rho_0 e_0\Pr(v=0) &= \\
\rho_0\Pr(v=1)-\rho_0 e_0 &= \\
\rho_0(\rho_0 e_0+\rho_1(1-e_1))-\rho_0 e_0 &= \\
\rho_0\rho_1(1-e_1)-\rho_0(1-\rho_0)e_0 &
\end{aligned}$$

Then because $1-\rho_0 = \rho_1$, we have $\rho_0\rho_1(1-e_0-e_1)$, as required. So $\Lambda(e)$ has the same value in M and M'. Alternatively, suppose that $\Lambda(e) = 0$ in the original MET M. Then, by Observation 2.3, either $\Pr(\rho=0) \in \{0,1\}$ or $e_0+e_1 = 1$. If $e_0+e_1 = 1$, then $\Pr(v=0) = e_1$ and $\Pr(v=1) = e_0$. Then $e_0' = \rho_1$ and $e_1' = \rho_0$ and therefore $e_0'+e_1' = 1$. Then $\Lambda(e) = 0$ in M'. Otherwise $\Pr(\rho=0) \in \{0,1\}$. Suppose $\rho_0 = 1$. Then $\Pr(v=1) = e_0$. Also, since $e_0' = \rho_1 e_1/\Pr(v=0)$, we have $e_0' = 0$, and since $e_1' = \rho_0 e_0/\Pr(v=1)$, we have $e_1' = \rho_0 = 1$. Therefore $e_0'+e_1' = 1$ and we have $\Lambda(e) = 0$.

The only edges affected by the transformation described in Observation 2.11 are $e = (u \to v)$ and the outgoing edges from v. In Observation 2.11 it was shown that this transformation preserves the joint distribution on the nodes adjacent to v. Since the only edges whose transition probabilities are altered are edges adjacent to v, these are the only edges that need to be checked. It is easy to check that $|1 - e'_0 - e'_1| = |1 - e_0 - e_1|$, and that $|1 - f'_0 - f'_1| = |1 - f_0 - f_1|$ for every outgoing edge $f = (v \to w)$. Also, the new value for $\Pr(v = 0)$ is equal to the original value of $\Pr(v = 1)$, so $\Pr(v = 0)\Pr(v = 1)$ is unchanged. Therefore $\Lambda(e)$ has the same value in the new MET. To show that $\Lambda(f)$ is unchanged, the only thing left to check if that $\Pr(w = 0)\Pr(w = 1)$ is preserved. This holds because we have already shown that the joint distribution on the adjacent nodes to v is preserved. $\square$

Observation 2.13 *Let $e = (u \to v)$ be an internal edge of T. If $\Lambda(e) = 1$ and $1 - e_0 - e_1 > 0$, let T' be the tree obtained by contracting the edge e and identifying u and v. Define M' on the topology T' by letting all the edges have their original transition probabilities. Then M' generates the same distribution as M.*

Proof: By Lemma 2.7, we have $1 - e_0 - e_1 = 1$, so $e_0 = 0$ and $e_1 = 0$. Therefore the process on e is the identity process, and identifying the two endpoints of e does not affect the distribution. $\square$

These observations can be used to show that for any Two-State MET M, there is some MET that generates the same distribution, but which has "good" internal edges, and which has no internal edges with $e_0 + e_1 = 1$. Observation 2.9 implies that we can think of METs as being unrooted. We now show that for any internal edge with $\Lambda(e) = 0$, the distribution generated is a product distribution of the two sub-METs obtained by disconnecting the edge e:

Observation 2.14 *Suppose that M is a MET and that $e = (u \to v)$ is an edge such that $\Lambda(e) = 0$. Then the distribution generated by M can be represented as the product of two Two-State MET distributions.*

Proof: For any string $s \in \{0,1\}^n$, let s_1 denote the part of s that lies along the leaves of the subtree rooted at v and s_2 denote the rest of the string. Define two new Two-State METs called M_1 and M_2. Let M_1 have the topology and edge probabilities of the subtree rooted at v and the root probability $\Pr[M_1](v = 0) = \Pr[M](v = 0)$; let M_2 have the topology obtained by disconnecting the subtree rooted at v and removing the edge $(u \to v)$, and have the edge probabilities and the root probability of the original MET M. Note that $\Pr[M](s_1 \mid v = 0) = \Pr[M_1](s_1 \mid v = 0)$ and $\Pr[M](s_1 \mid v = 1) = \Pr[M_1](s_1 \mid v = 1)$.

Now, because $\Lambda(e) = 0$, Observation 2.3 implies that either $\Pr[M](u = 0) \in \{0,1\}$ or $e_0+e_1 = 1$. If $\Pr[M](u = 0) = 0$, then if we write $\Pr[M](s)$ in terms of two conditional distributions conditioned on $u = 0$ and $u = 1$, then Observation 2.8 implies that

$$\begin{aligned}
\Pr[M](s) &= \Pr[M](s_1 \mid u = 1) \Pr[M](s_2 \mid u = 1) \\
&= \Pr[M](s_1 \mid u = 1) \Pr[M](s_2) \\
&= (e_1 \Pr[M](s_1 \mid v = 0) + (1 - e_1) \Pr[M](s_1 \mid v = 1)) \Pr[M](s_2) \\
&= (e_1 \Pr[M_1](s_1 \mid v = 0) + (1 - e_1) \Pr[M_1](s_1 \mid v = 1)) \Pr[M_2](s_2)
\end{aligned}$$

Also, $\Pr[M](v = 0) = e_1$ and $\Pr[M](v = 1) = (1-e_1)$, so $\Pr[M_1](s_1) = e_1 \Pr[M_1](s_1 \mid v = 0)+(1-e_1) \Pr[M_1](s_1 \mid v = 1))$ and therefore $\Pr[M](s) = \Pr[M_1](s_1) \Pr[M_2](s_2)$. The argument for $\Pr[M](u = 1) = 0$ is similar.

Alternatively, assume that $\Pr[M](u = 0) \in (0,1)$ and that $e_0+e_1 = 1$. Using Observation 2.8, we can write $\Pr(s)$ as:

$$\Pr[M](s) \quad = \quad \Pr[M](v = 0) \Pr[M](s_1 \mid v = 0) \Pr[M](s_2 \mid v = 0) +$$

$$\Pr[M](v=1)\Pr[M](s_1 \mid v=1)\Pr[M](s_2 \mid v=1)$$

Notice that

$$\Pr[M_1](s_1) = Pr[M](v=0)\Pr[M](s_1 \mid v=0) + \Pr[M](v=1)\Pr[M](s_1 \mid v=1)$$

Therefore, to show that $\Pr[M](s) = \Pr[M_1](s_1)\Pr[M_2](s_2)$, it is enough to show (a) if $\Pr[M](v=0) \neq 0$, then $\Pr[M](s_2 \mid v=0)$ equals $\Pr[M_2](s_2)$ (which equals $\Pr[M](s_2)$), and (b) if $\Pr[M](v=1) \neq 0$, then $\Pr[M](s_2 \mid v=1) = \Pr[M_2](s_2)$.

If $\Pr[M](v=0) = e_1 \neq 0$, then $\Pr[M](s_2 \mid v=0) = \Pr[M](s_2 \,\&\, v=0)/e_1$. Also,

$$\begin{aligned}\Pr[M](s_2 \,\&\, v=0) &= \Pr[M](u=0)\Pr[M](s_2|u=0)(1-e_0) + \\ &\quad \Pr[M](u=1)\Pr[M](s_2|u=1)e_1\end{aligned}$$

However, we know that $(1-e_0) = e_1$, so $\Pr[M](s_2 \,\&\, v=0)/e_1 =$

$$\Pr[M](u=0)\Pr[M](s_2|u=0) + \Pr[M](u=1)\Pr[M](s_2|u=1),$$

which is $\Pr[M](s_2)$. Also, as long as $\Pr[M](v=1) = e_0 \neq 0$, $\Pr[M](s_2 \mid v=1)$ is equal to $\Pr[M](s_2 \,\&\, v=1)/\Pr[M](v=1)$.

$$\begin{aligned}\Pr[M](s_2 \,\&\, v=1) &= \Pr[M](u=0)\Pr[M](s_2|u=0)e_0 + \\ &\quad \Pr[M](u=1)\Pr[M](s_2|u=1)(1-e_1)\end{aligned}$$

Now $e_0 = (1-e_1)$, so $\Pr[M](s_2 \,\&\, v=1)/e_0 = \Pr[M](s_2)$ also. Therefore we have

$$\begin{aligned}\Pr[M](s) &= \Pr[M](v=0)\Pr[M](s_1 \mid v=0)\Pr[M](s_2) + \\ &\quad \Pr[M](v=1)\Pr[M](s_1 \mid v=1)\Pr[M](s_2) \\ &= \Pr[M_1](s_1)\Pr[M_2](s_2)\end{aligned}$$

as required. $\square$

From now on we will refer to edges that satisfy $\Lambda(e) = 0$ as *product edges* or *cut edges*. It is easy to use the last observation to show that the location of a cut edge cannot be inferred from the distribution of a Two-State MET. Observation 2.14 implies that the distribution is the product of two sub-MET distributions. Now notice that if both these sub-METs are re-rooted at any two internal nodes r_1 and r_2 (Observation 2.9), then by adding a new edge $(r_1 \rightarrow r_2)$ between the two sub-METs, and defining $e_0 = \Pr(r_2 = 1)$ and $e_1 = \Pr(r_2 = 0)$, we will obtain the product distribution of the two sub-METs. However, r_1 and r_2 can be any internal nodes in their respective sub-METs.

Finally, we make the following observation, which will be very useful in Subsection 2.3.3:

Observation 2.15 *Let M be an MET on the rooted topology T such that none of its edges are cut edges. Partition the leaves of T into two sets S_1 and S_2 such that $\mathrm{cov}(s, s') > 0$ if either $s, s' \in S_1$ or $s, s' \in S_2$ holds, and such that $\mathrm{cov}(s, s') < 0$ otherwise (S_2 may be empty). There are at least two "good" labellings on the rooted topology T that generate the same distribution as M: for one of these labellings, the leaf edges to the S_1 leaves are good and the leaf edges to the S_2 leaves are bad; there is another labelling in which this situation is reversed.*

Proof: We already know that there is at least one labelling on T that generates the same distribution as M (Observation 2.11) and whose internal edges are all good. For this labelling, define S_1 to be the set consisting of all the leaves that have bad leaf edges and S_2 to contain all the other leaves (one of these sets may be empty). It is easy to check by Equation 2.10 that these sets satisfy the conditions above.

Now we show that there is another good labelling on T that generates M, such that the leaf edges to the S_1 leaves are good and the leaf edges to the S_2

leaves are bad. We define the new labelling as follows: Let $\rho'_0 = \rho_1$. For every internal edge, define $e'_0 = e_1$ and $e'_1 = e_0$, and for every leaf edge, let $e'_0 = 1 - e_1$ and $e'_1 = 1 - e_0$. Clearly all of the internal edges are still good, and the leaf edges have swapped bad and good roles. Let M' be the new MET. Let the *almost-leaves* of T be the set of nodes of T that are parents of leaves of T. Any labelling of T generates some joint distribution on the almost-leaves of T. Also, the probability that M generates s is equal to the sum, over all labellings ℓ for the almost-leaves, of the probability of ℓ multiplied by the probability that s is generated when the almost-leaves are labelled by ℓ. For any leaf $1 \leq i \leq n$, let s_i be value of the i-th leaf in T, and let u_i and e_i denote the ancestor and incoming edge of i. The probability that M' generates s on its leaves is

$$\sum_{all\ \ell} \Pr[M'](\ell) \left(\prod_{i=1}^{n} M'_{e_i}[\ell(u_i), s_i] \right)$$

Define $\bar{\ell}$ to be the labelling in which 0 and 1 are exchanged. We claim that $\Pr[M](\ell) = \Pr[M'](\bar{\ell})$ for every labelling on the almost-leaves. Then $M'_{e_i}[\ell(u_i), s_i] = M_{e_i}[\bar{\ell}(u_i), s_i]$ always holds, and substituting into the expression above, we find that $\Pr[M](s) = \Pr[M'](s)$.

Now we use induction to prove that for every labelling ℓ on the internal nodes of T, $\Pr[M](\ell) = \Pr[M'](\bar{\ell})$. This is obvious for the base case, when the only node is the root. Otherwise suppose the claim holds for a set of nodes, and consider all the children of those nodes. It is easy to check that when e'_0 and e'_1 are replaced by e_1 and e_0, and the induction hypothesis is used, that the claim holds for the set containing the original nodes and their child nodes. By induction, the claim holds. $\square$

2.3.3 Reconstructing an MET from its Exact Distribution

The results presented in Subsection 2.3.2 allow us to make certain assumptions about a Two-State MET, as long as we are only interested in reconstructing the distribution of the MET. We assume that $(1 - e_0 - e_1) \in (0, 1)$ holds for every internal edge e, and that every cut edge in the tree is adjacent to the root of the tree. Therefore we can view the topology as consisting of a root ρ, such that for every outgoing edge $(\rho \to r)$, the MET rooted at r does not contain any cut edges. Subsection 2.3.3 describes how to reconstruct a Two-State MET that satisfies the conditions above.

First of all we define a *leaf connectivity graph* whose vertices correspond to the leaves of M. There is a positive edge between x and y if $\text{cov}(x, y) > 0$, a negative edge between x and y if $\text{cov}(x, y) < 0$ and no edge otherwise. The maximal connected components of this graph correspond to the sub-METs obtained when all the cut edges in M are deleted. By Observation 2.14, we can consider each maximal related set C separately. If, for each maximal related set C, we construct a Two-State MET $M'(C)$ that generates the same distribution as the original induced distribution on the leaves in C, then the product distribution of these METs generates the original distribution M.

For any connected component C of the leaf connectivity graph, $\Lambda(e) \neq 0$ for every edge e in $M(C)$ (otherwise if $\Lambda(e) = 0$ for some edge in $M(C)$, then $\text{cov}(x, y) = 0$ holds for every pair of leaves whose path crosses e, and C would not be connected). Therefore, by the results of Steel [58] presented in Subsection 2.1.1, the unrooted topology $T(C)$ (with all degree 2 nodes contracted) can be reconstructed in polynomial-time from the absolute values of the leaf-pair covariances in C, as we discussed in Section 2.1. If we choose some internal node r of $T(C)$ to serve as the root of $M'(C)$, then by Observation 2.9, there is some labelling of $T(C)$ rooted at r that generates $M(C)$. Now partition C into two sets C_1 and C_2 so that the covariance of two leaves from the same set is positive and the covariance of two leaves from different sets is negative (see Observation 2.15). By Observation 2.15, there is a labelling $M'(C)$ of $T(C)$ rooted at r, such that all the internal edges are good, the leaf edges for C_1 are good and the leaf edges for C_2 are bad, and such that $M'(C)$ generates the same distribution as $M(C)$. We now show how to construct a Two-State MET $M'(C)$ that satisfies these conditions; in fact, we will show that if $|C| \geq 3$, there is exactly one such MET.

First assume that there are at least three leaves in C (the other two cases are easier and will be presented on page 70). Assume that x, y and z are three leaves in C whose connecting paths meet at some node u in $T(C)$, and assume without loss of generality that u lies on the path from the root r of $T(C)$ to the leaf y (this will hold for at least two of the three leaves). Let y_0 and y_1 denote the transition probabilities along the directed path $(u \to y)$ in $M'(C)$. We will first show how to calculate y_0, y_1 and $\Pr[M'](u = 1)$. Define

$$\begin{aligned}\mathrm{cov}(x, z, 0) &= \Pr(xyz = 101)\Pr(y = 0) - \Pr(xy = 10)\Pr(zy = 10), \\ \mathrm{cov}(x, z, 1) &= \Pr(xyz = 111)\Pr(y = 1) - \Pr(xy = 11)\Pr(zy = 11).\end{aligned} \tag{2.13}$$

(These two quantities are related to the conditional covariances of x and z, conditioned on $y = 0$ and $y = 1$ respectively.) We also define

$$F = \frac{1}{2}\left(\frac{\mathrm{cov}(x, z) + \mathrm{cov}(x, z, 0) - \mathrm{cov}(x, z, 1)}{\mathrm{cov}(x, z)}\right), \text{and} \tag{2.14}$$

$$D = F^2 - \mathrm{cov}(x, z, 0)/\mathrm{cov}(x, z) \tag{2.15}$$

Now suppose that we use the procedure defined in Observation 2.9 to re-root $M'(C)$ at u. Let x_0 and x_1 be the transition probabilities along the directed path $(u \to x)$ in this new MET and z_0 and z_1 be the transition probabilities along the directed path $(u \to z)$ in this new MET. By checking the proof of Observation 2.9, note that changing the root of M' from r to u does not change the transition probabilities along $(u \to y)$, nor does it change the probability $\Pr(u = 1)$. Therefore, to calculate y_0, y_1 and $\Pr[M'](u = 1)$, we may assume that M' is rooted at u. From now on, we will assume that $\Pr(u = 1)$ denotes $\Pr[M'](u = 1)$. It can be shown by algebraic manipulation of Equation 2.13 that

$$\mathrm{cov}(x, z, 0) = \Pr(u = 1)\Pr(u = 0)(1 - x_0 - x_1)(1 - z_0 - z_1)y_1(1 - y_0) \quad (2.16)$$

$$\mathrm{cov}(x, z, 1) = \Pr(u = 1)\Pr(u = 0)(1 - x_0 - x_1)(1 - z_0 - z_1)y_0(1 - y_1)$$

and by a little more algebraic manipulation of Equations 2.14 and 2.15, using Equation 2.10, that

$$F = \frac{1 + y_1 - y_0}{2} \quad (2.17)$$

$$D = \frac{(1 - y_0 - y_1)^2}{4} \quad (2.18)$$

First suppose that $y \in C_1$. Then the leaf edge to y is a good edge, and because we have assumed that $M'(C)$ is a good labelling, therefore all the edges on the path $(u \to y)$ are good edges. Then, since $(1 - y_0 - y_1) = \prod_{e \in (u \to y)}(1 - e_0 - e_1)$, we have $(1 - y_0 - y_1) > 0$. Then $\sqrt{D} = (1 - y_0 - y_1)/2$ and we find

$$y_0 = 1 - (\sqrt{D} + F) \quad \text{and that} \quad y_1 = F - \sqrt{D} \quad (2.19)$$

Note that since M' is a good labelling, then once we have decided which of the two sets is C_1 (the leaves with good leaf edges), there is a unique solution for y_0 and y_1. Therefore, once we have chosen r and C_1, there is only one good labelling $M'(C)$ that generates the same distribution as $M(C)$. To find the value of $\Pr(u = 1)$, notice that

$$\Pr(y = 0) = \Pr(u = 1)y_1 + (1 - \Pr(u = 1))(1 - y_0) \quad (2.20)$$

$$= (1 - y_0) - \Pr(u = 1)(1 - y_0 - y_1) \quad (2.21)$$

Then, since $\sqrt{D} = (1 - y_0 - y_1)/2$, and because $1 - y_0 = \sqrt{D} + F$, we obtain

$$\Pr(u = 1) \quad = \quad \frac{1}{2} + \frac{F - \Pr(y = 0)}{2\sqrt{D}} \tag{2.22}$$

so the value of $\Pr(u = 1)$ can also be obtained from the exact distribution.

Alternatively, suppose that $y \in C_2$. In this case, the leaf edge to y is a non-good edge. Then $(1 - y_0 - y_1) < 0$, so $-\sqrt{D} = (1 - y_0 - y_1)/2$ and therefore

$$y_0 = 1 + \sqrt{D} - F \quad \text{and that} \quad y_1 = F + \sqrt{D} \tag{2.23}$$

To find the value of $\Pr(u = 1)$ in this case, we substitute $-2\sqrt{D}$ for $(1 - y_0 - y_1)$ and substitute $F - \sqrt{D}$ for $1 - y_0$ into Equation 2.21, giving

$$\Pr(u = 1) \quad = \quad \frac{1}{2} + \frac{\Pr(y = 0) - F}{2\sqrt{D}} \tag{2.24}$$

The only thing left to do is to show how to use the probabilities for these directed paths to obtain a complete labelling of $M'(C)$. First of all, to find the root probability $\Pr[M'](r = 1)$, we choose any three leaves that meet at the root r of $T(C)$, and use Equation 2.22 (or Equation 2.24) to calculate $\Pr[M'](r = 1)$. For every leaf edge $(u \rightarrow y)$ in the tree, we choose two other leaves x and z in the tree, and either use Equations 2.19 (if $y \in C_1$) or Equations 2.23 (if $y \in C_2$) to calculate y_0 and y_1. To find the transition probabilities for an internal edge e of the tree, we use quartets of leaves that have the topology shown in Figure 2.1. In other words, we choose four leaves w, x, y and z such that the path between x and y contains e, the path from w to u only intersects the (x, y) path at u, and the path from z to v only intersects the (x, y) path at v.

Denote the probabilities along the entire path $(u \rightarrow y)$ by p_0 and p_1, and the transition properties for $(v \rightarrow y)$ by q_0 and q_1. Then, if we take the triple w, x and y, we can calculate p_0, p_1 using Equation 2.19 (if $y \in C_1$) or Equations 2.23 (if $y \in C_2$). Taking the triple x, y and z, we can calculate q_0, q_1 and $(1 - q_0 - q_1)$.

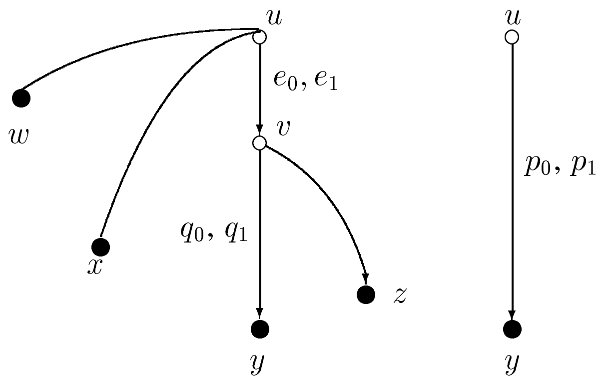


Figure 2.1: We assume that $(u \to y)$ is a directed path in the rooted tree that we are labelling, but make no assumptions about the (u, w) path or the (u, x) path.

By Observation 2.4, we know that $p_0 = q_0 + e_0(1 - q_0 - q_1)$ and that $p_1 = q_1 + e_1(1 - q_0 - q_1)$. Assuming that $(1 - q_0 - q_1)$ is non-zero (this must be true, or otherwise C could not be a related set), we find that

$$e_0 = \frac{p_0 - q_0}{(1 - q_0 - q_1)} \quad \text{and also} \quad e_1 = \frac{p_1 - q_1}{(1 - q_0 - q_1)} \tag{2.25}$$

Therefore we can construct an entire labelling $M'(C)$ for any related set C that has at least three leaves.

It is very easy to construct a labelling for $M(C)$ when C has less than 3 elements. If C has just one leaf, then we can construct $M'(C)$ by letting the single leaf x be the "root" of the tree and assigning $\Pr(x = 0)$ to be the probability at the root. When there are only two leaves x and y in the related set, there are many Two-State METs that will generate the distribution on these leaves. The topology $T(C)$ must consist of a path between the two leaves. We can assume that $\Pr(x = 0) \in (0, 1)$ holds, because $\operatorname{cov}(x, y) \neq 0$. Insert a root r somewhere on the path from x to y. We construct $M'(C)$ as follows: First define $\Pr(r = 0) = \Pr(x = 0)$. Then for the edge $e = (r \to x)$, define $e_0 = e_1 = 0$. For $f = (r \to y)$, define $f_0 = \Pr(xy = 01)/\Pr(x = 0)$

and $f_1 = \Pr(xy = 10)/\Pr(x = 1)$. Clearly this generates the distribution on x and y.

2.4 Estimating the topology of a related set

Now we describe an algorithm that allows us to reconstruct an approximate topology for any Two-State MET whose inter-leaf covariances are all non-zero. Our results will be described in terms of two parameters $c \in (0, 1/4)$ and $d \in (0, 1/2)$. The input to this algorithm consists of a set of leaves C, and for every pair of leaves x and y in this set, an estimate $\widehat{\mathrm{cov}}(x, y)$ of the covariance between those two leaves. The following assumptions are made.

Assumption: We assume that each of the estimated covariances is within additive error $c^3 d/128$ of its true value. Also, we assume that if we define a *leaf connectivity graph* on the leaves in C by adding an edge between x and y iff $|\widehat{\mathrm{cov}}(x, y)| \geq c$, then C is a *related set* for the threshold c (defined in a similar way to the related sets on page 48 of Section 2.2, but using the threshold c instead of $\epsilon_2/2$).

At this stage we should point out that we are most interested in solving the problem for $c = (\epsilon_2/2)$ and $d = \epsilon_3$, for the *maximal* related sets C that were constructed in Section 2.2. However, our theorem will hold for any related set C satisfying the assumption above:

Theorem 2.16 *There is an algorithm that finds a d-contraction $\widehat{T}(C)$ of the topology $T(C)$, when it is given an estimate covariance $\widehat{\mathrm{cov}}(x, y)$ for every pair of leaves $x, y \in C$, and these estimates satisfy the conditions above. The algorithm runs in time polynomial in the size of the input, which is the set of $\widehat{\mathrm{cov}}(x, y)$ estimates for C.*

Our algorithm constructs $\widehat{T}(C)$ inductively, adding one leaf at a time. At each state of the process, we will let S denote the related set of leaves from C which have already been added to the estimate topology. We will let $T(S)$ denote the induced topology of $T(C)$ on the leaves in S, in which all internal nodes have degree at least 3. We will represent the d-contraction of $T(S)$ by $\widehat{T}(S)$.

Our algorithm will maintain the following invariant, where S is the related set of leaves that have been added to the topology:

Invariant: $\widehat{T}(S)$ is a d-contraction of $T(S)$. Also, for any internal edge e of $T(S)$ that is *not* contracted in the topology $\widehat{T}(S)$, $\Lambda(e) \leq 1 - 7d/8$.

We will insist that at each stage of the construction process, if S is the subset of C which has already been added to the estimate topology, then we choose a new leaf ℓ to add to $\widehat{T}(S)$ by choosing some leaf ℓ such that $S \cup \{\ell\}$ is a related set; in other words $|\widehat{\mathrm{cov}}(\ell, x)| \geq c$ for some $x \in S$. Then there are three steps in the algorithm for adding the leaf ℓ to $\widehat{T}(S)$. Let ℓ' be the internal node of $T(S \cup \{\ell\})$ such that (ℓ', ℓ) is a leaf edge.

1. First we use the relatedness condition to show that we can find the approximate location of ℓ' in $\widehat{T}(S)$ by restricting attention to a subset of S, which we call S_ℓ.

2. Next we identify a single path in the tree $\widehat{T}(S)$ between two leaves from the set S_ℓ. This path has the property that a d-contraction of $T(S \cup \{\ell\})$ may be formed, either by adding a new edge between an existing node in $\widehat{T}(S)$ and the new leaf ℓ, or by introducing a new node along an edge in $\widehat{T}(S)$ and adding an edge between this new node and the new leaf ℓ.

3. Finally we describe two tests that can be performed along this path to find the node or edge where ℓ should be attached.

The first subsection presents some results that will be useful for proving the correctness of our algorithm.

2.4.1 Good estimators and apparently good estimators

The following observation follows immediately from the assumptions that we have made about the closeness of our covariance estimates and the connectedness of the leaf connectivity graph.

Observation 2.17 *Let S be a related set for the threshold c. For every edge e of $T(S)$, there are leaves x and y which are connected through e and satisfy $|\mathrm{cov}(x, y)| \geq (63c/64)$. Note that this Observation also holds for C, since C is a related set.*

Proof: Let $e = (u, v)$, and let S_1 be the leaves in S on the "u-side" of e and S_2 be the leaves in S on the "v-side" of e. Since (u, v) is an edge in $T(S)$, this is a partition of S. Since S is connected in the leaf connectivity graph, there is some $x \in S_1$ and $y \in S_2$ such that $|\widehat{\mathrm{cov}}(x, y)| \geq c$. Also, because all of the covariance estimates lie within additive error $c^3d/128$ of their true values, therefore $|\mathrm{cov}(x, y)| \geq c - c^3d/128$. Clearly $c - c^3d/128 \geq (63c/64)$. □

Before we continue, we will introduce some extra notation which we will use throughout this chapter. As we have already noted, for any related set S, we use $\widehat{T}(S)$ to denote the d-contraction of $T(S)$ that has already been constructed by our algorithm. We will use the notation $\mathcal{U}$ and $\mathcal{V}$ to refer to nodes in $\widehat{T}(S)$, and use $\mathcal{E}$ to refer to edges in $\widehat{T}(S)$.

Definition 2.4 *Let S be a related set, and suppose that $\widehat{T}(S)$ is some d-contraction of the topology $T(S)$. Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ be an edge in $\widehat{T}(S)$. Then there is some unique edge $e = (u, v)$ of $T(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$. We will adopt the convention that $\Lambda(\mathcal{E})$ denotes $\Lambda(e)$.*

Also, for for any two nodes u and v of $T(S)$, let $\Lambda(u,v)$ denote the product of the Λ-weights on the edges on the path (u,v). For any two nodes $\mathcal{U}$ and $\mathcal{V}$ in $\widehat{T}(S)$, let $u' \in \mathcal{U}$ and $v' \in \mathcal{V}$ be the unique pair of nodes in $T(S)$ such that the (u',v')-path in $T(S)$ does not contain any other nodes from $\mathcal{U}$ or $\mathcal{V}$. Then we will let $\Lambda(\mathcal{U},\mathcal{V})$ denote the value of $\Lambda(u',v')$ in $T(S)$.

At this point we note that if $\mathcal{U}$ and $\mathcal{V}$ are two nodes in $\widehat{T}(S)$ and $(\mathcal{U},\mathcal{V})$ is not an edge of $\widehat{T}(S)$, then $\Lambda(\mathcal{U},\mathcal{V})$ does not necessarily equal the product of $\Lambda(\mathcal{E})$ over all edges in the path $(\mathcal{U},\mathcal{V})$. This is because $\Lambda(\mathcal{U},\mathcal{V})$ is defined to be $\Lambda(u',v')$, and the path between u' and v' in $T(S)$ may contain edges that are contracted in $\widehat{T}(S)$.

Our algorithm relies heavily on the concepts of a good estimator and an apparently good estimator.

Definition 2.5 *Let $e = (u,v)$ be an internal edge in $T(S)$. We say that the four leaves w,x,y,z from S form a ξ-*good estimator *(for some $\xi > 0$) of the edge e iff*

1. *The edge e is an edge on the path between x and y, and $|\mathrm{cov}(x,y)| \geq \xi$;*

2. *The three leaves w, x and y meet at u in $T(S)$, and $\Lambda(w,u) \geq \xi$;*

3. *The three leaves z, x and y meet at v in $T(S)$, and $\Lambda(z,v) \geq \xi$.*

We will assume wlog that w and x both lie to the "u-side" of e, and write $(w,x \mid y,z)$. For a leaf edge (u,v) (under the assumption that v is the leaf) a good estimator is a pair of leaves (w,x) such that v,x and w meet at u and $|\mathrm{cov}(x,v)| \geq \xi$ and $\Lambda(w,u) \geq \xi$.

We can make the same definition for an internal path (u,v) (where u and v are both internal nodes) or a path ending at a leaf v (where u is an internal node and v is a leaf). Note that Equation 2.12 implies that $\Lambda(w,u) \geq \mathrm{cov}(w,u)$ for any internal node u and any leaf w. Also, if x and y are leaves in a Two-State MET, and u lies on the path from x to y, then by Equation 2.10

$$|\mathrm{cov}(x,y)| \quad = \quad \Lambda(x,u)\Lambda(u,y) \tag{2.26}$$

By Observation 2.6, this implies that $|\mathrm{cov}(x,y)| \leq \Lambda(x,u)$.

Whenever we try to add a new leaf ℓ to a d-contraction $\widehat{T}(S)$, we cannot assume that a node in $\widehat{T}(S)$ represents a single node from the true tree $T(S)$. We will denote a node of $\widehat{T}(S)$ by $\mathcal{U}$ than u, and an edge by $\mathcal{E}$ rather than e.

Definition 2.6 *Let $\mathcal{E} = (\mathcal{U},\mathcal{V})$ be an edge in $\widehat{T}(S)$. The four leaves w,x,y,z from S form an ξ-*apparently good estimator *of $\mathcal{E}$ if and only if*

1. *The edge $\mathcal{E}$ is an edge on the path between x and y, and $|\mathrm{cov}(x,y)| \geq \xi$;*
2. *The three leaves w, x and y meet at some u' in $T(S)$ such that $u' \in \mathcal{U}$ in $\widehat{T}(S)$, and $\Lambda(w,u') \geq \xi$;*
3. *The three leaves z, x and y meet at some v' in $T(S)$ such that $v' \in \mathcal{V}$ in $\widehat{T}(S)$, and $\Lambda(z,v') \geq \xi$;*

*We will assume wlog that w and x both lie to the "u'-side" of (u',v'), and write $(w,x \mid y,z)$. Again, a ξ-*apparently good estimator *of a leaf edge $(\mathcal{U},v)$ is a pair of leaves (w,x) such that $|\mathrm{cov}(x,v)| \geq \xi$ and there is some $u' \in \mathcal{U}$ such that v,x and w meet at u' and $\Lambda(u',w) \geq \xi$.*

This definition is easily modified in the obvious way to define an apparently good estimator for a path between two internal nodes $\mathcal{U}$ and $\mathcal{V}$, or for a path between the internal node $\mathcal{U}$ and the leaf node v.

The concept of a good estimator will be used in two situations. In Subsection 2.4.2, we will use good estimators to help construct the tree. In Section 2.5

we will show how to obtain transition probabilities for the edges using good estimators. Since we only construct a d-contraction of $T(S)$, a node in $\widehat{T}(S)$ may contain more than one node from $T(S)$. Remember that for any edge $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ in $\widehat{T}(S)$, $\widehat{\Lambda}(\mathcal{E})$ is defined to be $\Lambda(e)$, where $e = (u, v)$ is the unique edge of $T(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$. We will now show how to estimate $\Lambda(\mathcal{E})$, for any internal edge $\mathcal{E}$ of $\widehat{T}(S)$, for a related set S. Our method, which will be used to maintain the invariant in Step 3 (b) of the algorithm (see page 107), uses the following Observation:

Observation 2.18 *Let w, x, y and z be four leaves of the related set S such that w, x and y meet at the node u in $T(S)$ and z, x and y meet at the node v in $T(S)$. Assume wlog that w and x lie to the "u-side" of (u, v) in $T(S)$ and that y and z lie to the "v-side" of (u, v) in $T(S)$ (Figure 2.2 depicts the situation when (u, v) is an edge). Since S is related,* $\mathrm{cov}(w, x) \neq 0$ *and* $\mathrm{cov}(y, z) \neq 0$ *hold. Also,*

$$\Lambda(u, v) = \sqrt{\frac{\mathrm{cov}(w, z)\mathrm{cov}(x, y)}{\mathrm{cov}(w, x)\mathrm{cov}(y, z)}} = \sqrt{\frac{\mathrm{cov}(w, y)\mathrm{cov}(x, z)}{\mathrm{cov}(w, x)\mathrm{cov}(y, z)}} \tag{2.27}$$

Proof: First notice that if we use Equation 2.11 to substitute for the covariances, then we find that

$$\Lambda(u, v) = \sqrt{\frac{|\mathrm{cov}(w, z)\mathrm{cov}(x, y)|}{|\mathrm{cov}(w, x)\mathrm{cov}(y, z)|}} = \sqrt{\frac{|\mathrm{cov}(w, y)\mathrm{cov}(x, z)|}{|\mathrm{cov}(w, x)\mathrm{cov}(y, z)|}}$$

To prove that Equation 2.27 holds, we only need to show that $\mathrm{cov}(w, z)\mathrm{cov}(x, y)$ and $\mathrm{cov}(w, y)\mathrm{cov}(x, z)$ both have the same sign as $\mathrm{cov}(w, x)\mathrm{cov}(y, z)$. As an example, we will show that $\mathrm{cov}(w, z)\mathrm{cov}(x, y)$ has the same sign as $\mathrm{cov}(w, x)\mathrm{cov}(y, z)$. First of all, note that the re-rooting process defined in Observation 2.9 does not change the sign of $(1 - e_0 - e_1)$ along the edge e that is re-rooted. Therefore, to prove that $\mathrm{cov}(w, z)\mathrm{cov}(x, y)$ has the same sign as $\mathrm{cov}(w, x)\mathrm{cov}(y, z)$, we will make the assumption that the quartet is rooted at the internal node u. Let the probabilities on the path $(u \to v)$ be denoted by p_0 and p_1; the probabilities on

$(u \to w)$ be w_0 and w_1; the probabilities on $(u \to x)$ be x_0 and x_1; the probabilities on $(v \to y)$ be y_0 and y_1; and the probabilities on $(v \to z)$ be z_0 and z_1. Then, using Equation 2.10, and using Observation 2.4, we know that

$$\begin{aligned}
\mathrm{cov}(w,z) &= \Pr(u=0)\Pr(u=1)(1-w_0-w_1)(1-e_0-e_1)(1-z_0-z_1)\\
\mathrm{cov}(x,y) &= \Pr(u=0)\Pr(u=1)(1-x_0-x_1)(1-e_0-e_1)(1-y_0-y_1)\\
\mathrm{cov}(w,x) &= \Pr(u=0)\Pr(u=1)(1-w_0-w_1)(1-x_0-x_1)\\
\mathrm{cov}(y,z) &= \Pr(v=0)\Pr(v=1)(1-z_0-z_1)(1-y_0-y_1)
\end{aligned}$$

Then $\mathrm{cov}(w,z)\mathrm{cov}(x,y) = \mathrm{cov}(w,x)\mathrm{cov}(y,z)\frac{\Pr(u=0)\Pr(u=1)}{\Pr(v=0)\Pr(v=1)}(1-e_0-e_1)^2$. Since $(1-e_0-e_1)^2$ is positive, therefore $\mathrm{cov}(w,z)\mathrm{cov}(x,y)$ and $\mathrm{cov}(w,x)\mathrm{cov}(y,z)$ have the same sign. □

Estimating $\Lambda(\mathcal{E})$

The proof of the following relations is straightforward. We will typically apply them in situations in which ξ is the error of an approximation.

$$\begin{aligned}
\frac{r+\xi}{s-\xi} &= \frac{r}{s} + \left(\frac{\xi}{s-\xi}\right)\left(1+\frac{r}{s}\right) && (2.28)\\
\frac{1+\xi}{1-\xi} &\le 1+4\xi \quad \text{if } \xi \le 1/2 && (2.29)\\
\frac{1-\xi}{1+\xi} &\ge 1-2\xi \quad \text{if } \xi \ge 0 && (2.30)\\
\sqrt{r(1+2\xi)} &\le \sqrt{r}(1+\xi) \quad \text{if } r,\xi \ge 0 && (2.31)\\
\sqrt{r(1-\xi)} &\ge \sqrt{r}(1-\xi) \quad \text{if } r \ge 0 \text{ and } \xi < 1 && (2.32)
\end{aligned}$$

In the next few paragraphs we will show how to estimate $\Lambda(\mathcal{E})$ within multiplicative error $d/32$. There are two cases.

Case 1: $\mathcal{E}$ is an internal edge

Observation 2.19 *Let e be an edge of $T(S)$ and $(w, x \mid y, z)$ be an ξ-good estimator of e, and assume wlog that $|\mathrm{cov}(x, y)| \geq \xi$. Then $|\mathrm{cov}(w, x)| \geq \xi^2$, $|\mathrm{cov}(y, z)| \geq \xi^2$, $|\mathrm{cov}(w, y)| \geq \xi^2$ and $|\mathrm{cov}(x, z)| \geq \xi^2$ all hold. Also, $|\mathrm{cov}(w, z)| \geq \xi^3$.*

Proof: Each of these inequalities can be proved using the definition of a good estimator, Equation 2.26 and the fact that $\Lambda(u, v) \leq 1$ for any two nodes u and v in a Two-State MET. □

Lemma 2.20 *Let $(w, x \mid y, z)$ be any quartet such that w, x and y meet at u in $T(S)$ and z, x and y meet at v in $T(S)$. Wlog assume that w and x lie to the "u-side" of (u, v) (so $(w, x \mid y, z)$ satisfies the topological constraints of a good estimator of (u, v)). Suppose also that*

$$|\mathrm{cov}(w, x)| \geq (15c/16)^2 \quad |\mathrm{cov}(y, z)| \geq (15c/16)^2$$
$$|\mathrm{cov}(w, y)| \geq (15c/16)^2 \quad |\mathrm{cov}(x, z)| \geq (15c/16)^2$$

Then we can use Equation 2.27 to estimate $\Lambda(u, v)$ within multiplicative error $d/32$.

Proof: We will first show that our estimates of each of the four relevant covariances lie within multiplicative error $cd/32$ of their true values. First consider $\mathrm{cov}(w, x)$. We want to show that

$$|\mathrm{cov}(w, x) - \widehat{\mathrm{cov}}(w, x)| \leq |\mathrm{cov}(w, x)| cd/32.$$

Since we have assumed that $|\mathrm{cov}(w, x) - \widehat{\mathrm{cov}}(w, x)| \leq c^3 d/128$, we only need to show that $c^3 d/128 \leq |\mathrm{cov}(w, x)| cd/32$. This is true if and only if $c^2/4 \leq |\mathrm{cov}(w, x)|$. However, we know that $|\mathrm{cov}(w, x)| \geq (15c/16)^2$, and $(15c/16)^2 \geq c^2/4$, so therefore $|\mathrm{cov}(w, x) - \widehat{\mathrm{cov}}(w, x)| \leq |\mathrm{cov}(w, x)| cd/32$. This proof only depended on the fact that $|\text{cov}(w,x)|$ is at least $(15c/16)^2$, and on the fact that our estimate $\widehat{\text{cov}}(w,x)$ lies within additive error $c^3d/128$ of the true value. Therefore $\text{cov}(y,z)$, $\text{cov}(w,y)$ and $\text{cov}(x,z)$ also lie within multiplicative error $cd/32$ of their true values.

Using these observed covariances in Equation 2.27 with $\widehat{\text{cov}}(w,y)\widehat{\text{cov}}(x,z)$ as the numerator,

$$\begin{aligned}\sqrt{\frac{\widehat{\text{cov}}(w,y)\widehat{\text{cov}}(x,z)}{\widehat{\text{cov}}(w,x)\widehat{\text{cov}}(y,z)}} &\leq \sqrt{\frac{\text{cov}(w,y)\text{cov}(x,z)(1+cd/32)^2}{\text{cov}(w,x)\text{cov}(y,z)(1-cd/32)^2}} \\ &\leq \Lambda(u,v)\frac{1+cd/32}{1-cd/32}\end{aligned}$$

and by Inequality 2.29, this is at most $\Lambda(e)(1+cd/8)$. In the same way, we can show that $\widehat{\Lambda}(u,v)$ is at least $\Lambda(u,v)(1-cd/16)$, using Inequality 2.30. The result follows because $c \leq 1/4$. $\square$

Corollary 2.21 *Let $(w,x \mid y,z)$ be a $(15c/16)$-good estimator of the edge e in $T(S)$. Then we can use our estimates of covariances among the leaves w, x, y and z in Equation 2.27 to estimate $\Lambda(e)$ within multiplicative error $d/32$.*

Proof: By Observation 2.19, we know that for any $(15c/16)$-good estimator $(w,x \mid y,z)$ of the edge e, we have

$$\begin{aligned}&|\text{cov}(w,x)| \geq (15c/16)^2 \quad |\text{cov}(y,z)| \geq (15c/16)^2 \\ &|\text{cov}(w,y)| \geq (15c/16)^2 \quad |\text{cov}(x,z)| \geq (15c/16)^2\end{aligned}$$

Then Lemma 2.20 gives the result. $\square$

Observation 2.22 *Let $e = (u,v)$ be an internal edge in $T(S)$, for the related set S. Then there is a $(63c/64)$-good estimator $(w,x \mid y,z)$ of e in S. Also, if there is an internal edge $\mathcal{E} = (\mathcal{U},\mathcal{V})$ in $\widehat{T}(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$, then every $(63c/64)$-good estimator of e taken from S is a $(63c/64)$-apparently good estimator of $\mathcal{E}$. (Refer to Figure 2.2.)*

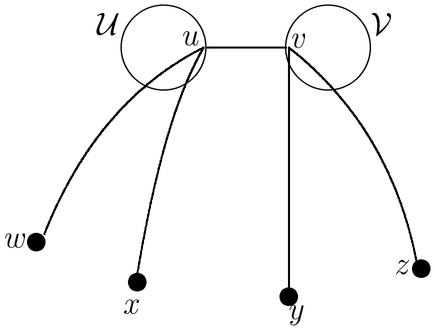


Figure 2.2: $(w, x \mid y, z)$ is a good estimator of $e = (u, v)$ and an apparently good estimator of $\mathcal{E} = (\mathcal{U}, \mathcal{V})$.

Proof: Leaves x and y can be found to satisfy the first criterion of the definition of a $(63c/64)$-good estimator for e by Observation 2.17. Then, since the degree of u is at least three, consider any edge $f = (u, t)$ that does not lie on the path between x and y. By Observation 2.17, there is a pair of leaves in S such that f lies on the path between these leaves and the absolute value of their covariance is at least $(63c/64)$. Choose w to be the leaf from this pair such that (u, w) does not overlap with (x, y). Then, by Equation 2.26, $\Lambda(u, w) \geq (63c/64)$. Leaf z can be found in a similar way. Therefore $(w, x \mid y, z)$ is a $(63c/64)$-good estimator for e. When there is an edge $\mathcal{E}$ satisfying the conditions of the problem statement, then $(w, x \mid y, z)$ is an apparently good estimator of $\mathcal{E}$ because only internal edges of $T(S)$ can be contracted in the d-contraction $\widehat{T}(S)$. □

Observation 2.23 *Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ be an internal edge in $\widehat{T}(S)$ and let $e = (u, v)$ be the edge in $T(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$. Suppose that $(w, x \mid y, z)$ is an ξ-apparently good estimator of $\mathcal{E}$, and let u' be the meeting point of x, w and y in $T(S)$ and v' be the meeting point of y, x and z in $T(S)$ (Refer to Figure 2.3). Then $(w, x \mid y, z)$ is a ξ-good estimator of the path (u', v').*

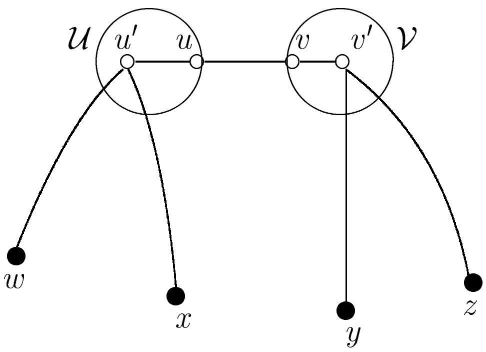


Figure 2.3: $(w, x \mid y, x)$ is an apparently good estimator of $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ and a good estimator of $p = (u', v')$.

Proof: The fact that $(c, b \mid a, d)$ is a good estimator of (u', v') follows from the definition of good estimator. $\square$

Since the Λ weights are multiplicative (see Equation 2.11), and because of Observation 2.6, $\Lambda(p) \leq \Lambda(e)$ for every internal path p that contains the edge e. We will use this fact to describe how, for any internal edge $\mathcal{E}$ of a related set S, we can use a collection of quartets that satisfy the conditions of an apparently good estimator (and satisfy some bounds on the absolute values of the inter-leaf covariances) to estimate $\Lambda(\mathcal{E})$ within multiplicative error $d/32$. Note that since we only have estimates of the covariances for pairs of leaves in S, it is not easy to check (even approximately) that conditions 2 and 3 of an apparently good estimator hold for a quartet. Therefore, we will prove the following result:

Lemma 2.24 *Let $\mathcal{E}$ be an internal edge from $\mathcal{U}$ to $\mathcal{V}$ in $\widehat{T}(S)$, for the related set S. Remember that our covariance estimates lie within additive error $c^3 d/128$ of their true values. Suppose that we estimate $\Lambda(\mathcal{E})$ as follows: For every quartet $(w', x' \mid y', z')$ in S that satisfies the* topological constraints *of an apparently good estimator for $\mathcal{E}$, such that*

$$|\widehat{\mathrm{cov}}(w', x')| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(y', z')| \geq (31c/32)^2$$

$$|\widehat{\mathrm{cov}}(w', y')| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(x', z')| \geq (31c/32)^2$$

we estimate $\Lambda(u', v')$ using Equation 2.27, if w, x and y meet at $u' \in \mathcal{U}$ and z, x and y meet at $v' \in \mathcal{V}$. Define $\widehat{\Lambda}(\mathcal{E})$ to be the largest of these estimates. Then

$$\widehat{\Lambda}(\mathcal{E}) \in [\Lambda(\mathcal{E})(1 - d/32), \Lambda(\mathcal{E})(1 + d/32)].$$

Proof: Suppose $e = (u, v)$ is the edge in $T(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$. Remember that by definition $\Lambda(\mathcal{E}) = \Lambda(e)$, so we need to show that our estimate lies within multiplicative error $d/32$ of $\Lambda(e)$. First remember that by Observation 2.22, there is some $(63c/64)$-good estimator $(w, x \mid y, z)$ for the $e = (u, v)$ in $T(S)$. Assume wlog that x and y are the two leaves whose absolute covariance is at least $(63c/64)$. Then, by the definition of a good estimator, and by Observation 2.19,

$$|\mathrm{cov}(w, x)| \geq (63c/64)^2 \quad |\mathrm{cov}(y, z)| \geq (63c/64)^2$$

$$|\mathrm{cov}(x, z)| \geq (63c/64)^2 \quad |\mathrm{cov}(w, y)| \geq (63c/64)^2$$

Therefore $|\widehat{\mathrm{cov}}(w, x)| \geq (63c/64)^2 - c^3d/128 = c^2((63/64)^2 - cd/128)$. Since $c < 1/4$ and $d < 1/2$ we have $|\widehat{\mathrm{cov}}(w, x)| \geq c^2((63/64)^2 - 1/1024)$. A check on a calculator verifies that $(63/64)^2 - 1/1024 \geq (31/32)^2$; therefore $|\widehat{\mathrm{cov}}(w, x)| \geq (31c/32)^2$. Similarly, each of the estimates $|\widehat{\mathrm{cov}}(y, z)|$, $|\widehat{\mathrm{cov}}(x, z)|$ and $|\widehat{\mathrm{cov}}(w, y)|$ are at least $(31c/32)^2$. Therefore, the $(63c/64)$-good estimator $(w, x \mid y, z)$ of e is one of the quartets considered by the procedure described in the statement of this Lemma.

Now suppose that $(w', x' \mid y', z')$ is any quartet such that

$$|\widehat{\mathrm{cov}}(w', x')| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(y', z')| \geq (31c/32)^2$$

$$|\widehat{\mathrm{cov}}(w', y')| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(x', z')| \geq (31c/32)^2$$

holds. For any estimated covariance which has absolute value at least $(31c/32)^2$, the true absolute value of that covariance is at least $(31c/32)^2 - c^3d/128$. Also,

$(31c/32)^2 - c^3d/128 = c^2((31/32)^2 - cd/128)$, and since $c < 1/4$ and $d < 1/2$, this is at least $c^2((31/32)^2 - 1/1024)$. A check with a calculator verifies that $(31/32)^2 - 1/1024 \geq (15/16)^2$. Therefore, for any covariance estimate whose absolute value is at least $(31c/32)^2$, the true absolute value of the covariance is at least $(15c/16)^2$. Therefore

$$|\mathrm{cov}(w', x')| \geq (15c/16)^2 \quad |\mathrm{cov}(y', z')| \geq (15c/16)^2$$
$$|\mathrm{cov}(w', y')| \geq (15c/16)^2 \quad |\mathrm{cov}(x', z')| \geq (15c/16)^2.$$

holds for any of the quartets that are tested by the procedure described in the statement of this Lemma. Then by Lemma 2.20, for every $(w', x' \mid y', z')$ that is tested, the estimate $\widehat{\Lambda}(u', v')$ obtained by using our covariance estimates in Equation 2.27 satisfies

$$\widehat{\Lambda}(u', v') \in [(1 - d/32)\Lambda(u', v'), (1 + d/32)\Lambda(u', v')].$$

Note that this bound also holds for the quartet $(w, x \mid y, z)$, when the path (u', v') being estimated is actually the edge e.

Now notice that for any two nodes $u' \in \mathcal{U}$ and $v' \in \mathcal{V}$, the path (u', v') contains the edge e. Also, by the definition of $\Lambda(u', v')$ along any path, and by Observation 2.6, $\Lambda(u', v') \leq \Lambda(e)$ holds for any two nodes $u' \in \mathcal{U}$ and $v' \in \mathcal{V}$.

Let $\widehat{\Lambda}(\mathcal{E})$ be the largest estimate obtained over all the quartets $(w', x' \mid y', z')$ satisfying the conditions of this Lemma. Denote the unknown path being estimated by the "winning" quartet by (u'', v''). Then $\widehat{\Lambda}(\mathcal{E}) \leq (1 + d/32)\Lambda(u'', v'')$, and since $\Lambda(u'', v'') \leq \Lambda(e)$, therefore $\widehat{\Lambda}(\mathcal{E}) \leq (1 + d/32)\Lambda(e)$. Also, we know that the estimate obtained using the quartet $(w, x \mid y, z)$ is at least $(1 - d/32)\Lambda(e)$. Therefore, since $\widehat{\Lambda}(\mathcal{E})$ was defined as the largest estimate, $\widehat{\Lambda}(\mathcal{E}) \geq \Lambda(e)(1 - d/32)$. So $\widehat{\Lambda}(\mathcal{E}) \in [\Lambda(\mathcal{E})(1 - d/32), \Lambda(\mathcal{E})(1 + d/32)]$, as required. □

Case 2: $\mathcal{E}$ is a leaf edge

In this case we will assume that $\mathcal{E} = (\mathcal{U}, v)$, where v is a leaf edge.

Observation 2.25 *Let $e = (u,v)$ be a leaf edge in the topology $T(S)$, for some related set S. There is a $(63c/64)$-good estimator of e in S. Also, if $\mathcal{E} = (\mathcal{U}, v)$ is a leaf edge in the topology $\widehat{T}(S)$ such that $u \in \mathcal{U}$, then every $(63c/64)$-good estimator of e taken from S is a $(63c/64)$-apparently good estimator of $\mathcal{E}$.*

Proof: First note that by Observation 2.17, there must be some $x \in S$ such that $|\mathrm{cov}(x,v)| \geq (63c/64)$, or otherwise S would not be a related set. Let this leaf be x in the definition of a $(63c/64)$-good estimator for e. Since u is not a leaf, it has degree at least three in $T(S)$. Therefore there must be another edge f in $T(S)$ which has u as one of its endpoints but which does not lie on (x,v). By Observation 2.17, there must be a pair of leaves $w, z \in S$ such that the (w,z) path contains f and $|\mathrm{cov}(w,z)| \geq (63c/64)$. Assume without loss of generality that w is not on the "u-side" of f. By Equation 2.26, $\Lambda(w,u) \geq (63c/64)$. Also, the pair (w,x) satisfy the topological constraints of a good estimator for (u,v).

The proof that every $(63c/64)$-good estimator of e is a $(63c/64)$-apparently good estimator of $\mathcal{E}$ follows directly from the two definitions. $\square$

Observation 2.26 *Let $\mathcal{E} = (\mathcal{U}, v)$ be a leaf edge in the topology $\widehat{T}(S)$, and let $e = (u,v)$ be the edge in $T(S)$ such that $u \in \mathcal{U}$. Then, if (w,x) is a ξ-apparently good estimator of $\mathcal{E}$ and u' is the meeting point of v, w and x in $T(S)$, (w,x) is a ξ-good estimator of the path (u',v) in $T(S)$.*

Proof: The proof that (w,x) is an ξ-good estimator of (u',v) depends on the fact that v, w and x meet at u' and that (w,x) is a ξ-apparently good estimator of $\mathcal{E}$. $\square$

2.4.2 The algorithm

Here is the algorithm that is used to add a leaf ℓ to an existing d-contraction $\widehat{T}(S)$. Remember that by construction, we assume that S is a related set and that the new leaf ℓ is related to one of the elements of this set. In other words, we choose ℓ such that $|\widehat{\text{cov}}(\ell, x)| \geq c$ for some $x \in S$. Since C is a related set for the threshold c, we know that if $S \neq C$, some ℓ satisfying this condition must exist. Then $S \cup \{\ell\}$ is also a related set for the threshold c. We will denote the point in $T(S)$ where ℓ is attached to the tree by ℓ'; that is, (ℓ', ℓ) is a leaf edge in $T(S)$. As we mentioned before, the algorithm maintains the following invariant:

> After a new leaf has been added to the topology, $\widehat{T}(S)$ is a d-contraction of $T(S)$. Also, for every internal edge $\mathcal{E}$ in the d-contraction, it is guaranteed that $\Lambda(\mathcal{E}) \leq 1 - 7d/8$.

Remember from Definition 2.4 that if $\mathcal{E} = (\mathcal{U}, \mathcal{V})$, then $\Lambda(\mathcal{E})$ denotes the value of $\Lambda(e)$ in $T(S)$, where $e = (u, v)$ is the edge of $T(S)$ such that $u \in \mathcal{U}$ and $v \in \mathcal{V}$.

Step 1: Define $S_\ell = \{x \in S : |\widehat{\text{cov}}(\ell, x)| \geq (15c/16)^3\}$

Let $T(S_\ell)$ be the induced topology obtained from $T(S)$ by contracting any degree 2 nodes in the subtree of $T(S)$ that spans the leaves in S_ℓ. Let $\widehat{T}(S_\ell)$ be defined in the same way from the d-contraction $\widehat{T}(S)$. Here are two interesting facts about S_ℓ:

- The first interesting fact is that for any $x, y \in S_\ell$, $|\text{cov}(x, y)| \geq (7c/8)^6$. To see why this is true, remember that $|\text{cov}(x, y) - \widehat{\text{cov}}(x, y)| \leq c^3 d/128$ for every pair of leaves $x, y \in C$. By construction, $|\widehat{\text{cov}}(\ell, x)| \geq (15c/16)^3$ for every $x \in S_\ell$. Therefore if $x \in S_\ell$, then by our assumption about the closeness of estimates, $|\mathrm{cov}(\ell, x)| \geq (15c/16)^3 - c^3d/128 = c^3((15/16)^3 - d/128)$. A check with a calculator verifies that $(15/16)^3 - d/128 \geq (7/8)^3$ (since $d < 1/2$), and therefore $|\mathrm{cov}(\ell, x)| \geq (7c/8)^3$ for every $x \in S_\ell$. Also, for any pair of leaves $x, y \in S_\ell$, Equation 2.26 implies that the absolute value of the covariance between x and y is at least $(7c/8)^6$.

- The next few observations show that to find the approximate position of ℓ' in $\widehat{T}(S)$ (allowing edges with $\Lambda(e)$ close to 1 to be contracted), we only need to find the approximate position of ℓ' in $\widehat{T}(S_\ell)$.

Observation 2.27 *For every edge e in $T(S \cup \{\ell\})$, if ℓ' is one of the endpoints of e, there is a $(63c/64)$-good estimator of e in $S \cup \{\ell\}$, such that ℓ is one of the leaves of this estimator.*

Proof: Let the edge e be (ℓ', v), and note that by definition of ℓ', ℓ' is an internal node (with degree at least 3) of $T(S \cup \{\ell\})$. Remember that $S \cup \{\ell\}$ was constructed to be a related set (for the threshold c). First suppose that (ℓ', v) is an internal edge of $T(S \cup \{\ell\})$. Then by Observation 2.22, there is some $(63c/64)$-good estimator $(w, x \mid y, z)$ for (ℓ', v) in $S \cup \{\ell\}$. If ℓ is one of the four leaves w, x, y and z, then we are finished. Otherwise, assume wlog that w and x lie to the "ℓ'-side" of (ℓ', v) and that $|\mathrm{cov}(x, y)| \geq (63c/64)$. We now show that $(\ell, x \mid y, z)$ is also a $(63c/64)$-good estimator of (ℓ', v). Since $|\mathrm{cov}(x, y)| \geq (63c/64)$ and (ℓ', v) lies on the path (x, y), the first condition of a good estimator is satisfied. Also, since $(w, x \mid y, z)$ was an $(63c/64)$-good estimator for (ℓ', v), x, y and z meet at v in $T(S \cup \{\ell\})$ and $\Lambda(v, z) \geq (63c/64)$. So the third condition of a $(63c/64)$-good estimator holds for $(\ell, x \mid y, z)$. Finally, since ℓ is related to S, then by Observation 2.17 and Equation 2.26, the leaf edge (ℓ', ℓ) satisfies $\Lambda(\ell', \ell) \geq (63c/64)$. Also, x, y and ℓ meet at ℓ'. Therefore $(\ell, x \mid y, z)$ satisfies the second and final condition of a $(63c/64)$-good estimator of (ℓ', v) in $T(S \cup \{\ell\})$.

Now suppose that (ℓ', v) is a leaf edge of $T(S \cup \{\ell\})$. By Observation 2.25, there is some $(63c/64)$-good estimator (w, x) of (ℓ', v) in $S \cup \{\ell\}$. If v is the leaf ℓ, then ℓ is certainly one of the leaves of the estimator, so we are finished. If v is not the leaf ℓ, but ℓ is either w or x, then we are also finished. Otherwise, assume wlog that $|\mathrm{cov}(x, v)| \geq (63c/64)$. Now notice that since (ℓ', ℓ) is a leaf edge in $T(S \cup \{\ell\})$, ℓ, x and v meet at ℓ' in $T(S \cup \{\ell\})$. Also, by Observation 2.17 and Equation 2.26, the leaf edge (ℓ', ℓ) satisfies $\Lambda(\ell', \ell) \geq (63c/64)$. So (ℓ, x) satisfies the conditions of a $(63c/64)$-good estimator of (ℓ', v). $\square$

Observation 2.28 *Suppose that e is an edge in $T(S \cup \{\ell\})$ and there is some $(63c/64)$-good estimator of e such that one of the leaves of the estimator is ℓ. Then all the other leaves of that estimator (three leaves if e is internal, or two leaves if e is a leaf edge) lie in S_ℓ.*

Proof: First of all consider an internal edge e with the $(63c/64)$-good estimator $(w, x \mid y, \ell)$. By Observation 2.19, $|\mathrm{cov}(w, \ell)|$, $|\mathrm{cov}(x, \ell)|$ and $|\mathrm{cov}(y, \ell)|$ are all at least $(63c/64)^3$. If $e = (u, v)$ is a leaf edge in which v is the leaf, and (ℓ, x) is the $(63c/64)$-good estimator of e, then using Equation 2.26, we find that $|\mathrm{cov}(x, \ell)|$ and $|\mathrm{cov}(v, \ell)|$ are both at least $(63c/64)^2$, which is certainly at least $(63c/64)^3$.

To finish the proof, we only need to show that for every covariance whose true absolute value is at least $(63c/64)^3$, the absolute value of the estimate will be at least $(15c/16)^3$. Then all the leaves of these special good estimators will lie in S_ℓ. Let $|\mathrm{cov}(x, \ell)| \geq (63c/64)^3$. Then, because our covariance estimates lie within additive error $c^3 d/128$ of their true values, we know that $|\widehat{\mathrm{cov}}(x, \ell)| \geq (63c/64)^3 - c^3 d/128 = c^3((63/64)^3 - d/128)$. We know $d < 1/2$, and a check on a calculator verifies that $(63/64)^3 - 1/256 \geq (15/16)^3$. So $|\widehat{\mathrm{cov}}(x, \ell)| \geq (15c/16)^3$ and therefore $x \in S_\ell$. $\square$

Corollary 2.29 *ℓ' is a node (with degree at least 3) in $T(S_\ell \cup \{\ell\})$.*

Proof: First assume that there is some internal edge (ℓ', v) in $T(S \cup \{\ell\})$ with ℓ' as one of its endpoints. Then by Observation 2.27, there is some $(63c/64)$-good estimator of (ℓ', v) in $S \cup \{\ell\}$ which has ℓ as one of its leaves. Let this estimator be $(\ell, x \mid y, z)$. By Observation 2.28 , each of x, y and z lie in S_ℓ, so therefore ℓ' will have degree at least 3 in $T(S_\ell \cup \{\ell\})$. If there is no internal edge in $T(S \cup \{\ell\})$ with endpoint ℓ', there must be at least one leaf edge (ℓ', v) in $T(S \cup \{\ell\})$ such that $v \neq \ell$, and by Observation 2.27 there is a $(63c/64)$-good estimator (ℓ, x) of (ℓ', v) in $S \cup \{\ell\}$). By Observation 2.28, x and v both lie in S_ℓ, so ℓ' will have degree at least 3 in $T(S_\ell \cup \{\ell\})$. $\square$

Note that Corollary 2.29 indicates that if our aim was to reconstruct the topology *exactly* (for METs for which this is possible), then if we know $T(S)$ and $T(S_\ell \cup \{\ell\})$, we can reconstruct $T(S \cup \{\ell\})$. We will prove something slightly weaker about $\widehat{T}(S_\ell \cup \{\ell\})$ in Lemma 2.33 at the end of Step 1. The following two Observations are important.

Observation 2.30 *If ℓ' lies on the edge $e = (u, v)$ of $T(S)$ and $u \in \mathcal{U}$ and $v \in \mathcal{V}$ for some edge $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ in $\widehat{T}(S)$, then $\mathcal{E}$ is an edge in $\widehat{T}(S_\ell)$.*

Proof: Remember that by Observation 2.27, each of the edges (u, ℓ') and (ℓ', v) has a $(63c/64)$-good estimator in $S \cup \{\ell\}$, and we can assume that each of these estimators has ℓ as one of its leaves. By Observation 2.28, all the "non-ℓ" leaves in these two estimators belong to S_ℓ.

Assume wlog that u is an internal node of $T(S)$. Then if $(w, x \mid \ell, y)$ is the $(63c/64)$-good estimator for (u, ℓ'), we know that w, x and y meet at u in $T(S)$, and since $w, x, y \in S_\ell$, therefore u is also a node (with degree at least 3) in $T(S_\ell)$. If v is also an internal node of $T(S)$, then we can show that v has degree at least 3 in $T(S_\ell)$ in the same way. So (u, v) will be an edge in $T(S_\ell)$ and $\mathcal{E}$ will be an edge in $\widehat{T}(S_\ell)$. If v is a leaf of $T(S)$, let (ℓ, x) denote the $(63c/64)$-good estimator for (ℓ', v). Then by Observation 2.28, $v \in S_\ell$. Then, using the fact that u has degree at least 3 in $T(S_\ell)$, we find that (u, v) is an edge in $T(S_\ell)$, and $\mathcal{E}$ is an edge in $\widehat{T}(S_\ell)$. $\square$

Observation 2.31 *If ℓ' lies on the edge (u, v) of $T(S)$ and u and v are both contained in node $\mathcal{U}$ of $\widehat{T}(S)$ then $\mathcal{U}$ is a node of $\widehat{T}(S_\ell)$. Alternatively, if ℓ' is already a node in $T(S)$ and ℓ' is contained in the node $\mathcal{U}$ in $\widehat{T}(S)$ then $\mathcal{U}$ is also a node in $\widehat{T}(S_\ell)$.*

Proof: First suppose that ℓ' lies on some edge (u, v) of $T(S)$ that is contained in node $\mathcal{U}$. Note that this implies that (u, v) is an internal edge of $T(S)$ (leaf edges are never contracted). Then Observation 2.27 implies that there is a $(63c/64)$-good estimator for (u, ℓ') with ℓ as one of its leaves and a $(63c/64)$-good estimator for (ℓ', v) with ℓ as one of its leaves. By Observation 2.28, u and v will both have degree at least three in $T(S_\ell)$. Therefore $\mathcal{U}$ will be a node in $\widehat{T}(S_\ell)$.

If ℓ' is already a node in $T(S)$, then ℓ' has degree at least 3 in $T(S)$. Also, by Observation 2.27, there is a $(63c/64)$-good estimator for every edge e adjacent to ℓ' in $T(S \cup \{\ell\})$, such that ℓ is a leaf of this good estimator. By Observation 2.28, all the leaves of these estimators will belong to S_ℓ. Then, because ℓ' has degree at least 3 in $T(S)$, ℓ' will also have degree at least 3 in $T(S_\ell)$. Therefore $\mathcal{U}$ will be a node in $\widehat{T}(S_\ell)$. $\square$

The algorithm presented in the paper by Cryan, Goldberg and Goldberg [15] used tests on the edges of $\widehat{T}(S_\ell)$ to determine an approximate location (allowing edges with $\Lambda(\mathcal{E}) \geq 1 - d$ to be contracted) for ℓ'. Under the assumption that we had estimates of covariances within multiplicative error for *every* pair of leaves in S_ℓ, we showed how to use these estimates to add ℓ to the topology. The version of the algorithm presented here is very similar to the original one; the only difference is that the covariance estimates only need to lie within multiplicative error of their true values for the covariances whose absolute value is at least $(15c/16)^3$.

In Step 3 (a) of the algorithm, we will show how to add ℓ to $\widehat{T}(S_\ell)$ to obtain a d-contraction $\widehat{T}(S_\ell \cup \{\ell\})$ of $T(S_\ell \cup \{\ell\})$. The following Observation will be useful in Step 3 (a):

Observation 2.32 *Remember that S_ℓ was defined on page 85. Let $\widehat{T}(S_\ell)$ be the subtree of $\widehat{T}(S)$ induced by the leaves in S_ℓ, in which any degree 2 nodes have been contracted. Then for any internal edge $\mathcal{E}$ of $\widehat{T}(S_\ell)$, $\Lambda(\mathcal{E}) \leq (1 - 7d/8)$.*

Proof: Remember that we have maintained the invariant that for every internal edge e of $T(S)$ that is not contracted in $\widehat{T}(S)$, $\Lambda(e) \leq (1 - 7d/8)$.

Now suppose $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ is an internal edge of $\widehat{T}(S_\ell)$. Let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ be the two nodes of $T(S_\ell)$ such that (u, v) is an edge in $T(S_\ell)$ (so u and v will have degree at least 3 in $T(S_\ell)$). Remember that $\Lambda(\mathcal{E})$ is defined as $\Lambda(u, v)$. Clearly the edge (u, v) of $T(S_\ell)$ corresponds to some path (u, v) in $T(S)$. Also, there must be some edge f of $T(S)$ on the path (u, v) in $T(S)$, such that f is *not* contracted in $\widehat{T}(S)$ (otherwise (u, v) would not correspond to an edge in $\widehat{T}(S_\ell)$). Then $\Lambda(f) \leq (1 - 7d/8)$. By definition, we know that $\Lambda(u, v) = \prod_{e \in (u,v)} \Lambda(e)$, multiplying along the path (u, v) in $T(S)$. Then, by Equation 2.11 and Observation 2.6, $\Lambda(u, v) \leq \Lambda(f) \leq (1 - 7d/8)$, and we are finished. $\square$

The following Lemma proves that we can use the d-contraction $\widehat{T}(S_\ell \cup \{\ell\})$ to construct a d-contraction of $T(S \cup \{\ell\})$:

Lemma 2.33 *Suppose that $\widehat{T}(S)$ is a d-contraction of the related set S satisfying the invariant quoted at the beginning of Subsection 2.4.2. Let ℓ be related to S and let $\widehat{T}(S_\ell)$ be the subtree induced by the leaves in S_ℓ. Finally, let $\widehat{T}(S_\ell \cup \{\ell\})$ be a d-contraction of $T(S_\ell \cup \{\ell\})$ constructed by adding ℓ to a node or along an* *edge of $\widehat{T}(S_\ell)$. Then the node or edge of $\widehat{T}(S_\ell)$ where ℓ was attached is also a node or edge of $\widehat{T}(S)$. Also, if we attach ℓ to the same node or edge in $\widehat{T}(S)$ to give the topology $T'(S \cup \{\ell\})$, then $T'(S \cup \{\ell\})$ is a d-contraction of $T(S \cup \{\ell\})$.*

Proof: We assume that $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$ constructed by adding ℓ to a node or along an edge of $\widehat{T}(S_\ell)$. Remember that ℓ' is the node in $T(S \cup \{\ell\})$ such that (ℓ', ℓ) is a leaf edge in $T(S \cup \{\ell\})$. Also remember that by Corollary 2.29, ℓ' lies in $T(S_\ell \cup \{\ell\})$.

First suppose that $\widehat{T}(S_\ell \cup \{\ell\})$ was constructed from $\widehat{T}(S_\ell)$ by attaching ℓ to an existing node $\mathcal{U}$ of $\widehat{T}(S_\ell)$. Note that because $\widehat{T}(S_\ell)$ was defined as the induced topology of $\widehat{T}(S)$ on the leaves in S_ℓ (where all nodes have degree 3), every node in $\widehat{T}(S_\ell)$ is also a node in $\widehat{T}(S)$. Also, every node in $T(S_\ell)$ is also a node in $T(S)$. So in this case, $\mathcal{U}$ lies in $\widehat{T}(S)$, and we can construct $T'(S \cup \{\ell\})$ by attaching ℓ to the node $\mathcal{U}$ in $\widehat{T}(S)$. We need to show that $T'(S \cup \{\ell\})$ will then be a d-contraction of $T(S \cup \{\ell\})$.

Since $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$, there are only three explanations for attaching ℓ to $\mathcal{U}$ in $\widehat{T}(S_\ell)$: The first possibility is that ℓ' is a node of $T(S_\ell)$ that lies inside $\mathcal{U}$ in $\widehat{T}(S_\ell)$. Then ℓ' is also a node in $T(S)$, and ℓ' lies in $\mathcal{U}$ in $\widehat{T}(S)$. Attaching ℓ to $\mathcal{U}$ in $\widehat{T}(S)$ does not contract any edges of $T(S \cup \{\ell\})$ that were not already contracted in $\widehat{T}(S)$, so $T'(S \cup \{\ell\})$ is a d-contraction of $T(S \cup \{\ell\})$. The second possibility is that ℓ' lies on some edge (u, v) of $T(S_\ell)$ such that u, v lie within $\mathcal{U}$ in $\widehat{T}(S_\ell)$. By our definition of $\widehat{T}(S_\ell)$, u and v must lie in $\mathcal{U}$ in $\widehat{T}(S)$; also, by Observations 2.27 and 2.28, (u, v) will be an edge (rather than a path) in $T(S)$. Therefore, the new topology $T'(S \cup \{\ell\})$ can be obtained from $T(S \cup \{\ell\})$ by contracting some edges of $T(S \cup \{\ell\})$). Also, the only edges of $T(S \cup \{\ell\})$ that are "newly" contracted by the addition of ℓ to $\widehat{T}(S)$ are (u, ℓ') and (ℓ', v). Since (u, v) was contracted in $\widehat{T}(S)$, $\Lambda(u, v) \geq (1-d)$. Using Equation 2.26

and Observation 2.6, $\Lambda(u, \ell') \geq (1 - d)$ and $\Lambda(\ell', v) \geq (1 - d)$. Then $T'(S \cup \{\ell\})$ is a d-contraction of $T(S \cup \{\ell\})$. Finally, if ℓ' was attached to $\mathcal{U}$ to give $\widehat{T}(S_\ell \cup \{\ell\})$, and ℓ' does not lie inside $\mathcal{U}$ in $\widehat{T}(S_\ell)$, $\mathcal{U}$, then the only other possibility is that ℓ' lies along an edge that is adjacent to $\mathcal{U}$ in $\widehat{T}(S_\ell)$ (if ℓ' lies within another node $\mathcal{V}$ or on an edge that is not adjacent to $\mathcal{U}$ in $\widehat{T}(S_\ell)$, then attaching ℓ to $\mathcal{U}$ would not give a d-contraction of $T(S_\ell \cup \{\ell\})$). Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ be the edge of $\widehat{T}(S_\ell)$ such that ℓ' lies on $\mathcal{E}$, and suppose that $u \in \mathcal{U}$ and $v \in \mathcal{V}$ are the nodes of $T(S_\ell)$ such that (u, v) is an edge in $T(S_\ell)$. Then ℓ' lies on (u, v) in $T(S_\ell)$; also, by Observation 2.30, (u, v) must be an edge of $T(S)$. Then, by attaching ℓ to $\mathcal{U}$ in $\widehat{T}(S)$ to give $T'(S \cup \{\ell\})$, we are contracting the edge (u, ℓ'). However, since this edge was contracted in $\widehat{T}(S_\ell)$, it must be the case that $\Lambda(u, \ell') \geq (1 - d)$. Therefore, $T'(S \cup \{\ell\})$ is a d-contraction of $T(S \cup \{\ell\})$.

Now suppose that $\widehat{T}(S_\ell \cup \{\ell\})$ was constructed from $\widehat{T}(S_\ell)$ by attaching ℓ to an existing edge $\mathcal{E}$ of $\widehat{T}(S_\ell)$. Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$, and suppose that $u \in \mathcal{U}$ and $v \in \mathcal{V}$ are the nodes of $T(S_\ell)$ such that (u, v) is an edge in $T(S_\ell)$. Then by Observation 2.30, (u, v) must be an edge (rather than a path) of $T(S)$. Also, since ℓ was attached to $\widehat{T}(S_\ell)$ by splitting $\mathcal{E}$ and attaching a new leaf edge, then since $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$, ℓ' must lie on (u, v) in $T(S_\ell)$ (and also in $T(S)$). Therefore, by attaching ℓ to the edge $\mathcal{E}$ in $\widehat{T}(S)$, we obtain a d-contraction of $T(S \cup \{\ell\})$. $\square$

Step 2: Find two leaves x and y such that ℓ' lies close to the path (x, y)

Next, by considering $\widehat{T}(S)$ and using our estimates (which lie within additive error $c^3 d/128$ of their true values) of covariances between pairs of leaves in $S \cup \{\ell\}$, we will find $x, y \in S_\ell$ such that the following conditions hold:

(i) $|\widehat{\mathrm{cov}}(\ell, y)| \geq c$;

(ii) $|\widehat{\mathrm{cov}}(x,\ell)| \geq (31c/32)^2$;

(iii) $|\widehat{\mathrm{cov}}(x,y)| \geq (31c/32)^2$;

(iv) One of the following two conditions holds:

- Either ℓ' lies on the path from x to y in $T(S_\ell)$, or
- If ℓ'' is the meeting point of x, y and ℓ in $T(S_\ell)$, then $\Lambda(\ell'',\ell') \geq (1 - d/32)$.

We will soon see that in order to find x and y, we only need to use the estimates of interleaf covariances for pairs of leaves in $S_\ell \cup \{\ell\}$. The choice of y is straightforward: Since the leaf ℓ is related to S, there is at least one leaf in S such that the absolute value of the estimated covariance between this leaf and ℓ is at least c. Without loss of generality, this is the leaf that we will choose as y. By definition of S_ℓ, clearly $y \in S_\ell$. Therefore condition (i) is satisfied by construction. Note that by the closeness of our estimates, we know that $|\mathrm{cov}(\ell,y)| \geq (63c/64)$ for this particular y. The method that we use to find an x satisfying the conditions above involves estimating the Λ-weight of a leaf path. The procedure that we will use to choose x is given on page 95, but first we need some definitions and observations.

Definition 2.7 *For any three leaves x',y',z' from S_ℓ, if u is the meeting point of x', y' and z' in $T(S \cup \{\ell\})$, we will let $\Lambda_{z'}(x',y',z')$ denote $\Lambda(u,z')$, where $\Lambda(u,z')$ is the product of the Λ weights along the (u,z')-path in $T(S)$ (see Definition 2.4). Note that since x', y' and z' also belong to S_ℓ, u also lies in $T(S_\ell)$, and therefore we will also informally refer to $\Lambda(u,z')$ in the tree $T(S_\ell)$.*

Equation 2.10 implies that if $\mathrm{cov}(x',y') \neq 0$,

$$\Lambda_{z'}(x',y',z') = \sqrt{\frac{\mathrm{cov}(x',z')\mathrm{cov}(y',z')}{\mathrm{cov}(x',y')}}. \tag{2.33}$$

By Corollary 2.29, ℓ' has degree at least 3 in $T(S_\ell \cup \{\ell\})$. Then ℓ' must lie along some path in $T(S_\ell)$. Therefore there is some $x \in S_\ell$ such that ℓ' lies on the (x, y)-path in $T(S_\ell)$. Then $\Lambda_\ell(x, y, \ell) = \Lambda(\ell', \ell)$. However, we can also prove the following Observation, which is a slightly stronger version of Observation 2.27:

Observation 2.34 *There is a $(63c/64)$-good estimator of the path from ℓ' to the leaf ℓ in $S_\ell \cup \{\ell\}$ such that y and ℓ are both leaves of this estimator. Let the third leaf be called "real x". In the terms of Equation 2.33 above, $\Lambda_\ell($"real x"$, y, \ell) = \Lambda(\ell', \ell)$. Also, "real x"$\in S_\ell$. For any x' in S_ℓ, $\Lambda_\ell(x', y, \ell) \le \Lambda(\ell', \ell)$.*

Proof: By the closeness of our estimates, we know $|\mathrm{cov}(\ell, y)| \ge (63c/64)$. First assume that ℓ' lies on the edge $e = (u, v)$ in $T(S)$ and assume wlog that y lies on the "v-side" of (u, v) (it is possible that v is a leaf and $v = y$). Since S is a related set, there is some "real x" on the "u-side" of e such that $\Lambda(v,$ "real x"$) \ge (63c/64)$, and therefore $\Lambda(\ell',$ "real x"$) \ge (63c/64)$. Therefore ("real x"$, y)$ is a $(63c/64)$-good estimator of (ℓ', ℓ). Alternatively, if ℓ' is a node in $T(S)$, let v be the node adjacent to ℓ' in $T(S)$ such that v lies on the path between ℓ' and y (it is possible that v is a leaf and $v = y$). Let (u, ℓ') be any edge in $T(S)$ such that $u \ne v$. Then, because S is a related set, Observation 2.17 ensures that there are two leaves $x, z \in S$ such that the (x, z)-path contains the edge (u, ℓ') and $|\mathrm{cov}(x, z)| \ge (63c/64)$. Assume wlog that x does not lie to the "v-side" of the edge (ℓ', v), and now denote x by "real x". By Equation 2.26, $\Lambda($"real x"$, \ell') \ge (63c/64)$, and ("real x"$, y)$ is a $(63c/64)$-good estimator of (ℓ', ℓ). By Observation 2.28, whether ℓ' lies on an edge of $T(S)$ or is a node of $T(S)$, both "real x" and y belong to S_ℓ.

The fact that $\Lambda_\ell(x', y, \ell) \le \Lambda(\ell', \ell)$ follows directly from Equation 2.11 and Observation 2.6. $\square$

Observation 2.35 *Let x' be a leaf in S such that $|\widehat{\mathrm{cov}}(x', \ell)|$ and $|\widehat{\mathrm{cov}}(x', y)|$ are both at least $(31c/32)^2$. Therefore, by our definition of S_ℓ, $x' \in S_\ell$. Then, using*

Equation 2.33, with our estimates of the inter-leaf covariances between x' and y, between x' and ℓ, and between y and ℓ, we can estimate $\Lambda_\ell(x', y, \ell)$ within multiplicative error $d/64$. We can also estimate $\Lambda_{x'}(x', y, \ell)$ and $\Lambda_y(x', y, \ell)$ to the same multiplicative error.

Proof: First note that by the closeness of our estimates, the absolute values of $\text{cov}(x', \ell)$, $\text{cov}(x', y)$ and $\text{cov}(\ell, y)$ are all at least $(15c/16)^2$. Since our covariance estimates lie within additive error $c^3d/128$ of their true values, these estimates lie within multiplicative error $cd/64$ of their true values. Substituting the covariance estimates into Equation 2.33, our estimate will satisfy

$$\begin{aligned}
\widehat{\Lambda}_\ell(x', y, \ell) &\leq \sqrt{\frac{\text{cov}(x', \ell)\text{cov}(\ell, y)(1 + cd/64)^2}{\text{cov}(x', y)(1 - cd/64)}} \\
&\leq \Lambda_\ell(x', y, \ell)(1 + cd/16)
\end{aligned}$$

by Inequality 2.29. Also,

$$\begin{aligned}
\widehat{\Lambda}_\ell(x', y, \ell) &\geq \sqrt{\frac{\text{cov}(x', \ell)\text{cov}(\ell, y)(1 - cd/64)^2}{\text{cov}(x', y)(1 + cd/64)}} \\
&\geq \Lambda_\ell(x', y, \ell)(1 + cd/32)
\end{aligned}$$

using Inequality 2.30. Since $c < 1/4$, we have estimated $\Lambda_\ell(x', y, \ell)$ within multiplicative error $d/64$. Note that this proof uses exactly the same assumptions about the absolute value of each of the three covariances. Therefore, the same argument shows that we can obtain estimates for $\Lambda_{x'}(x', y, \ell)$ and $\Lambda_y(x', y, \ell)$ within multiplicative error $d/64$. $\square$

Next we will show how our algorithm chooses the leaf x that satisfies the conditions listed on page 92. The choice of x is made by using each "potential x" to estimate the value of $\Lambda(\ell', \ell)$, and then choosing the leaf x that gives the "best" estimate. We have already chosen y such that $|\text{cov}(\ell, y)| \geq c$. Next, for every leaf $x' \in S_\ell$ that satisfies the following conditions

$$|\widehat{\text{cov}}(x', \ell)| \geq (31c/32)^2 \quad \text{and} \quad |\widehat{\text{cov}}(x', y)| \geq (31c/32)^2,$$

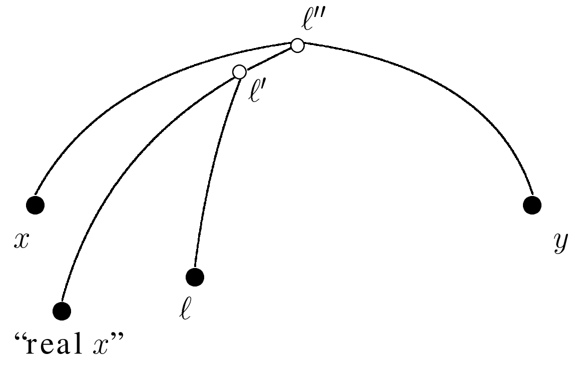


Figure 2.4: The situation in $T(S_\ell \cup \{\ell\})$. The node ℓ' lies along the path between "real x" and y; ℓ' does not lie on the path (x,y), but $\Lambda(\ell'',\ell')$ is close to 1.

we calculate $\widehat{\Lambda}_\ell(x',y,\ell)$. Then we choose x to be the leaf x' that gives the largest value for this estimate, and we calculate $\widehat{\Lambda}_x(x,y,\ell)$ and $\widehat{\Lambda}_y(x,y,\ell)$. We will let ℓ'' denote the meeting point of y, ℓ and the x that has just been chosen. First note that *by construction*, the x that we choose will satisfy conditions (ii) and (iii) from page 92. The following Lemma proves condition (iv):

Lemma 2.36 *Suppose we use the procedure described in the previous paragraph to choose x. Then ℓ' either lies along the path between x and y* or else *the multiplicative distance between ℓ' and ℓ'' is at least* $(1 - d/32)$.

Proof: Suppose that ℓ' does not lie along the path between y and the x that we choose. By Observation 2.34, there is a $(63c/64)$-good estimator in $S_\ell \cup \{\ell\}$ for (ℓ',ℓ) such that y and ℓ are both leaves of this estimator. This estimator is represented in Figure 2.4 by "real x" and y; the figure also shows the actual x chosen by the procedure. By the closeness of our estimates, note that the "real x" will have been tested above.

By Observation 2.35, both $\widehat{\Lambda}_\ell(x, y, \ell)$ and $\widehat{\Lambda}_\ell(\text{“real } x\text{”}, y, \ell)$ are within multiplicative error $d/64$ of their true values. This means that

$$\begin{aligned} \frac{\widehat{\Lambda}_\ell(\text{“real } x\text{”}, y, \ell)}{\widehat{\Lambda}_\ell(x, y, \ell)} &\geq \frac{\Lambda_\ell(\text{“real } x\text{”}, y, \ell)(1 - d/64)}{\Lambda_\ell(x, y, \ell)(1 + d/64)} \\ &\geq \frac{1}{\Lambda(\ell'', \ell')}(1 - d/32), \end{aligned}$$

by Inequality 2.30. Since $\widehat{\Lambda}_\ell(\text{“real } x\text{”}, y, \ell)$ was less than $\widehat{\Lambda}_\ell(x, y, \ell)$, it must be true that $\Lambda(\ell'', \ell') \geq (1 - d/32)$. □

Step 3: Find the location of ℓ'

Finally we describe how to use the fact that ℓ' “almost” lies on the path (x, y) (see condition (iv) on page 92) to construct the d-contraction $\widehat{T}(S \cup \{\ell\})$. Remember that the meeting point of ℓ, x and y in $T(S_\ell \cup \{\ell\})$ is denoted by by ℓ'', and that ℓ' denotes the node in $T(S_\ell \cup \{\ell\})$ such that (ℓ', ℓ) is a leaf edge. In Observation 2.35 of Step 2, we showed that we could calculate estimates for $\Lambda(\ell'', y)$ and of $\Lambda(\ell'', x)$ that lie within multiplicative error $d/64$ of their true values. Throughout Step 3, we will assume that these estimates have already been calculated, and denote them by $\widehat{\Lambda}(\ell'', x)$ and $\widehat{\Lambda}(\ell'', y)$.

There are two sub-steps in Step 3. First of all, in Step 3 (a), we will show how to add ℓ to a node of $\widehat{T}(S_\ell)$ or on an edge of $\widehat{T}(S_\ell)$, to give a d-contraction of $T(S_\ell \cup \{\ell\})$. The position where ℓ will be inserted into $\widehat{T}(S_\ell)$ is determined by two tests which will be used to measure “distance from x” and “distance from y” for certain leaves in S_ℓ (see page 99 for the procedure). The d-contraction of $T(S_\ell \cup \{\ell\})$ that will be constructed by this method will be denoted by $\widehat{T}(S_\ell \cup \{\ell\})$.

By Lemma 2.33, we know that we can attach ℓ to the same node or edge in $\widehat{T}(S)$ to obtain a topology $T'(S \cup \{\ell\})$ that is a d-contraction of $T(S \cup \{\ell\})$. However, $T'(S \cup \{\ell\})$ will not necessarily satisfy the invariant described at the beginning of Subsection 2.4.2; we will see that the proof that Step 3 (a) constructs a d-contraction of $T(S_\ell \cup \{\ell\})$ relies on the fact that the invariant holds for $\widehat{T}(S)$ (in particular, the proof of Lemma 2.41 needs to make this assumption). The procedure of Step 3 (b) described on page 107 shows how to modify $T'(S \cup \{\ell\})$ to give a topology $\widehat{T}(S \cup \{\ell\})$ such that every edge $\mathcal{E}$ in $\widehat{T}(S \cup \{\ell\})$ satisfies $\Lambda(\mathcal{E}) \le (1 - 7d/8)$. Therefore, Step 3 (a) and Step 3 (b) together prove Theorem 2.16.

Step 3 (a): Adding ℓ to $\widehat{T}(S_\ell)$

Definition 2.8 *Let $\mathcal{U}$ be an internal node on (x,y) in $\widehat{T}(S_\ell)$. Let z be a leaf from S_ℓ such that x, y and z meet at $\mathcal{U}$* and $|\widehat{\mathrm{cov}}(z,y)| \ge (15c/16)^3$ *and* $|\widehat{\mathrm{cov}}(z,x)| \ge (15c/16)^3$. *Then we say z is a* good tester *for $\mathcal{U}$.*

For each internal node $\mathcal{U}$ on the path (x,y) in $\widehat{T}(S_\ell)$, and each good tester $z \in S_\ell$ for $\mathcal{U}$, we can perform the following tests.

- **Test$_y(\mathcal{U}, z)$:** The test succeeds iff
$$\frac{\widehat{\Lambda}_y(x,z,y)}{\widehat{\Lambda}(\ell'',y)} \le 1 - 3d/4$$
where $\widehat{\Lambda}(\ell'',y)$ was obtained at the end of Step 2, and $\widehat{\Lambda}_y(x,z,y)$ is obtained using the covariance estimates with Equation 2.33.

- **Test$_x(\mathcal{U}, z)$:** The test succeeds if and only if
$$\frac{\widehat{\Lambda}_x(x,z,y)}{\widehat{\Lambda}(\ell'',x)} \le 1 - 3d/4$$
where $\widehat{\Lambda}(\ell'',x)$ was obtained at the end of Step 2, and $\widehat{\Lambda}_x(x,z,y)$ is obtained using the covariance estimates with Equation 2.33.

It is best to think of Test$_y$ as testing that ℓ'' is to the "y-side" of the meeting point of x, y and z. Similarly, we can think of Test$_x$ as testing that ℓ'' is to the "x-side" of the meeting point of x, y and z.

The leaf ℓ is added to $\widehat{T}(S_\ell)$ in the following way. First $\mathcal{U}$ is initialized to be the internal node of $\widehat{T}(S_\ell)$ that is closest to the leaf y. The algorithm moves along the path towards the leaf x, until the leaf ℓ is placed somewhere. For each node $\mathcal{U}$ in $\widehat{T}(S_\ell)$ on this path, tests will be performed for every good tester $z \in S_\ell$ that meets the path (x,y) at $\mathcal{U}$. The decision on where to place ℓ is made in the following way: First of all, for every leaf $z \in S_\ell$ that is a good tester for $\mathcal{U}$, $\text{Test}_y(\mathcal{U}, z)$ is performed. If all of these tests succeed, then the leaf ℓ is attached to $\widehat{T}(S_\ell)$ by splicing a new node into the adjacent edge on the "y-side" of $\mathcal{U}$ and connecting ℓ to $\widehat{T}(S)$ by an edge to this new node. Otherwise, if even one Test_y fails, $\text{Test}_x(\mathcal{U}, z)$ is performed for every good tester $z \in S_\ell$. If even one Test_x fails then ℓ is attached to the existing topology by the edge $(\mathcal{U}, \ell)$. If all of the Test_x tests succeed, then x will be placed somewhere to the "x-side" of $\mathcal{U}$. If the leaf x is adjacent to $\mathcal{U}$ in $T(S_\ell)$, an edge attaching ℓ is spliced into the middle of $(\mathcal{U}, x)$; otherwise, we move from $\mathcal{U}$ to the next node in $\widehat{T}(S_\ell)$ towards x, and look for the location of ℓ along the rest of the path.

The new topology obtained by using this procedure to attach ℓ to $\widehat{T}(S_\ell)$ will be denoted by $\widehat{T}(S_\ell \cup \{\ell\})$. The following Observations and Lemmas prove that $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$. This will be shown in Lemma 2.41 and Lemma 2.42. First we will show that there is some good tester for every node $\mathcal{U}$ on the path (x,y) in $\widehat{T}(S_\ell)$.

Observation 2.37 *Let $\mathcal{U}$ be an internal node on the path between x and y in $\widehat{T}(S_\ell)$. Let u' be any node in $T(S_\ell)$ such that u' lies on the (x,y)-path in $T(S_\ell)$ and $u \in \mathcal{U}$ in $\widehat{T}(S_\ell)$. Then there is some $z \in S_\ell$ that meets the path (x,y) at u in $T(S_\ell)$ and that is a* good tester *for $\mathcal{U}$ in $\widehat{T}(S_\ell)$.*

Proof: First let e' be any edge in $T(S_\ell)$ such that e' is adjacent to u' but does not lie on the (x,y) path. Note that e' will correspond to some *path* in the tree $T(S)$, and suppose e is the edge of this path that is adjacent to u'. By Observation 2.17, there is some leaf $z \in S$ such that z, x and y meet at u' in $T(S)$ and such that $\Lambda(z,u') \geq (63c/64)$. Also, we know that $|\widehat{\text{cov}}(x,y)| \geq (31c/32)^2$, by the choice of x and y in Step 2. Since we know that $|\widehat{\text{cov}}(x,y) - \text{cov}(x,y)| \leq c^3d/128$ therefore $|\text{cov}(x,y)| \geq (31c/32)^2 - c^3d/128 = c^2((31/32)^2 - cd/128)$. Therefore by Equation 2.26, $\Lambda(u',x) \geq c^2((31/32)^2 - cd/128)$ and $\Lambda(u',y) \geq c^2((31/32)^2 - cd/128)$. Also, because $\Lambda(z,u') \geq (63c/64)$, then, using Equation 2.26, we have $|\text{cov}(x,z)| \geq c^3(63/64)((31/32)^2 - cd/128)$ and $|\text{cov}(y,z)| \geq c^3(63/64)((31/32)^2 - cd/128)$. We know that $c < 1/4$ and $d < 1/2$, and using a calculator, we can verify that $(63/64)((31/32)^2 - 1/1024) \geq (31/32)^3$. Therefore $|\text{cov}(x,z)| \geq (31c/32)^3$ and $|\text{cov}(y,z)| \geq (31c/32)^3$.

Then the estimated covariances $\widehat{\text{cov}}(x,z)$ and $\widehat{\text{cov}}(y,z)$ have absolute values of at least $(31c/32)^3 - c^3d/128$. Another check with a calculator verifies that this value is at least $(15c/16)^3$.

Finally, we need to show that $z \in S_\ell$. First suppose that u' lies to the "y-side" of ℓ'' on the path between x and y. Then by Equation 2.26 we know that $|\text{cov}(z,\ell)| = \Lambda(z,u')\Lambda(u',\ell)$ and also that $\Lambda(u',\ell) \geq |\text{cov}(y,\ell)|$. By construction, $|\text{cov}(y,\ell)| \geq (63c/64)$, so $|\text{cov}(z,\ell)| \geq (63c/64)^2$. Otherwise, if u' lies to the "x-side" of ℓ'', then $|\text{cov}(z,\ell)| \geq \Lambda(z,u')|\text{cov}(x,\ell)|$. Remember that condition (ii) for Step 2 ensures that $|\widehat{\text{cov}}(x,\ell)| \geq (31c/32)^2$. Also, since $\widehat{\text{cov}}(x,\ell)$ lies within additive error $c^3d/128$ of $\text{cov}(x,\ell)$, therefore $|\text{cov}(z,\ell)| \geq (63c/64)((31c/32)^2 - c^3d/128) = c^3(63/64)((31/32)^2 - cd/128)$. A check with a calculator verifies that $(63/64)((31/32)^2 - cd/128) \geq (31/32)^3$. Therefore $|\text{cov}(z,\ell)| \geq (31c/32)^3$. Also, because $\widehat{\text{cov}}(z,\ell)$ lies within additive error $c^3d/128$ of $\text{cov}(z,\ell)$, another check with a calculator verifies that $\widehat{\text{cov}}(z,\ell) \geq (15c/16)^3$, so z will belong to S_ℓ.

The argument above holds for *every* u' that lies on (x,y) in $T(S_\ell)$ and lies inside $\mathcal{U}$ in $\widehat{T}(S_\ell)$. □

The next observation will be useful for proving that the procedure described on page 99 constructs a d-contraction of $T(S_\ell \cup \{\ell\})$.

Observation 2.38 *Let z be any leaf in S_ℓ that forms a good tester for some node along the path (x, y) in $\widehat{T}(S_\ell)$. Then, if we estimate $\Lambda_y(x, z, y)$ and $\Lambda_x(x, z, y)$ using Equation 2.33, we obtain estimates that lie within multiplicative error $d/16$ of the true values.*

Proof: We will estimate $\Lambda_y(x, z, y)$ and $\Lambda_x(x, z, y)$ using $\widehat{\text{cov}}(x, y)$, $\widehat{\text{cov}}(x, z)$ and $\widehat{\text{cov}}(y, z)$ with Equation 2.33. By condition (iii) from page 92, we know that $|\widehat{\text{cov}}(x, y)| \geq (31c/32)^2$. Also, by our definition of a good tester, we know that $|\widehat{\text{cov}}(x, z)| \geq (15c/16)^3$ and $|\widehat{\text{cov}}(y, z)| \geq (15c/16)^3$. Therefore each of the three relevant covariance estimates has an absolute value of at least $(15c/16)^3$. We will use this fact, together with our assumption that our covariance estimates lie within additive error $c^3d/128$ of their true values, to show that each of the three covariance estimates lie within multiplicative error $d/64$ of their true values.

First consider $\text{cov}(x, z)$. We will first show that $|\text{cov}(x, z)| \geq (7c/8)^3$. Since $\widehat{\text{cov}}(x, z)$ is within additive error $c^3d/128$ of its true value, therefore $|\text{cov}(x, z)| \geq (15c/16)^3 - c^3d/128 = c^3((15/16)^3 - d/128)$. $((15/16)^3 - d/128) \geq ((15/16)^3 - 1/256)$, because $d < 1/2$. A check on a calculator verifies that $((15/16)^3 - 1/256) \geq (7/8)^3$. So $|\text{cov}(x, z)| \geq (7c/8)^3$. We also know that $|\widehat{\text{cov}}(x, z) - \text{cov}(x, z)| \leq c^3d/128$, so to show that $|\widehat{\text{cov}}(x, z) - \text{cov}(x, z)| \leq |\text{cov}(x, z)|d/64$, we only need to show that $c^3d/128 \leq |\text{cov}(x, z)|d/64$. This is equivalent to showing that $c^3/2 \leq |\text{cov}(x, z)|$. However, we know that $|\text{cov}(x, z)| \geq (7c/8)^3$, and $(7c/8)^3 \geq c^3/2$, so therefore $c^3/2 \leq |\text{cov}(x, z)|$, and $|\widehat{\text{cov}}(x, z) - \text{cov}(x, z)| \leq |\text{cov}(x, z)|d/64$ holds.

Finally, note that our proof that $\widehat{\text{cov}}(x, z)$ lies within multiplicative error $d/64$ of $\text{cov}(x, z)$ only used the facts that $|\widehat{\text{cov}}(x, z)| \geq (15c/16)^3$ and that $\widehat{\text{cov}}(x, z)$

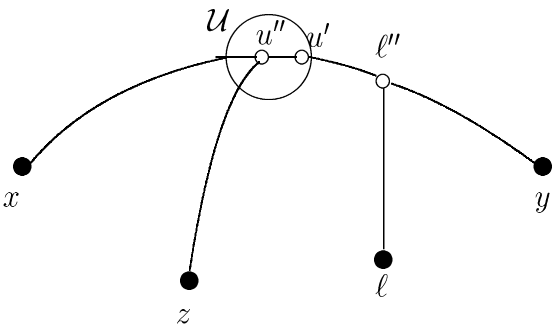


Figure 2.5: $\text{Test}_y(\mathcal{U}, z)$ succeeds for every z that meets the (x, y) path at $\mathcal{U}$.

lies within additive error $c^3d/128$ of its true value. Therefore $\widehat{\text{cov}}(x, y)$ and $\widehat{\text{cov}}(y, z)$ will also lie within multiplicative error $d/64$ of their true values.

Then, calculating $\widehat{\Lambda}_y(x, z, y)$, we have

$$\begin{aligned}
\sqrt{\frac{\widehat{\text{cov}}(x,y)\widehat{\text{cov}}(y,z)}{\widehat{\text{cov}}(x,z)}} &\leq \sqrt{\frac{\text{cov}(x,y)\text{cov}(x,z)(1+d/64)^2}{\text{cov}(x,z)(1-d/64)}} \\
&\leq \Lambda_y(x,z,y)\frac{(1+d/64)}{(1-d/64)} \\
&\leq \Lambda_y(x,z,y)(1+d/16)
\end{aligned}$$

by Inequality 2.29. Likewise, we can show that $\widehat{\Lambda}_y(x, z, y) \geq \Lambda_y(x, z, y)(1-d/32)$, using Inequality 2.30. The proof for $\widehat{\Lambda}_x(x, z, y)$ is identical. $\square$

Lemma 2.39 *Let $\mathcal{U}$ be an internal node in $\widehat{T}(S_\ell)$ on the path between x and y, and suppose that ℓ'' lies to the "y-side" of $\mathcal{U}$ and that $\Lambda(\mathcal{U}, \ell'') \leq (1 - 7d/8)$. Then Test$_y(\mathcal{U}, z)$ succeeds for every good tester z at $\mathcal{U}$.*

Proof: The situation described in the statement is depicted in Figure 2.5. The meeting point of x, y and z is denoted by u''; u' denotes the internal node of $T(S_\ell)$ in $\mathcal{U}$ that is closest to y. Notice that $\Lambda_y(x, z, y)$ is the multiplicative weight for the path between u'' and y. Also, note that $\Lambda(u'', \ell'') \leq \Lambda(u', \ell'')$. We know, by the proof of Observation 2.35, that $\widehat{\Lambda}(\ell'', y)$ lies within multiplicative error $d/64$ of its true value. By Observation 2.38,

$$\begin{aligned}
\frac{\widehat{\Lambda}_y(x,z,y)}{\widehat{\Lambda}(\ell'',y)} &\leq \frac{\Lambda_y(x,z,y)(1+d/16)}{\Lambda(\ell'',y)(1-d/64)} \\
&\leq \Lambda(u'',\ell'')(1+d/16)(1+d/32) \\
&\leq (1-7d/8)(1+d/8) \\
&\leq (1-3d/4)
\end{aligned}$$

where the first step uses the fact that $1/(1-\xi) \leq (1+2\xi)$ for any $\xi \leq 1/2$. $\square$

A symmetric argument shows that the same result holds for Test_x when x and y swap roles.

Lemma 2.40 *Let $\mathcal{U}$ be an internal node in $\widehat{T}(S_\ell)$ on the path between x and y. Suppose that* either

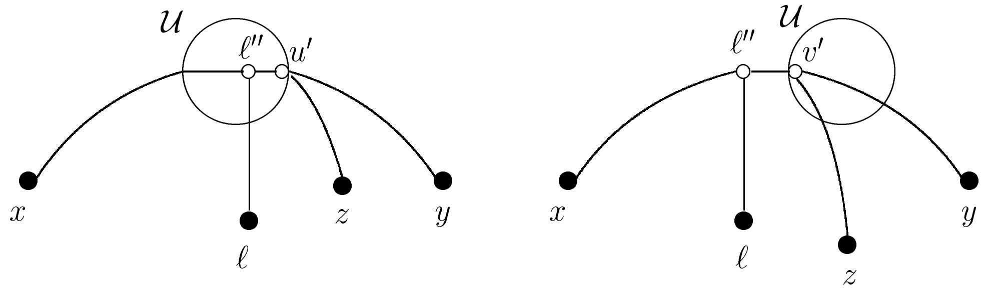
- *ℓ'' lies on some edge of $T(S_\ell)$ that is contracted in the node $\mathcal{U}$,* or
- *that ℓ'' lies to the "x-side" of $\mathcal{U}$.*

Then there is some good tester z such that $Test_y(\mathcal{U},z)$ fails.

Proof: First assume that ℓ'' lies inside $\mathcal{U}$. Let u' be the node of $T(S_\ell)$ that lies in $\mathcal{U}$ and is the closest node to y among all possible candidates. The proof of Observation 2.37 guarantees the existence of a good tester z that meets the path at u' (See Figure 2.6). Then if we perform $\text{Test}_y(\mathcal{U},z)$, we find that

$$\begin{aligned}
\frac{\widehat{\Lambda}_y(x,z,y)}{\widehat{\Lambda}(\ell'',y)} &\geq \frac{\Lambda_y(x,z,y)(1-d/16)}{\Lambda(\ell'',y)(1+d/64)} \\
&= \frac{(1-d/16)}{\Lambda(\ell'',u')(1+d/64)} \\
&\geq (1-d/16)(1-d/32) \\
&\geq (1-d/8)
\end{aligned}$$

where the second step holds because $1/(1+\xi) \geq (1-2\xi)$ for every $\xi \in (0,1)$. Therefore $\text{Test}_y(\mathcal{U},z)$ must fail.



(i) ℓ'' lies inside $\mathcal{U}$ (ii) ℓ'' lies to the "x-side" of $\mathcal{U}$

Figure 2.6: Here are pictures of the two different good testers used in the proof of Lemma 2.40.

For the second case, consider the node v' of $T(S_\ell)$ that lies within $\mathcal{U}$ and is closest to x. Again, Observation 2.37 guarantees that there is a good tester z that meets the path (x, y) at v'. Performing $\text{Test}_y(\mathcal{U}, z)$, we find that

$$\begin{aligned}\frac{\widehat{\Lambda}_y(x,z,y)}{\widehat{\Lambda}(\ell'',y)} &\geq \frac{\Lambda_y(x,z,y)(1-d/16)}{\Lambda(\ell'',y)(1+d/64)} \\ &= \frac{(1-d/16)}{\Lambda(v',\ell'')(1+d/64)} \\ &\geq (1-d/16)(1-d/32) \\ &\geq (1-d/8)\end{aligned}$$

and therefore $\text{Test}_y(\mathcal{U}, z)$ must fail. □

Like Lemma 2.39, the same Lemma can be proven by reversing the roles of x and y.

The next two Lemmas prove that $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$:

Lemma 2.41 *Suppose that the procedure on page 99 adds ℓ to the topology $\widehat{T}(S_\ell)$ by adding the edge $(\mathcal{U}, \ell)$, for some internal node $\mathcal{U}$ of $\widehat{T}(S_\ell)$. Then this new topology is a d-contraction of $T(S_\ell \cup \{\ell\})$.*

Proof: If the algorithm attaches ℓ to $\mathcal{U}$, then $\text{Test}_y(\mathcal{U}, z)$ must have failed for some z. Lemma 2.39 then implies that if ℓ'' does lie to the "y-side" of $\mathcal{U}$, then

$\Lambda(\mathcal{U},\ell'') > (1-7d/8)$. Also, since we stop at the node $\mathcal{U}$, some $\mathrm{Test}_x(\mathcal{U},z)$ must have failed. Using Lemma 2.39 again, this implies that if ℓ'' lies to the "x-side" of $\mathcal{U}$, then $\Lambda(\mathcal{U},\ell'') > (1-7d/8)$. Remember that $\Lambda(\mathcal{U},\ell'')$ is defined as $\Lambda(u',\ell'')$, where $u' \in \mathcal{U}$ is the node of $T(S_\ell)$ such that the path (u',ℓ') in $T(S_\ell \cup \{\ell\})$ does not contain any other nodes of $T(S_\ell)$ that are contracted in $\mathcal{U}$ in $\widehat{T}(S_\ell)$.

Now we will show that if ℓ'' does not lie within the node $\mathcal{U}$ in $\widehat{T}(S_\ell \cup \{\ell\})$, then it must lie on one of the edges adjacent to $\mathcal{U}$. Assume wlog that ℓ'' lies to the "x-side" of $\mathcal{U}$, and suppose that $\mathcal{V}$ is the next node on the "x-side" of $\mathcal{U}$ in $\widehat{T}(S_\ell)$. If $(\mathcal{U},\mathcal{V})$ is a leaf edge, then $\mathcal{V}=x$, and certainly ℓ'' must lie on this edge in $\widehat{T}(S_\ell \cup \{\ell\})$. Otherwise, by Observation 2.32, we know that $\Lambda(\mathcal{U},\mathcal{V}) \leq (1-7d/8)$; in terms of $T(S_\ell)$, if $u \in \mathcal{U}$ and $v \in \mathcal{V}$ are the two nodes of $T(S_\ell)$ such that (u,v) is an edge in $T(S_\ell)$, then $\Lambda(u,v) \leq (1-7d/8)$. Then, since ℓ'' lies to the "x-side" of u, and since v lies on the path from u to x, either ℓ'' lies on the edge (u,v) of $T(S_\ell)$, or ℓ'' lies to the "x-side" of v. However, if ℓ'' lay to the "x-side" of v, then Equation 2.26 would imply that $\Lambda(\ell'',u) = \Lambda(\ell'',v)\Lambda(v,u)$. Now, we know that if ℓ'' lies to the "x-side" of $\mathcal{U}$, then $\Lambda(\ell'',u) > (1-7d/8)$. Also, we know that $\Lambda(u,v) \leq (1-7d/8)$. Therefore if ℓ'' lay to the "x-side" of v, then Equation 2.11 and Observation 2.6 would imply that $\Lambda(\ell'',u) \leq (1-7d/8)$, which is a contradiction. So, if ℓ' lies to the "x-side" of $\mathcal{U}$, then it must lie on an edge adjacent to $\mathcal{U}$. A symmetric argument shows that if ℓ' lies to the "y-side" of $\mathcal{U}$, then it must lie on an edge adjacent to $\mathcal{U}$.

Now we turn to the question of showing that the new topology is a d-contraction of $T(S_\ell \cup \{\ell\})$. First suppose that ℓ'' lies on the adjacent edge $\mathcal{E} = (\mathcal{U},\mathcal{V})$ on the path (x,y) in $\widehat{T}(S_\ell)$, and assume wlog that ℓ'' lies to the "x-side" of $\mathcal{U}$. Let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ be the nodes of $T(S_\ell)$ such that (u,v) is an edge in $T(S_\ell)$. We have already shown that $\Lambda(u,\ell'') > (1-7d/8)$. We will now complete the proof for this case by showing that in this situation, ℓ'' must equal ℓ'. Then, by attaching ℓ to $\mathcal{U}$, we are contracting the edge (u, ℓ'), and since $\Lambda(u, \ell') > (1 - 7d/8)$, $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$. To show that ℓ'' must equal ℓ' in this situation, remember from Corollary 2.29 that ℓ' has degree at least three in $T(S_\ell \cup \{\ell\})$. Therefore, ℓ' must lie along some path of $T(S_\ell)$. Also, since ℓ'' is defined as the node where x, y and ℓ meet in $T(S_\ell \cup \{\ell\})$, therefore ℓ'' is also the point where x, y and ℓ' meet in $T(S_\ell \cup \{\ell\})$. Also, ℓ' lies on some edge of $T(S_\ell)$. So if ℓ'' was not the node ℓ', then ℓ'' would have degree at least 3 in $T(S_\ell)$. This is a contradiction (we assumed that ℓ'' lay on the edge (u, v) of $T(S_\ell)$). Therefore, in this case, $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$.

If ℓ'' lies along an edge that is already contracted in $\mathcal{U}$ (or if ℓ'' is a node in $T(S_\ell)$ contained in $\mathcal{U}$), there are two possible scenarios. It may be the case that ℓ' also lies along an edge of $T(S_\ell)$ contracted in $\mathcal{U}$, and in this situation, adding ℓ to $\mathcal{U}$ certainly creates a d-contraction of $T(S_\ell \cup \{\ell\})$. Otherwise there is an edge of $\widehat{T}(S_\ell)$ that does *not* lie on the path (x, y) but which is adjacent to $\mathcal{U}$, such that ℓ' lies on this edge in $T(S_\ell \cup \{\ell\})$. (ℓ' must lie on an edge of $\widehat{T}(S_\ell)$ that is adjacent to the (x, y) path for the following reason: condition (iv) from Step 2 guarantees that $\Lambda(\ell', \ell'') \geq (1 - d/32)$. Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ be the adjacent edge to $\mathcal{U}$ such that ℓ' either lies on $\mathcal{E}$ or to the "$\mathcal{V}$-side" of $\mathcal{E}$. If ℓ' lay to the "$\mathcal{V}$-side" of $\mathcal{E}$, then using Equation 2.26 and Observation 2.6, we could show that $\Lambda(\ell'', \ell') \leq (1 - 7d/8)$, which is a contradiction. Let $u' \in \mathcal{U}$ be the node of $T(S_\ell)$ such that (u', ℓ') is an edge in $T(S_\ell \cup \{\ell\})$. Therefore by attaching ℓ to $\widehat{T}(S_\ell)$ by the edge $(\mathcal{U}, \ell)$, we are contracting the edge (u', ℓ'). Also, by Equation 2.26 and Observation 2.6, $\Lambda(u', \ell') \geq (1 - d/32)$. So $\widehat{T}(S_\ell \cup \{\ell\})$ is a d-contraction of $T(S_\ell \cup \{\ell\})$. □

Lemma 2.42 *Suppose that the procedure on page 99 adds ℓ to the topology $\widehat{T}(S_\ell)$ by inserting a new node into the middle of the edge $\mathcal{E}$ and attaching ℓ to this new* *node. Then this topology is a d-contraction of* $T(S_\ell \cup \{\ell\})$.

Proof: Let the two endpoints of $\mathcal{E}$ in $\widehat{T}(S_\ell)$ be $\mathcal{U}_1$ and $\mathcal{U}_2$, and assume wlog that $\mathcal{U}_2$ is closer to y than $\mathcal{U}_1$ is. Here are two obvious facts: if $\mathcal{U}_2$ is the leaf y, then ℓ'' definitely lies to the "x-side" of $\mathcal{U}_2$; also, if $\mathcal{U}_1$ is the leaf x, then ℓ'' definitely lies to the "y-side" of $\mathcal{U}_1$.

If $\mathcal{U}_2$ is not the leaf y, then $\text{Test}_x(\mathcal{U}_2, z)$ must have succeeded for every z, or otherwise the algorithm would not have moved past this node to $\mathcal{U}_2$. Then, by Lemma 2.40, ℓ'' must lie to the "x-side" of $\mathcal{U}_2$. Also, if $\mathcal{U}_1$ is not the leaf x, then $\text{Test}_y(\mathcal{U}_1, z)$ must have succeeded for every z, or otherwise the algorithm would have attached ℓ to $\mathcal{U}_1$. By Lemma 2.40, ℓ'' must lie to the "y-side" of $\mathcal{U}_1$. Since we know that ℓ'' lies to the "y-side" of $\mathcal{U}_1$ and to the "x-side" of $\mathcal{U}_2$, therefore ℓ'' must lie somewhere on the edge $(\mathcal{U}_1,\mathcal{U}_2)$. Then $\ell'' = \ell'$, and therefore the new topology is a d-contraction of $T(S_\ell \cup \{\ell\})$. □

Taken together, Lemma 2.41 and Lemma 2.42 prove that the procedure of Step 3 (a), described on page 99, constructs a topology $\widehat{T}(S_\ell \cup \{\ell\})$ that is a d-contraction of $T(S_\ell \cup \{\ell\})$. In the section of this chapter dealing with Step 1 of the algorithm, Lemma 2.33 proved that if we add ℓ to $\widehat{T}(S)$ at the same edge or node where ℓ was added to $\widehat{T}(S_\ell)$, we obtain a d-contraction of $T(S \cup \{\ell\})$. Step 3 (b) describes how to add ℓ to $\widehat{T}(S)$ so that the invariant is maintained.

Step 3 (b): Maintaining the invariant

After the leaf ℓ is added to an edge or a node of $\widehat{T}(S_\ell)$ to give $\widehat{T}(S_\ell \cup \{\ell\})$, Lemma 2.33 implies that we can attach ℓ to the same node or edge of $\widehat{T}(S)$ to obtain a topology that is a d-contraction of $T(S \cup \{\ell\})$. Let $T'(S \cup \{\ell\})$ be this new topology. We now show how to modify $T'(S \cup \{\ell\})$ to obtain another topology $\widehat{T}(S \cup \{\ell\})$ such that $\widehat{T}(S \cup \{\ell\})$ is a d-contraction of $T(S \cup \{\ell\})$ and every edge $\mathcal{E}$ of $\widehat{T}(S \cup \{\ell\})$ satisfies $\Lambda(\mathcal{E}) \leq (1 - 7d/8)$. Then $\widehat{T}(S \cup \{\ell\})$ will satisfy the invariant stated at the beginning of Subsection 2.4.2. Therefore, we prove Theorem 2.16.

By construction, the set $S \cup \{\ell\}$ is a related set for the threshold c. For every internal edge $\mathcal{E}$ in $T'(S \cup \{\ell\})$, let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ denote the two nodes of $T(S \cup \{\ell\})$ such that (u, v) is an edge of $T(S \cup \{\ell\})$. Then if we perform the procedure described in the statement of Lemma 2.24, we will obtain an estimate $\widehat{\Lambda}(\mathcal{E})$ of $\Lambda(\mathcal{E})$ such that $\widehat{\Lambda}(\mathcal{E})$ lies within multiplicative error $d/32$ of its true value in $T'(S \cup \{\ell\})$ (in $T'(S \cup \{\ell\})$, $\Lambda(\mathcal{E})$ is $\Lambda(u, v)$). Therefore,

$$\widehat{\Lambda}(\mathcal{E}) \in [(1 - d/32)\Lambda(u, v), (1 + d/32)\Lambda(u, v)]$$

Now define the topology $\widehat{T}(S \cup \{\ell\})$ as follows: consider all the internal edges of $T'(S \cup \{\ell\})$ in any order. For each internal edge $\mathcal{E}$, we will identify the two endpoints of $\mathcal{E}$ iff $\widehat{\Lambda}(\mathcal{E}) \geq (1 - 15d/16)$. In terms of $T(S \cup \{\ell\})$, identifying the endpoints of $\mathcal{E}$ means that the edge (u, v) of $T(S \cup \{\ell\})$ will be contracted in $\widehat{T}(S \cup \{\ell\})$. Let $\widehat{T}(S \cup \{\ell\})$ be the topology obtained from $T'(S \cup \{\ell\})$ after $\Lambda(\mathcal{E})$ has been estimated for every internal edge $\mathcal{E}$ of $T'(S \cup \{\ell\})$, and some of these edges have been contracted.

To show that the topology $\widehat{T}(S \cup \{\ell\})$ satisfies the invariant stated at the beginning of Subsection 2.4.2, first suppose that we identify the two endpoints $\mathcal{U}$ and $\mathcal{V}$ of some edge $\mathcal{E}$. Then we must have had $\widehat{\Lambda}(\mathcal{E}) \geq (1 - 15d/16)$. Then since $\widehat{\Lambda}(\mathcal{E})$ lies within multiplicative error $d/32$ of its true value, we have $(1 + d/32)\Lambda(u, v) \geq (1 - 15d/16)$ and therefore $\Lambda(u, v) \geq (1 - 7d/8)/(1 + d/32) \geq (1 - 7d/8)(1 - d/16) \geq (1 - 7d/8 - d/16) > (1 - d)$. Therefore, we only contract an edge (u, v) of $T(S \cup \{\ell\})$ if $\Lambda(u, v) \geq (1 - d)$. So $\widehat{T}(S \cup \{\ell\})$ will also be a d-contraction of $T(S \cup \{\ell\})$. Suppose that the internal edge $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ is not contracted in $\widehat{T}(S \cup \{\ell\})$. Then we must have had $\widehat{\Lambda}(\mathcal{E}) < (1 - 15d/16)$, and therefore $\Lambda(u,v)(1-d/32) < (1-15d/16)$. Then $\Lambda(u,v) < (1-15d/16)/(1-d/32) \leq (1-15d/16)(1+d/16) \leq (1-7d/8)$. So every edge (u,v) of $T(S \cup \{\ell\})$ that is not contracted in $\widehat{T}(S \cup \{\ell\})$ satisfies $\Lambda(u,v) \leq (1-7d/8)$. Therefore, we have proved Theorem 2.16.

2.4.3 Note: relationship to previous research

Finally, we will relate our research to previous research on "getting the topology". The papers of Erdös et al. [23, 24] and Csűrös and Kao [16, 17] present results on the problem of recovering the unrooted topology of an evolutionary tree in various restricted models of evolution (see Subsection 2.1.3 for more details). However, the results all depend on the existence of a multiplicative weighting W on the edges of the tree, where W is defined for that specific model and W satisfies Equation 2.4. Most of the results in these papers (for different models) show how to reconstruct the topology of a tree, under the assumption that there are two values a and b such that $0 < a \leq b < 1$ and that $\mathrm{W}(e) \in [a,b]$ for every edge in the tree. Remember that in the Two-State General Markov Model, the multiplicative weight is denoted by $\Lambda(e)$, for every edge e.

Now suppose that M is a Two-State MET with the topology T such that $\Lambda(e) \in [a,b]$ for some $0 < a \leq b < 1$. If we define the depth of a tree, denoted by $depth(M)$, in the same way as in Subsection 2.1.3, and define a leaf connectivity graph on the leaves of M by using the *exact* covariances, then the entire graph is connected for the threshold $a^{2depth(M)+2}$. If we are given *estimates* of covariances that lie within additive error $(a^{2depth(M)+2}/2)^3((1-b)/2)/128$ of their true values, then using the threshold $a^{2depth(M)+2}/2$, the set of leaves forms a related set in this graph. By Theorem 2.16, our algorithm will construct a $(1-b)/2$-contraction $\widehat{T}$ of the true tree. Then the edge e is contracted in $\widehat{T}$ if and only if $\Lambda(e) \geq (1+b)/2$. However, $(1+b)/2 > b$ when $b < 1$, and therefore *every* edge of M satisfies $\Lambda(e) < (1+b)/2$. Therefore the $(1-b)/2$-contraction is the true topology of M.

Using Chernoff bounds, Lemma 2.2 shows that we only need to take

$$O\left(\frac{\log(n/\delta)}{a^{12(depth(M)+1)}(1-b)^2}\right) \tag{2.34}$$

samples from M to ensure that with probability at least $(1-\delta)$, all our covariance estimates lie within additive error $(a^{2depth(M)+2}/2)^3((1-b)/2)/128$ of their true values. Then, with probability at least $(1-\delta)$, our algorithm constructs the original (unrooted) topology of M.

Although we were interested in reconstructing a d-contraction for restricted Two-State METs, the results of Section 2.4 only depend on three things: (1) The assumptions that we made about the closeness of our estimates and the connectivity of the graph; (2) The multiplicative properties of covariances in relation to the Λ weights; (3) The fact that $\Lambda(e) \in (0,1]$ for every edge e. It has already been shown that when $\mathrm{W}(e)$ is defined as $(1-2p_e)$ for the Cavender-Farris-Neyman model, then these weights satisfy conditions (2) and (3) for the inter-leaf distances $(1-2\Pr(x \neq y))$. If we assume that $(1-2p_e) \in [a,b]$ for every edge of a Cavender-Farris tree, then the number of samples described in Expression 2.34 can be used to obtain estimates of $(1-2\Pr(x \neq y))$ that lie within additive error $(a^{2depth(M)+2}/2)^3((1-b)/2)/128$, and our algorithm will reconstruct the topology of any Cavender-Farris-Neyman tree satisfying these conditions.

2.5 Labelling the topology of a related set

Remember that in Section 2.2, Lemma 2.2 allowed us to estimate all of the covariances among the leaves in the original Two-State MET M to within additive error ϵ_4. Remember that ϵ_4 was defined as $(\epsilon_2/2)^3\epsilon_3/2^7$. Now suppose that C is one of the maximal related sets that we constructed from the leaf connectivity graph defined with the threshold $\epsilon_2/2$ in Section 2.2. Then, if c is $\epsilon_2/2$, and d is ϵ_3, the connectivity graph for C satisfies the assumptions that are necessary for the topology algorithm in Section 2.4. Then, by Theorem 2.16, the algorithm in Section 2.4 will construct a topology $\widehat{T}(C)$ such that $\widehat{T}(C)$ is a ϵ_3-contraction of $T(C)$. Therefore we can assume that we know the ϵ_3-contraction $\widehat{T}(C)$.

Now suppose that we are given estimates of the joint distribution on every three leaves x, y, z of C, such that each estimated probability lies within additive error $\epsilon_5/32$ of its true value. Remember that we showed how to estimate the $\Pr(xyz = i_1 i_2 i_3)$ probabilities within additive error $\epsilon_5/32$ on page 50. We will show how to construct a Two-State MET $\widehat{M}(C)$ on the topology $\widehat{T}(C)$ such that every parameter of this MET lies within additive error ϵ_1 of the corresponding parameter in $M'(C)$, where $M'(C)$ is some Two-State MET that generates the original distribution on the leaves in C.

First note that every MET defined on $\widehat{T}(C)$ can be interpreted as a labelling of $T(C)$ in the following way. Let $\mathcal{R}$ be the root of $\widehat{M}(C)$. It is easy to check that the following labelling on $T(C)$ generates the same distribution as $\widehat{M}(C)$: Let $T(C)$ be rooted at any node r that lies within $\mathcal{R}$ in $\widehat{T}(C)$, and define the probability at r to be the probability of $\mathcal{R}$ in $\widehat{M}(C)$. For every edge e that is contracted in $\widehat{T}(C)$, define $e_0 = 0$ and $e_1 = 0$. For every edge e that corresponds to some edge $\mathcal{E}$ in $\widehat{T}(C)$, let $e_0 = \mathcal{E}_0$ and $e_1 = \mathcal{E}_1$.

The first step in constructing a labelling on $\widehat{T}$ uses the *leaf connectivity graph* on C. This is the graph defined for the threshold $\epsilon_2/2$ in Section 2.2, using estimates for the interleaf covariances that lay within additive error ϵ_4 of their true values. Note that since $\epsilon_4 = (\epsilon_2/2)^3(\epsilon_3/2^7)$, clearly $\epsilon_4 < \epsilon_2/2$. Therefore every pair of leaves that satisfies $|\widehat{\mathrm{cov}}(x, y)| \geq \epsilon_2/2$ has the same sign as its true value. Note that Equation 2.10 implies that in any good labelling with good leaves C_1' and bad leaves C_2', $\mathrm{cov}(x, y) > 0$ iff $x, y \in C_1'$ or $x, y \in C_2'$ holds. We can partition C into C_1' and C_2' by using the estimated covariances that have their original signs: first choose any leaf x from the tree and add this to C_1. Then we will add leaves to C_1 and C_2 in the following way, until every leaf in C lies in one of the sets. For every leaf x that does not lie in either set, if $\widehat{\mathrm{cov}}(x, c) \geq \epsilon_2/2$ holds for some $c \in C_1 \cup C_2$, we add x to the set containing c, and if $\widehat{\mathrm{cov}}(x, c) \leq -\epsilon_2/2$ holds for some $c \in C_1 \cup C_2$, add x to the set that does not contain c. Since the leaf connectivity graph forms a related set, this process will terminate after at most n rounds. Also, since all the covariances that we used have their true signs, C_1 and C_2 are the sets C_1' and C_2', although we won't know whether $C_1 = C_1'$ or $C_1 = C_2'$. However, by Observation 2.15, there is a good labelling on $T(C)$ that generates $M(C)$ such that all the leaves in C_1 are good and all the leaves in C_2 are bad.

Next, choose some internal node $\mathcal{R}$ to serve as the root of $\widehat{T}(C)$. Let $M'(C)$ be a labelling of $T(C)$ that generates $M(C)$, such that its root is r for some $r \in \mathcal{R}$ and such that the leaf edges for C_1 are good and the leaf edges for C_2 are bad (by Observation 2.9 and Observation 2.15, we know that this labelling must exist). We now show how to label the edges of $\widehat{M}(C)$ so that every parameter of $\widehat{M}(C)$ is within additive error ϵ_1 of the corresponding value in $M'(C)$. First we prove that the probabilities on the edges that were contracted in $\widehat{T}$ are small.

Observation 2.43 *Let e be an edge in $T(C)$ that is contracted in $\widehat{T}(C)$. Then, for* any *good labelling $M'(C)$ of $T(C)$ that generates $M(C)$, $e_0' \leq \epsilon_1$ and $e_1' \leq \epsilon_1$.*

Proof: By construction, we know that $\Lambda(e) \geq 1 - \epsilon_3$. By Lemma 2.7, we know that $|1 - e_0 - e_1| \geq 1 - 2\epsilon_3$. Since e is an internal edge and $M'(C)$ is a good labelling of $T(C)$, we find $e_0 + e_1 \leq 2\epsilon_3$. Clearly both e_0 and e_1 are less than ϵ_1.

$\square$

First we will show how to estimate the transition probabilities and the root probability of a path in $T(C)$. The root probability for $\mathcal{R}$ will be any of the estimates we obtain for the root probability of a path from $\mathcal{R}$ to a leaf. The following Lemma shows that this lies within $\pm\epsilon_1$ of $\Pr(r = 1)$ for some $r \in \mathcal{R}$, as required. We will then show how to use the estimates for paths to obtain transition probabilities for the leaf edges and internal edges of $\widehat{T}(C)$.

2.5.1 Estimating the transition probabilities for paths

Let x, y, z be three leaves from C such that

(1) The path from x to y passes through $\mathcal{U}$ in $\widehat{T}(C)$;

(2) $|\widehat{\text{cov}}(x,y)| \geq 31(\epsilon_2/2)/32$, and $|\widehat{\text{cov}}(y,z)|$ and $|\widehat{\text{cov}}(x,z)|$ are both at least $(31(\epsilon_2/2)/32)^2$

Let u' be the node in $T(C)$ where x, y and z meet, and assume that u' lies within $\mathcal{U}$ in $\widehat{T}(C)$. Then, assuming that the path (u', y) is the directed path $(u' \to y)$ in $T(C)$, we denote the transition probabilities on the path $(u' \to y)$ by p_0 and p_1.

Lemma 2.44 *Suppose x, y, z are three leaves satisfying conditions (1) and (2) above. Using our estimate of the joint distribution on these three leaves, we can obtain $\widehat{p}_0$, $\widehat{p}_1$ and $\widehat{\Pr}(u' = 1)$ so that $\widehat{\Pr}(u' = 1)$ lies within additive error ϵ_1 of its true value and $\widehat{p}_0$ and $\widehat{p}_1$ lie within additive error $\epsilon_1\epsilon_2/16$ of their true values. Also, we guarantee that $(1 - \widehat{p}_0 - \widehat{p}_1)$ lies within multiplicative error $\epsilon_1\epsilon_2/16$ of its true value.*

Proof: First, note that because we have assumed (from Section 2.2) that all of our covariance estimates lie within additive error ϵ_4 of their true values, then for any three leaves whose covariance estimates satisfy conditions (1) and (2) above, we can show that

$$\begin{aligned} |\mathrm{cov}(x,y)| &\geq 15(\epsilon_2/2)/16 \\ |\mathrm{cov}(y,z)| &\geq (15(\epsilon_2/2)/16)^2 \\ |\mathrm{cov}(x,z)| &\geq (15(\epsilon_2/2)/16)^2 \end{aligned}$$

From the observed distribution, we define

$$\widehat{\mathrm{cov}}(x,z,0) = \widehat{\Pr}(xyz = 101)\widehat{\Pr}(y = 0) - \widehat{\Pr}(xy = 10)\widehat{\Pr}(zy = 10)$$

Now, since we have an estimate $\widehat{\Pr}(xyz = i_1 i_2 i_3)$ that lies within additive error $\epsilon_5/32$ (see page 50) for every $i_1 i_2 i_3 \in \{0,1\}^3$, therefore each of $\widehat{\Pr}(xyz = 101)$, $\widehat{\Pr}(y = 0)$, $\widehat{\Pr}(xy = 10)$ and $\widehat{\Pr}(zy = 10)$ lies within additive error $\epsilon_5/8$ of its true value. Therefore the estimate $\widehat{\mathrm{cov}}(x,z,0)$ lies within additive error ϵ_5 of $\mathrm{cov}(x,z,0)$. We can also define $\widehat{\mathrm{cov}}(x,z,1)$ so that this estimate lies within additive error ϵ_5 of its true value. Also, if we use re-estimate $\widehat{\mathrm{cov}}(x,z)$ by defining

$$\widehat{\mathrm{cov}}(x,z) =_{def} \widehat{\Pr}(xz = 11) - \widehat{\Pr}(x = 1)\widehat{\Pr}(z = 1)$$

with the new estimates of $\widehat{\Pr}(xz = 11)$, $\widehat{\Pr}(x = 1)$ and $\widehat{\Pr}(z = 1)$ (that lie within additive error $\epsilon_5/8$ of their true values), then this $\widehat{\mathrm{cov}}(x,z)$ lies within additive error ϵ_5 of its true value.

The key to estimating the transition probabilities on the path $(u' \to y)$ is estimating the quantities F and D (see Subsection 2.3.3) for the triplet x, y, z. We will define $\widehat{F}$ and $\widehat{D}$ by substituting our estimates $\widehat{\mathrm{cov}}(x,z)$, $\widehat{\mathrm{cov}}(x,z,0)$ and $\widehat{\mathrm{cov}}(x,z,1)$ into Equations 2.14 and 2.15 respectively, with one exception: if $\widehat{\mathrm{cov}}(x,z) + \widehat{\mathrm{cov}}(x,z,0) - \widehat{\mathrm{cov}}(x,z,1)$ and $\widehat{\mathrm{cov}}(x,z)$ have different signs, we will define $\widehat{F}$ to be 0. We will now show that the closeness of our estimates ensures that

$$\widehat{F} \text{ lies within additive error } (\epsilon_1 \epsilon_2^3/2^{11}) \text{ of its true value, and}$$

$\widehat{D}$ lies within additive error $(\epsilon_1\epsilon_2^3/2^9)$ of its true value.

The most important part of this Lemma is showing that $\widehat{F}$ satisfies the bounds described above.

First assume that $\mathrm{cov}(x,z)$ is positive. We will deal with the exception first. Note that since we assumed that $\mathrm{cov}(x,z) \geq (15(\epsilon_2/2)/16)^2$ and since we have calculated an estimate $\widehat{\mathrm{cov}}(x,z)$ that lies within additive error ϵ_5 of its true value, we can assume that $\widehat{\mathrm{cov}}(x,z)$ is also positive. However, suppose that $\widehat{\mathrm{cov}}(x,z) + \widehat{\mathrm{cov}}(x,z,0) - \widehat{\mathrm{cov}}(x,z,1)$ is negative. We know that $\widehat{\mathrm{cov}}(x,z) + \widehat{\mathrm{cov}}(x,z,0) - \widehat{\mathrm{cov}}(x,z,1)$ lies within additive error $3\epsilon_5$ of its true value. By Equation 2.17, $\mathrm{cov}(x,z) + \mathrm{cov}(x,z,0) - (x,z,1)$ is equal to $\mathrm{cov}(x,z)(1+p_1-p_0)$, where p_0 and p_1 are the transition probabilities along $(u' \to y)$. Since the expression $\widehat{\mathrm{cov}}(x,z)+\widehat{\mathrm{cov}}(x,z,0)-\widehat{\mathrm{cov}}(x,z,1)$ is negative, therefore $\mathrm{cov}(x,z)(1+p_1-p_0) \leq 3\epsilon_5$. We will finish this bound by finding an upper bound for $3\epsilon_5/\mathrm{cov}(x,z)$. We know that $\mathrm{cov}(x,z) \geq (15(\epsilon_2/2)/16)^2$, so $3\epsilon_5/\mathrm{cov}(x,z) \leq 3\epsilon_5(16(2/\epsilon_2)/15)^2$. By definition, $\epsilon_5 = (\epsilon_2/2)^2(\epsilon_4/4)$. Therefore, $3\epsilon_5/\mathrm{cov}(x,z) \leq 3(\epsilon_2/2)^2(\epsilon_4/4)(16(2/\epsilon_2)/15)^2$, so $3\epsilon_5/\mathrm{cov}(x,z) \leq (3\epsilon_4/4)(16/15)^2$, which is less than ϵ_4. Then $(1+p_1-p_0) \leq \epsilon_4$. By Equation 2.17, $F = (1+p_1-p_0)/2$. Therefore, in this case, by defining $\widehat{F}$ to be 0, our estimate for F lies within additive error $\epsilon_4/2$ of its true value.

Now consider the general case for $\widehat{F}$. By the closeness of our estimates, we have

$$\begin{aligned}
\widehat{F} &\leq \frac{1}{2}\left(\frac{\mathrm{cov}(x,z)+\mathrm{cov}(x,z,0)-\mathrm{cov}(x,z,1)+3\epsilon_5}{\mathrm{cov}(x,z)-\epsilon_5}\right) \\
&\leq \frac{1}{2}\left(2F + \frac{3\epsilon_5}{\mathrm{cov}(x,z)-3\epsilon_5}(1+2F)\right)
\end{aligned}$$

by Equation 2.28. Remember that $3\epsilon_5/\mathrm{cov}(x,z) \leq \epsilon_4$. Then, writing $\mathrm{cov}(x,z) - 3\epsilon_5$ as $\mathrm{cov}(x,z)(1-3\epsilon_5/\mathrm{cov}(x,z))$, we have $\mathrm{cov}(x,z) - 3\epsilon_5 \geq \mathrm{cov}(x,z)(1-\epsilon_4)$. So

$$\widehat{F} \leq F + \frac{3\epsilon_5}{2\mathrm{cov}(x,z)(1-\epsilon_4)}(1+2F)$$

Also, because $1/(1-\xi) \leq (1+2\xi)$ for every $\xi < 1/2$, we can show that

$$\widehat{F} \quad \leq \quad F + \frac{3\epsilon_5}{2\text{cov}(x,z)}\,(1+2\epsilon_4)\,(1+2F)$$

Using $3\epsilon_5/\text{cov}(x,z) \leq \epsilon_4$ again, and using the fact that $F \leq 1$, we find that $\widehat{F} \leq F + 2\epsilon_4$. To bound $\widehat{F}$ from below, note that under the assumption that $\widehat{\text{cov}}(x,z) + \widehat{\text{cov}}(x,z,0) - \widehat{\text{cov}}(x,z,1)$ is positive,

$$\widehat{F} \quad \geq \quad \frac{1}{2}\left(\frac{\text{cov}(x,z) + \text{cov}(x,z,0) - \text{cov}(x,z,1) - 3\epsilon_5}{\text{cov}(x,z) + \epsilon_5}\right)$$

Then, because $\epsilon_5/\text{cov}(x,z) \leq \epsilon_4/2$, we have

$$\begin{aligned}\widehat{F} \quad &\geq \quad \frac{1}{2}\left(\frac{\text{cov}(x,z) + \text{cov}(x,z,0) - \text{cov}(x,z,1) - 3\epsilon_5}{\text{cov}(x,z)(1+\epsilon_4/2)}\right) \\ &\geq \quad \frac{1}{2}\left(2F - \frac{3\epsilon_5}{\text{cov}(x,z)}\right)(1-\epsilon_4)\end{aligned}$$

using the fact that $1/(1+\xi) \leq (1-2\xi)$ for every $\xi < 1/2$. Then, in the same way as before, we find that $\widehat{F} \geq F - 2\epsilon_4$. So in the general case, $\widehat{F} \in (F - 2\epsilon_4, F + 2\epsilon_4)$. Also, for the special case when we define $\widehat{F}$ to be 0, $\widehat{F}$ easily satisfies this bound. Then expanding ϵ_4 out as $\epsilon_3(\epsilon_2/2)^3/2^7$, and using the fact that $\epsilon_3 \leq \epsilon_1/4$, we can replace ϵ_4 in the inequality above to give

$$\widehat{F} \quad \in \quad \left[F - \frac{\epsilon_1\epsilon_2^3}{2^{11}}, F + \frac{\epsilon_1\epsilon_2^3}{2^{11}}\right] \tag{2.35}$$

Alternatively, suppose that $\text{cov}(x,z) < 0$. Then Equations 2.14 and 2.17 show that the sign of $\text{cov}(x,z) + \text{cov}(x,z,0) - \text{cov}(x,z,1)$ is also negative. Therefore, if we take the absolute values of $\widehat{\text{cov}}(x,z) + \widehat{\text{cov}}(x,z,0) - \widehat{\text{cov}}(x,z,1)$ and $\widehat{\text{cov}}(x,z)$ (assuming that they have the same sign) and use these to calculate $\widehat{F}$, we obtain the same estimate as we would have by using the original values. Then using the proofs above (since we know $|\text{cov}(x,z)| \geq (15(\epsilon_2/2)/16)^2$), we can prove the same result when $\text{cov}(x,z) < 0$.

To estimate D, we first show how to estimate $\mathrm{cov}(x,z,0)/\mathrm{cov}(x,z)$. The estimate is obtained in the following way: If $\widehat{\mathrm{cov}}(x,z,0) \leq 0$, then we will redefine $\widehat{\mathrm{cov}}(x,z,0) = 0$, so that $\widehat{\mathrm{cov}}(x,z,0)/\widehat{\mathrm{cov}}(x,z)$ will then equal 0. To justify this, note that if $\widehat{\mathrm{cov}}(x,z,0) \leq 0$, then $\mathrm{cov}(x,z,0) \leq \epsilon_5$. Remember that by Equation 2.16 the value of $\mathrm{cov}(x,z,0) = p_1(1-p_0)\mathrm{cov}(x,z)$. Therefore, in this situation, $p_1(1-p_0)\mathrm{cov}(x,z) \leq \epsilon_5$, and therefore $p_1(1-p_0) \leq \epsilon_5/\mathrm{cov}(x,z)$. We know from beforehand that $\epsilon_5/\mathrm{cov}(x,z) \leq \epsilon_4/2$, so our new estimate lies within additive error $\epsilon_4/2$ of the true value of $\mathrm{cov}(x,z,0)/\mathrm{cov}(x,z)$. Otherwise, we have

$$\begin{aligned}
\frac{\widehat{\mathrm{cov}}(x,z,0)}{\widehat{\mathrm{cov}}(x,z)} &\leq \frac{\mathrm{cov}(x,z,0)+\epsilon_5}{\mathrm{cov}(x,z)-\epsilon_5} \\
&\leq \frac{\mathrm{cov}(x,z,0)+\epsilon_5}{\mathrm{cov}(x,z)(1-\epsilon_4/2)} \\
&\leq \frac{\mathrm{cov}(x,z,0)+\epsilon_5}{\mathrm{cov}(x,z)}(1+\epsilon_4)
\end{aligned}$$

where the first step follows because we have already shown that $\epsilon_5/\mathrm{cov}(x,z) \leq \epsilon_4$. Then using this fact again we obtain

$$\begin{aligned}
\frac{\widehat{\mathrm{cov}}(x,z,0)}{\widehat{\mathrm{cov}}(x,z)} &\leq \left(\frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} + \epsilon_4/2\right)(1+\epsilon_4) \\
&\leq \frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} + 2\epsilon_4
\end{aligned}$$

Similarly, we can show that $\widehat{\mathrm{cov}}(x,z,0)/\widehat{\mathrm{cov}}(x,z) \geq \mathrm{cov}(x,z,0)/\mathrm{cov}(x,z) - 2\epsilon_4$. Then, expanding ϵ_4 out in terms of ϵ_1 and ϵ_2, we have

$$\frac{\widehat{\mathrm{cov}}(x,z,0)}{\widehat{\mathrm{cov}}(x,z)} \in \left[\frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} - \frac{\epsilon_1\epsilon_2^3}{2^{11}}, \frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} + \frac{\epsilon_1\epsilon_2^3}{2^{11}}\right] \tag{2.36}$$

Now, if we define $\widehat{D}$ to be equal to $\widehat{F}^2 - \widehat{\mathrm{cov}}(x,z,0)/\widehat{\mathrm{cov}}(x,z)$, then

$$\begin{aligned}
\widehat{D} &\leq \left(F + \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\right)^2 - \left(\frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} - \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\right) \\
&\leq D + \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\left(1 + 2F + \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\right) \\
&\leq D + \frac{\epsilon_1{\epsilon_2}^3}{2^9}
\end{aligned}$$

because $F \leq 1$. Also,

$$\begin{aligned}\widehat{D} &\geq \left(F - \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\right)^2 - \left(\frac{\mathrm{cov}(x,z,0)}{\mathrm{cov}(x,z)} + \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}\right) \\ &\geq D - \frac{\epsilon_1{\epsilon_2}^3}{2^{11}}(1+2F) \\ &\geq D - \frac{\epsilon_1{\epsilon_2}^3}{2^9}\end{aligned}$$

So $\widehat{D} \in [D - \epsilon_1{\epsilon_2}^3/2^9, D + \epsilon_1{\epsilon_2}^3/2^9]$. Remember that item (2) on page 113 implies that $|\mathrm{cov}(x,y)| \geq (15(\epsilon_2/2)/16)$. Then, because u' lies on the path between x and y, Equation 2.10 implies that $|1-p_0-p_1| \geq (15(\epsilon_2/2)/16)$. By Equation 2.18, we know that the true value of D is $(1-p_0-p_1)^2/4$. Therefore, the true value of D is at least $\epsilon_2^2/2^5$. Notice that $\widehat{D}$ is within multiplicative error $\epsilon_1\epsilon_2/2^4$ of D. Then, by Inequalities 2.31 and 2.32, $2\sqrt{\widehat{D}}$ lies within multiplicative error $\epsilon_1\epsilon_2/2^4$ of its true value. Also, since the absolute value of $\sqrt{D}$ is at most $1/2$, $\sqrt{\widehat{D}}$ lies within additive error $\epsilon_1\epsilon_2/2^5$ of its true value.

Case 1: $y \in C_1$

We will first assume that $y \in C_1$, and show how to substitute $\widehat{D}$ and $\widehat{F}$ into Equations 2.19 to obtain estimates for p_0 and p_1. We already know that $\sqrt{\widehat{D}}$ lies within additive error $\epsilon_1\epsilon_2/2^5$ of its true value, and Equation 2.35 implies that $\widehat{F}$ satisfies the same bounds. Therefore $\widehat{p}_0$ and $\widehat{p}_1$ lie within additive error $\epsilon_1\epsilon_2/2^4$ of their real values.

The only quantity that remains to be bounded is $\widehat{\Pr}(u'=1)$. Following Equation 2.22, we define

$$\widehat{\Pr}(u'=1) = \frac{2\sqrt{\widehat{D}} + \widehat{F} - \widehat{\Pr}(y=0)}{2\sqrt{\widehat{D}}}$$

Now, because we assumed that every probability of the distribution on x, y, z was estimated to within additive error $\epsilon_5/32$, we can certainly assume that $\widehat{\Pr}(y=0)$ lies within additive error $\epsilon_1\epsilon_2/32$ of its true value. Then, using the bounds that we have already obtained for $2\sqrt{\widehat{D}}$ and $\widehat{F}$, we know that $2\sqrt{\widehat{D}} + \widehat{F} - \widehat{\Pr}(y=0)$ lies within additive error $\epsilon_1\epsilon_2/8$ of $2\sqrt{D} + F - \Pr(y=0)$.

We will note that since $(1 - p_0 - p_1) > 0$ when $y \in C_1$, therefore by Equation 2.22, the true value of $2\sqrt{D} + F - \Pr(y = 0)$ is also positive. Therefore, when $2\sqrt{\widehat{D}} + \widehat{F} - \widehat{\Pr}(y = 0)$ is negative, we will not use Equation 2.22 to define $\widehat{\Pr}(u' = 1)$, but simply define $\widehat{\Pr}(u' = 1)$ to be 0. We will first show that $\widehat{\Pr}(u' = 1)$ is a good estimate when this exception takes place. Note that since $2\sqrt{\widehat{D}} + \widehat{F} - \widehat{\Pr}(y=0)$ lies within additive error $\epsilon_1\epsilon_2/8$ of its real value, therefore it must be the case that $2\sqrt{D} + F - \Pr(y = 0) \leq \epsilon_1\epsilon_2/8$. Then, since we know by Equation 2.22 that $2\sqrt{D} + F - \Pr(y = 0) = 2\sqrt{D}\Pr(u' = 1)$, we find that $2\sqrt{D}\Pr(u' = 1) \leq \epsilon_1\epsilon_2/8$. Now $2\sqrt{D} \geq \epsilon_2/4$, so therefore it must be the case that $\Pr(u' = 1) \leq \epsilon_1/2$. Therefore, for this exception, $\widehat{\Pr}(u' = 1)$ will lie within additive error ϵ_1 of its true value.

Alternatively, when $2\sqrt{\widehat{D}} + \widehat{F} - \widehat{\Pr}(y = 0)$ is positive, we have

$$\widehat{\Pr}(u' = 1) \quad \leq \quad \frac{2\sqrt{D} + F - \Pr(y=0) + \epsilon_1\epsilon_2/8}{2\sqrt{D}(1 - \epsilon_1\epsilon_2/16)}$$

because we already know that $2\sqrt{\widehat{D}}$ lies within multiplicative error $\epsilon_1\epsilon_2/16$ of its true value. Therefore, using the fact that $1/(1-\xi) \leq (1+2\xi)$ when $\xi < 1/2$, we have

$$\begin{aligned}
\widehat{\Pr}(u' = 1) \quad &\leq \quad \frac{2\sqrt{D}\Pr(u'=1) + \epsilon_1\epsilon_2/8}{2\sqrt{D}}(1 + \epsilon_1\epsilon_2/8) \\
&\leq \quad \left(\Pr(u'=1) + \frac{\epsilon_1\epsilon_2/8}{(1 - p_0 - p_1)}\right)(1 + \epsilon_1\epsilon_2/8) \\
&\leq \quad \Pr(u'=1) + \epsilon_1/2(1 + \epsilon_1/2)
\end{aligned}$$

where the last step follows because we know that $(1 - p_0 - p_1) \geq \epsilon_2/4$. We can also show that $\widehat{\Pr}(u' = 1) \geq \Pr(u' = 1) - \epsilon_1$.

Case 2: $y \in C_2$

The proof for this case can be obtained as a corollary of the proof for $y \in C_1$. Let p'_0 and p'_1 be the true transition probabilities along p when $y \in C_2$. Remember that by Equations 2.23,

$$p'_0 = 1 + \sqrt{D} - F \quad \text{and} \quad p'_1 = F + \sqrt{D}$$

and by Equations 2.19, the probabilities p_0 and p_1 (the probabilities if $y \in C_1$) are

$$p_0 = 1 - \sqrt{D} - F \quad \text{and} \quad p_1 = F - \sqrt{D}$$

Therefore

$$p'_0 = 1 - p_1 \quad \text{and} \quad p'_1 = 1 - p_0$$

Therefore, if we always calculate estimates for the transition probabilities by assuming $(1 - p_0 - p_1)$ is positive, we can obtain estimates for any $y \in C_2$ by defining

$$\widehat{p}'_0 = 1 - \widehat{p}_1 \quad \text{and} \quad \widehat{p}'_1 = 1 - \widehat{p}_0$$

These estimates will then lie within the same error bounds as $\widehat{p}_0$ and $\widehat{p}_1$. Also, using Equations 2.22 and 2.24 from Subsection 2.3.3, we see that the probability at u' when $y \in C_2$ is equal to $1 - \Pr(u' = 1)$, where $\Pr(u' = 1)$ denotes the probability at u' when $y \in C_1$. Therefore, if we obtain the probability at u' by assuming $(1-p_0-p_1)$ is positive, and then redefine the probability as $1-\widehat{\Pr}(u' = 1)$ when $y \in C_2$, Equations 2.22 and 2.24 imply that this value is within additive error ϵ_1 of the true probability at u' for $y \in C_2$. $\square$

2.5.2 Estimating the probabilities along a leaf edge

Suppose that $\mathcal{E} = (\mathcal{U}, y)$ in $\widehat{T}(C)$. By Observation 2.25, if u is the node of $T(C)$ such that $e = (u, y)$ is an edge in $T(C)$, then there exists a pair of leaves (x, z) from C that form a $63(\epsilon_2/2)/64$-good estimator of e. By Observation 2.25, this is also a $63(\epsilon_2/2)/64$-apparently good estimator of $\mathcal{E}$. By the definition of a good estimator for a leaf edge, x, z and y must meet at u in $T(C)$, and wlog in choosing between x and z, we have

$$|\mathrm{cov}(x, y)| \geq (63(\epsilon_2/2)/64)$$

and $\Lambda(u, z) \geq (63(\epsilon_2/2)/64)$. Clearly x and z satisfy condition (1) on page 113 for the estimation of a path. Also, by Equation 2.26, we have

$$\begin{aligned} |\mathrm{cov}(x, z)| &\geq (63(\epsilon_2/2)/64)^2 \\ |\mathrm{cov}(y, z)| &\geq (63(\epsilon_2/2)/64)^2 \end{aligned}$$

Also, remember that since our covariance estimates always lie within additive error ϵ_4 throughout the entire thesis, therefore $|\widehat{\mathrm{cov}}(x, y)| \geq (63(\epsilon_2/2)/64) - \epsilon_4 = (63(\epsilon_2/2)/64) - (\epsilon_2/2)^3(\epsilon_3/2^7) \geq (\epsilon_2/2)(63/64 - (\epsilon_2/2)^2(\epsilon_3/2^7))$, and this value will be at least $31(\epsilon_2/2)/32$. Also, each of the two estimated absolute values $|\widehat{\mathrm{cov}}(x, z)|$ and $|\widehat{\mathrm{cov}}(y, z)|$ are lower-bounded by $(63(\epsilon_2/2)/64)^2 - \epsilon_4$, which equals $(63(\epsilon_2/2)/64)^2 - (\epsilon_2/2)^3(\epsilon_3/2^7)$. This is equal to $(\epsilon_2/2)^2((63/64)^2 - (\epsilon_2/2)(\epsilon_3/2^7))$, and since $\epsilon_2 < 1$ and $\epsilon_3 < 1$, this quantity is at least $(\epsilon_2/2)^2((63/64)^2 - 1/2^8) \geq (15(\epsilon_2/2)/16)^2$. Therefore x and z satisfy condition (2) on page 113. Let u' be the node in $T(C)$ where x, y and z meet in $\mathcal{U}$, and let $(u \to y)$ be the edge in $M'(C)$ that corresponds to $\mathcal{E}$. Denote the transition probabilities for the edge $(u \to y)$ by y_0 and y_1 (See Figure 2.7), and the probabilities for $(u' \to y)$ by p_0 and p_1. By Lemma 2.44, we know that we can obtain estimates $\widehat{p}_0$, $\widehat{p}_1$ and $\widehat{\Pr}(u' = 0)$ that lie within additive error $\epsilon_1/2$ of their true values (this is a much weaker

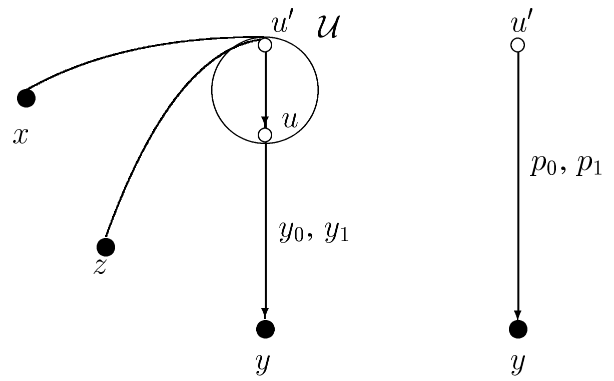


Figure 2.7: x, z is a good estimator for the path $(u' \to y)$.

re-statement of Lemma 2.44). We will define

$$\widehat{y}_0 = \widehat{p}_0 \quad \text{and} \quad \widehat{y}_1 = \widehat{p}_1$$

Since the estimates for the path from u' to y are within additive error $\epsilon_1/2$ of their true values, the following Observation suffices to show that $\widehat{y}_0$ and $\widehat{y}_1$ will lie within additive error ϵ_1 of their true values, as required.

Corollary 2.45 *Let (u', y) be a path in $T(C)$ such that u lies on this path and the edges between u' and u are all contracted in $\widehat{T}(C)$. Denote the probabilities for the path $(u' \to y)$ by p_0 and p_1 and the probabilities for the path $(u \to y)$ by y_0 and y_1. Then $|p_0 - y_0| \leq \epsilon_1/2$ and $|p_1 - y_1| \leq \epsilon_1/2$ both hold.*

Proof: Since the set C has at most n leaves, the path between u' and u can contain at most n edges. By construction, every edge e along the path satisfies $\Lambda(e) \geq (1-\epsilon_3)$, and therefore $\Lambda(u', u) \geq (1-\epsilon_3)^n$. Also, because $\epsilon_3 \leq 1/2$, $(1-\epsilon_3)^n \geq 1 - n\epsilon_3$, and therefore $\Lambda(u', u) \geq \epsilon_1/4$.

Suppose that the transition probabilities for $(u \to u')$ are denoted by f_0 and f_1. By Lemma 2.7, and because the labelling that we construct has good internal edges, $1 - f_0 - f_1 \geq 1 - \epsilon_1/2$ or alternatively, $f_0 + f_1 \leq \epsilon_1/2$. By Observation 2.4, $p_0 - y_0 = f_0(1 - y_0 - y_1)$, so the absolute value of $p_0 - y_0$ can be at most $\epsilon_1/2$. A similar argument shows that the bound on $|p_1 - y_1|$ holds. □

2.5.3 Estimating the probabilities along an internal edge

Let $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ be an internal edge in $\widehat{T}(C)$, and let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ be the two nodes of $T(C)$ such that $e = (u, v)$ is an edge in $T(C)$. By Observation 2.22, there is some $(63c/64)$-good estimator $(w, x \mid y, z)$ of e in C. Then, by Observation 2.19 and using the fact that all our covariance estimates lie within additive error of at least ϵ_4 of their true values, we will have

$$|\widehat{\mathrm{cov}}(x, y)| \geq (31c/32)$$
$$|\widehat{\mathrm{cov}}(w, x)| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(y, z)| \geq (31c/32)^2$$
$$|\widehat{\mathrm{cov}}(x, z)| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(w, y)| \geq (31c/32)^2$$

To estimate the edge probabilities for the edge e that corresponds to $\mathcal{E}$ in $\widehat{T}(C)$, we will choose any quartet $(w, x \mid y, z)$ that satisfies the topological constraints to be an apparently good estimator of $\mathcal{E}$, and whose relevant estimated covariances satisfy the bounds above. Now we will show that we can construct edge probabilities for e that lie within additive error ϵ_1 of their correct values.

Lemma 2.46 *Suppose that $\mathcal{E} = (\mathcal{U}, \mathcal{V})$ is an internal edge of $\widehat{T}(C)$ and let $u \in \mathcal{U}$ and $v \in \mathcal{V}$ be the nodes of $T(C)$ such that $e = (u, v)$ is an edge in $T(C)$. Now suppose that $(w, x \mid y, z)$ is a quartet of $\widehat{T}(C)$ such that the (x,y)-path in $\widehat{T}(C)$ contains the edge $\mathcal{E}$, the nodes x, y and w meet at $\mathcal{U}$ in $\widehat{T}(C)$, and the nodes x, y and z meet at $\mathcal{V}$ in $\widehat{T}(C)$. Assume wlog that w and x lie to the '$\mathcal{U}$-side" of $\mathcal{E}$, and suppose that*

$$|\widehat{\mathrm{cov}}(x, y)| \geq (31c/32)$$
$$|\widehat{\mathrm{cov}}(w, x)| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(y, z)| \geq (31c/32)^2$$

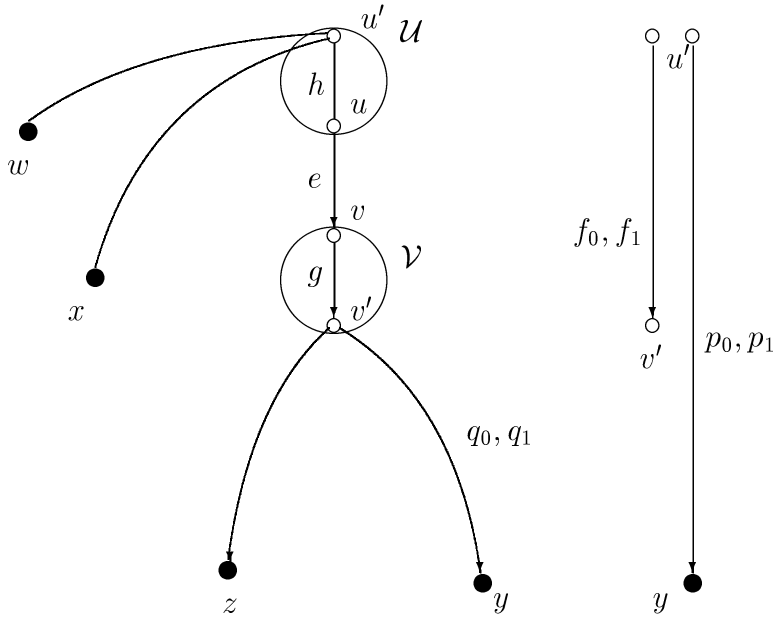


Figure 2.8: The situation in $\widehat{T}(C)$ and in $T(C)$ for estimating the transition probabilities on $(\mathcal{U}, \mathcal{V})$.

$$|\widehat{\text{cov}}(x,z)| \geq (31c/32)^2 \quad |\widehat{\text{cov}}(w,y)| \geq (31c/32)^2$$

holds. Then we can calculate transition probabilities $\widehat{e}_0$ *and* $\widehat{e}_1$ *that lie within additive error* ϵ_1 *of their true values.*

Proof: First note that we assume wlog that the edge $\mathcal{E}$ is directed from $\mathcal{U}$ to $\mathcal{V}$ in the rooted topology $\widehat{T}(C)$. Then we want to estimate the probabilities e_0 and e_1 for $e = (u \to v)$ in the Two-State MET $M'(C)$ (this is the normalized Two-State MET that generates the same distribution as the original Two-State MET $M(C)$). In this proof, we will denote the edge $(u \to v)$ in the MET $M'(C)$ by e and the path $(v \to v')$ in $M'(C)$ from v to v' by g. Now, we know that in $M'(C)$, the tree $T(C)$ is rooted at some node of $T(C)$ that lies "above" u, with respect to the directed path $(u \to z)$. Now consider the topology $M''(C)$ which is obtained by re-rooting the underlying tree $T(C)$ at u' (see Observation 2.9). Notice that since the original root in $T(C)$ lay somewhere above the path $(u \to z)$, changing the root to u' in $M''(C)$ does not change any of the probabilities on the path $(u \to v')$. Therefore, to estimate e_0 and e_1 in $M'(C)$, it is enough to estimate e_0 and e_1 in the MET $M''(C)$. Let h denote the path $(u \to u')$ in the MET $M''(C)$. We will now show how to estimate the values of e_0 and e_1 in the MET $M''(C)$, and therefore obtain estimates for e_0 and e_1 in $M'(C)$, as required. Figure 2.8 describes the scenario in $M''(C)$.

Let p denote the path $(u' \to y)$ and let q denote the path $(v' \to y)$. Now we will show that the triple x, y, w satisfies conditions (1) and (2) on page 113 for the path $p = (u' \to y)$. First notice that the path between x and y contains the node $\mathcal{U}$, and w, x and y meet at $\mathcal{U}$ in $\widehat{T}(C)$. Also, by assumption,

$$|\widehat{\mathrm{cov}}(x,y)| \geq (31c/32) \quad |\widehat{\mathrm{cov}}(w,x)| \geq |(31c/32)^2 \quad |\widehat{\mathrm{cov}}(w,y)| \geq (31c/32)^2$$

which is condition (2) for x, y and w. Therefore, by Lemma 2.44, we can estimate p_0 and p_1 within additive error $\epsilon_1\epsilon_2/16$ of their true values such that $(1 - p_0 - p_1)$ lies within multiplicative error $\epsilon_1\epsilon_2/8$ of its true value. Now we will show that x, y, z satisfies conditions (1) and (2) on page 113 for the path $q = (v' \to y)$. Note that $\mathcal{V}$ lies on the path between x and y, and x, y and z meet at $\mathcal{V}$ in $\widehat{T}(C)$. So condition (1) is satisfied. Also, by assumption,

$$|\widehat{\mathrm{cov}}(x,y)| \geq (31c/32) \quad |\widehat{\mathrm{cov}}(y,z)| \geq (31c/32)^2 \quad |\widehat{\mathrm{cov}}(x,z)| \geq (31c/32)^2$$

which is condition (2) for x, y and z. By Lemma 2.44, we can estimate q_0 and q_1 within additive error $\epsilon_1\epsilon_2/16$ of their true values such that $(1 - q_0 - q_1)$ lies within multiplicative error $\epsilon_1\epsilon_2/8$ of its true value.

Next we use the estimates of p_0, p_1, q_0 and q_1 to define $\widehat{e}_0$ and $\widehat{e}_1$

$$\widehat{e}_0 = \frac{\widehat{p}_0 - \widehat{q}_0}{(1 - \widehat{q}_0 - \widehat{q}_1)} \quad \text{and} \quad \widehat{e}_1 = \frac{\widehat{p}_1 - \widehat{q}_1}{(1 - \widehat{q}_0 - \widehat{q}_1)},$$

mimicking Equations 2.25. We will prove that these estimates lie within additive error ϵ_1 of their true values. Let f be the path from u' to v', and let f_0

and f_1 be the probabilities along this path. When we have the exact values for the transition probabilities on p and q, then Equations 2.25 allow us to reconstruct f_0 and f_1 exactly. First we will show that $\widehat{e}_0$ and $\widehat{e}_1$ lie within additive error $\epsilon_1/2$ of f_0 and f_1 respectively. We will complete the proof by showing that e_0 and e_1 lie within additive error $\epsilon_1/2$ of f_0 and f_1 respectively.

Assume that $(1 - q_0 - q_1) \geq 0$. Also, because $|\mathrm{cov}(x, y)| \geq (15(\epsilon_2/2)/16)$, Equation 2.10 implies that $(1 - q_0 - q_1) \geq (15(\epsilon_2/2)/16)$. Then

$$\begin{aligned}
\widehat{e}_0 &\leq \frac{p_0 - q_0 + \epsilon_1\epsilon_2/8}{(1 - q_0 - q_1)(1 - \epsilon_1\epsilon_2/16)} \\
&\leq \left(f_0 + \frac{\epsilon_1\epsilon_2/8}{(1 - q_0 - q_1)}\right)(1 + \epsilon_1\epsilon_2/8) \\
&\leq (f_0 + 2\epsilon_1/3)(1 + \epsilon_1\epsilon_2/8) \\
&\leq f_0 + \epsilon_1/2
\end{aligned}$$

Also,

$$\begin{aligned}
\widehat{e}_0 &\geq \frac{p_0 - q_0 - \epsilon_1\epsilon_2/8}{(1 - q_0 - q_1)(1 + \epsilon_1\epsilon_2/16)} \\
&\geq \left(f_0 - \frac{\epsilon_1\epsilon_2/8}{(1 - q_0 - q_1)}\right)(1 - \epsilon_1\epsilon_2/8) \\
&\geq f_0 - \epsilon_1/2
\end{aligned}$$

The proof for $\widehat{e}_1$ is identical. If $(1 - q_0 - q_1) < 0$, then note that defining

$$\widehat{e}_0 = \frac{\widehat{q}_0 - \widehat{p}_0}{(\widehat{q}_0 + \widehat{q}_1 - 1)} \quad \text{and} \quad \widehat{e}_1 = \frac{\widehat{q}_1 - \widehat{p}_1}{(\widehat{q}_0 + \widehat{q}_1 - 1)}$$

gives exactly the same values as the original equation above. Also, since the denominator of these new definitions is positive, we can use the same proofs as the ones described above to show that $\widehat{e}_0$ and $\widehat{e}_1$ are within additive error $\epsilon_1/2$ of f_0 and f_1.

Now suppose that there are n_1 edges along g and n_2 edges contracted along h. If we consider the path g first, we know that by the construction of

$\widehat{T}(C)$, every contracted edge g' on g in $T(C)$ must satisfy $\Lambda(g') \geq (1-\epsilon_3)$. Therefore, we must have $\Lambda(g) \geq (1-\epsilon_3)^{n_1} \geq (1-n_1\epsilon_3)$, where the last step follows from the binomial expansion of $(1-\epsilon_3)^{n_1}$. Then, since we assume that all internal edges of $M'(C)$ are "good" edges, we know $g_0 + g_1 \leq 1$. Then by Lemma 2.7, we know that $(1-g_0-g_1) \geq (1-2n_1\epsilon_3)$, so $g_0 + g_1 \leq 2n_1\epsilon_3$. In exactly the same way, we can show that $h_0 + h_1 \leq 2n_2\epsilon_3$. Also, by Observation 2.4,

$$f_0 = h_0 + e_0(1-h_0-h_1) + g_0(1-e_0-e_1)(1-h_0-h_1)$$

$$f_1 = h_1 + e_1(1-h_0-h_1) + g_1(1-e_0-e_1)(1-h_0-h_1)$$

If we rearrange this we can see that $|f_0 - e_0|$ is at most $2(n_1+n_2)\epsilon_3$, and that the same holds for $|f_0 - e_0|$. n_1+n_2 can be at most n, so substituting $\epsilon_1/4n$ for ϵ_3, $|f_0 - e_0| \leq 2n\epsilon_1/4n = \epsilon_1/2$. Then f_0 lies within additive error $\epsilon_1/2$ of e_0, and f_1 lies within additive error $\epsilon_1/2$ of e_1, as required. □

2.5.4 Related sets with less than three leaves

If C has a single leaf x, define $\widehat{M}(C)$ to have the topology consisting of the root x and define the probability for x to be $\widehat{\Pr}(x=1)$. Clearly $\widehat{\Pr}(x=1)$ lies within additive error ϵ_1 of its true value.

If C has two leaves x and y, we follow Subsection 2.3.3 and let the tree consist of a root r with probability $\widehat{\Pr}(x=1)$ and two edges $e = (r \to x)$ and $f = (r \to y)$. We define $e_0 = e_1 = 0$, and

$$\widehat{f}_0 = \frac{\widehat{\Pr}(xy=01)}{\widehat{\Pr}(x=0)} \quad \text{and} \quad \widehat{f}_1 = \frac{\widehat{\Pr}(xy=10)}{\widehat{\Pr}(x=1)}$$

The proof that $\widehat{f}_0$ and $\widehat{f}_1$ lie within additive error ϵ_1 of f_0 and f_1 depends on the fact that $\Pr(x=0)$ and $\Pr(x=1)$ are both at least $\epsilon_2/4$, which we will prove shortly. Since we have assumed that $\widehat{\Pr}(xyz = i_1i_2i_3)$ lies within additive error

$\epsilon_5/32$ of the true value $\Pr(xyz = i_1 i_2 i_3)$ for every $i_1 i_2 i_3 \in \{0,1\}^3$, therefore

$$\begin{aligned}
\frac{\widehat{\Pr}(xy = 01)}{\widehat{\Pr}(x = 0)} &\leq \frac{\Pr(xy = 01) + \epsilon_5}{\Pr(x = 0) - \epsilon_5} \\
&\leq \frac{\Pr(xy = 01) + \epsilon_5}{\Pr(x = 0)(1 - 4\epsilon_5/\epsilon_2)} \\
&\leq \frac{\Pr(xy = 01) + \epsilon_5}{\Pr(x = 0)}(1 + 8\epsilon_5/\epsilon_2)
\end{aligned}$$

and because $\epsilon_5/\epsilon_2 \leq \epsilon_1/16$ (this is a very weak bound), we find that

$$\begin{aligned}
\frac{\widehat{\Pr}(xy = 01)}{\widehat{\Pr}(x = 0)} &\leq (1 + \epsilon_1/2)\left(\frac{\Pr(xy = 01)}{\Pr(x = 0)} + \frac{\epsilon_5}{\Pr(x = 0)}\right) \\
\frac{\widehat{\Pr}(xy = 01)}{\widehat{\Pr}(x = 0)} &\leq (1 + \epsilon_1/2)\left(f_0 + \epsilon_1/4\right)
\end{aligned}$$

by using the fact that $\Pr(x = 0) \geq \epsilon_2/4$ again. So $\widehat{f_0}$ is at most $f_0 + \epsilon_1$. We can also show that $\widehat{f_0} \geq f_0 - \epsilon_1$, and bound $\widehat{f_1}$ to within the same error bound.

To show that $\Pr(x = 0)\Pr(x = 1) \geq \epsilon_2/4$, consider re-rooting the leaf edge f at y. Although this is not really a MET, we can re-root and still preserve the joint distribution on the endpoints of f. Let f_0' and f_1' be the new probabilities along f. By Equation 2.12, $\mathrm{cov}(x, y)$ is $\Pr(x = 0)\Pr(x = 1)(1 - f_0' - f_1')$. Since we know that $|\mathrm{cov}(x, y)| \geq \epsilon_2/4$, we find that $\Pr(x = 0)\Pr(x = 1) \geq \epsilon_2/4$, as required.

2.6 Proof of the main theorem

Let $\widehat{M}$ be a Two-State MET on n leaves which is constructed from M as described in Section 2.2. For every related set C, let r_C be be the root of $\widehat{M}(C)$ and let $\widehat{\Pr}(r_C = 1)$ is the probability at the root of this MET. The product $\widehat{M}$ is constructed by introducing a new root r for $\widehat{M}$, and for every related set of leaves C, adding the edge $\widehat{e[C]} = (r \rightarrow r_C)$. The transition probabilities on these edges are defined by $\widehat{e[C]}_0 = \widehat{\Pr}(r_C = 1)$ and $\widehat{e[C]}_1 = (1 - \widehat{\Pr}(r_C = 1))$. By Observation 2.14, $\widehat{M}$ is the product of all the component METs.

Theorem 1.1 will be proved in two steps. First of all, consider the collection of related sets formed by the leaf connectivity graph, when $\epsilon_2/2$ is used as the threshold for partitioning the graph and the estimates of the covariances all lie within additive error ϵ_4. For every related set C, let $M'(C)$ be a Two-State MET that generates the distribution on M on the leaves in C. Let M' be the product of all the $M'(C)$ distributions. In this section, we will assume that the estimates of our covariances, and of the parameters of the $\widehat{M}(C)$ subMETs, are sufficiently close to their true values (this happens with probability $(1-\delta)$). We will prove that $\mathrm{var}(M, \widehat{M}) \leq \epsilon$ by showing that $\mathrm{var}(M, M') \leq \epsilon/2$ and $\mathrm{var}(M', \widehat{M}) \leq \epsilon/2$.

Lemma 2.47 $\mathrm{var}(M, M') \leq \epsilon/2$.

Before we prove Lemma 2.47, we provide some background material. In this section, for any rooted tree, we will define $w(e)$ for an edge e to be $|1-e_0-e_1|$ and define the weight $w(\ell)$ of a leaf ℓ to be the product of the $w(e)$ values on the path from the root to ℓ. We will use the following lemma.

Lemma 2.48 *In any Two-State MET with root r, the variation distance between the distribution on the leaves conditioned on $r = 1$ and the distribution on the leaves conditioned on $r = 0$ is at most $2\sum_{\ell} w(\ell)$, where the sum is over all leaves ℓ.*

Proof: We proceed by induction on the number of edges in the MET. In the base case r is a leaf, and the result trivially holds. For the inductive step, let e be an edge from r to node x. For any string s on the leaves of the MET, let s_1 be the portion of the string on the leaves below x and s_2 be the string on the other leaves. Then

$$\Pr(s_1 s_2 \mid r = 0) = \Pr(s_2 \mid r = 0)(e_0 \Pr(s_1 \mid x = 1) + (1 - e_0) \Pr(s_1 \mid x = 0).$$

Algebraic manipulation shows that $\Pr(s_1 s_2 \mid r = 1) - \Pr(s_1 s_2 \mid r = 0)$ is

$$\begin{aligned} & (1 - e_0 - e_1)\, \Pr(s_2 \mid r = 1)\, (\Pr(s_1 \mid x = 1) - \Pr(s_1 \mid x = 0)) \\ + \quad & \Pr(s_1 \mid r = 0)\, (\Pr(s_2 \mid r = 1) - \Pr(s_2 \mid r = 0)). \end{aligned} \tag{2.37}$$

It follows that the variation distance is at most the sum over all $s_1 s_2$ of the absolute value of the quantity in Equation 2.37, which is at most

$$\begin{aligned} & |1 - e_0 - e_1| \left(\sum_{s_2} \Pr(s_2 \mid r = 1) \right) \left(\sum_{s_1} |\Pr(s_1 \mid x = 1) - \Pr(s_1 \mid x = 0)| \right) \\ + \quad & \left(\sum_{s_1} \Pr(s_1 \mid r = 0) \right) \left(\sum_{s_2} |\Pr(s_2 \mid r = 1) - \Pr(s_2 \mid r = 0)| \right). \end{aligned}$$

The result follows by induction. □

Lemma 2.49 *Suppose that m is a Two-State MET with n leaves and let e be an edge from node u to node v. Let m' be the MET derived from m by replacing e_0 with $\Pr(v = 1)$ and e_1 with $\Pr(v = 0)$ (We take the product distribution of the two subMETs obtained by disconnecting e). Then $\mathrm{V}(m, m') \le 4 \sum |\mathrm{cov}(x, y)|$, where the sum is taken over all pairs (x, y) of leaves which are connected via e in m.*

Proof: By Observation 2.9, we can assume without loss of generality that u is the root of m. For any string s_1 on the leaves below v and any string s_2 on the remaining leaves, some algebraic manipulation shows that the difference between the probability that m outputs $s_1 s_2$ and the probability that m' does is

$$\mathrm{cov}(u, v)(\Pr(s_2 \mid u = 1) - \Pr(s_2 \mid u = 0))(\Pr(s_1 \mid v = 1) - \Pr(s_1 \mid v = 0)).$$

Remember by Equation 2.12 that $\mathrm{cov}(u, v) = \Pr(u = 1)\Pr(u = 0)(1 - e_0 - e_1)$. Summing over all s_1 and s_2, this shows that the variation distance between m and m' is $\Pr(u = 1)\Pr(u = 0)(1 - e_0 - e_1)$ times the product of the variation distance between the distribution on the leaves below v conditioned on $v = 1$

and the distribution on the leaves below v conditioned on $v = 0$ and the variation distance between the distribution on the remaining leaves conditioned on $u = 1$ and the distribution on the remaining leaves conditioned on $u = 0$. By Lemma 2.48, this is at most

$$\Pr(u = 0)\Pr(u = 1)\left(2 \sum_{\ell \text{ below } v} w(\ell)\right)\left(2 \sum_{\text{other } \ell} w(\ell)\right),$$

which by Equation 2.10 is

$$4 \sum_{(x,y) \text{ connected via } e} |\mathrm{cov}(x, y)|.$$

□

Lemma 2.50 *Suppose that S is a set of leaves in M such that for every edge in $M(S)$, there is some pair of leaves x and y in S such that $|\mathrm{cov}(x, y)| \geq (3\epsilon_2/4)$. Then if we define a leaf connectivity graph on S using the exact inter-leaf covariances, the entire set S is connected for the threshold $(3\epsilon_2/4)$.*

Proof: The proof is by induction on adjacent edges in $T(S)$. Let $e = (u, v)$ and $f = (v, w)$ be adjacent edges in $T(S)$, and let $S(u)$ be the leaves on the "u-side" of e and $S(w)$ be the leaves on the "w-side" of f. We show that if every pair of leaves $x_1, x_2 \in S(u)$ are connected in the leaf connectivity graph on the set S, and the same holds for every pair of leaves $y_1, y_2 \in S(w)$, then every pair of leaves $x \in S(u)$ and $y \in S(w)$ has a connecting path in the graph. To prove connectivity, it is enough to show that some $x \in S(u)$ and $y \in S(w)$ have a connecting path in the graph.

We know that there is some $x \in S(u)$ and some $x' \in S$ such that the connecting path in $T(S)$ from x to x' contains e and $|\mathrm{cov}(x, x')| \geq (3\epsilon_2/4)$. Also, there is some $y \in S(w)$ and some $y' \in S$ such that f lies on the path (y, y') and $|\mathrm{cov}(y, y')| \geq (3\epsilon_2/4)$. If either of $x' \in S(w)$ or $y' \in S(u)$ holds, then there is a connecting path from x to y in the exact graph and we are finished. Otherwise, assume that $x' \neq y'$ (otherwise we are finished). Then, since $x' \notin S(w)$, the path (v, x') does not contain e or f. Also, since $y' \notin S(u)$, the path (v, y') does not contain e or f. Now consider the quartet $(x, y \mid x'y')$. This will be a star if (v, x') and (v, y') only intersect at v.

Let v' be the node on the path (x', y') that is closest to v. Then $(x, y \mid x'y')$ is a quartet with the central path (v, v') (if the topology on x, y, x', y' is a star, $v' = v$). By Equation 2.11,

$$\mathrm{cov}(x, x')\mathrm{cov}(y, y') = \mathrm{cov}(x, y')\mathrm{cov}(y, x')$$

Therefore, either $|\mathrm{cov}(x, y')| \geq (3\epsilon_2/4)$ or $|\mathrm{cov}(y, x')| \geq (3\epsilon_2/4)$ must hold. In either case, we have connected x and y. $\square$

We can now use these Lemmas to prove that $\mathrm{V}(M, M') \leq \epsilon/2$.

Proof of Lemma 2.47: We begin by showing that there is a set of edges $\{e[1], \ldots, e[t]\}$ in M such that

- For every $e[i]$ and every pair of leaves x and y whose connecting path contains $e[i]$, $|\mathrm{cov}(x, y)| \leq 3\epsilon_2/4$;
- Disconnecting all of the $e[i]$ edges gives a partition of the leaf set of M. Every related set C is the union of one or more of the sets in this partition.

To prove this, disconnect all of the edges in M that satisfy the first requirement. Let S be any set in the partition induced by the disconnection. By Lemma 2.50, the set S is related for the threshold $(3\epsilon_2/4)$, when the exact covariances are used to define the leaf connectivity graph. By the closeness of the covariance estimates, it follows that S must be related for the threshold $(\epsilon_2/2)$ in the estimated graph. Then every related set C must be the union of sets from this partition.

We have already defined $M'(C)$ for a related set, and defined M' as the product of these distributions. Now we consider the partition induced on the leaves of M when we disconnect every edge e in M that does not have any pair of leaves x, y connected through e that satisfy $|\mathrm{cov}(x, y)| > (3\epsilon_2/4)$. For every set S in this partition, let $M''(S)$ be any Two-State MET that generates the distribution on the leaves in S. Define M'' as the product of the $M''(S)$ subMETs. We will show that $\mathrm{V}(M, M'') \leq \epsilon/4$ and $\mathrm{V}(M'', M') \leq \epsilon/4$, therefore giving $\mathrm{V}(M, M') \leq \epsilon/2$.

To show that $\mathrm{V}(M, M'') \leq \epsilon/4$, assume that i edges were "cut" in M to give M'', for some i. Assume there is some ordering on these edges, and define M_1 to be the MET obtained by disconnecting the first edge and taking the product of the two subMETs that are obtained; for every $i' < i$, define $M_{i'+1}$ as the MET obtained from $M_{i'}$ by disconnecting the $i' + 1$th edge and taking the product of the two subMETs of $M_{i'}$ that are obtained this way. Note that $M_i = M''$. Then $\mathrm{V}(M, M'') \leq \sum_{i'<i} \mathrm{V}(M_{i'}, M_{i'+1})$. By Lemma 2.49, $\mathrm{V}(M_{i'}, M_{i'+1}) \leq 4 \sum |\mathrm{cov}(x, y)|$ (using the convention that M_0 is the MET M), where the sum is taken over all pairs of leaves x and y that contain the $i' + 1$th disconnecting edge in their path. Also note that if x and y are two leaves in M and the (x, y) path contains more than one of the edges that are "cut", then if the first of these edges to be disconnected is the i'th edge, then $|\mathrm{cov}(x, y)| = 0$ in every MET from $M_{i'}, \ldots M_i$. Therefore every $|\mathrm{cov}(x, y)|$ is counted only once. So $\mathrm{V}(M, M'') \leq 4 \sum |\mathrm{cov}(x, y)|$, where the sum is taken over all pairs of leaves x and y that contain *any* of the disconnected edges in their path, and $|\mathrm{cov}(x, y)|$ is the value of the covariance between x and y in the original MET M. We know that any absolute covariance in this sum is at most $(3\epsilon_2/4)$. Then, since there are at most $n^2/2$ pairs of leaves in the entire tree, we find that $\mathrm{V}(M, M'') \leq 3\epsilon_2 n^2/2$, and since we defined $\epsilon_2 = \epsilon/(8n^2)$, therefore $\mathrm{V}(M, M'') \leq \epsilon/4$.

The proof that $\mathrm{V}(M'', M') \leq \epsilon/4$ is almost exactly the same. The results in Section 2.3 imply that we can assume that M' is a Two-State MET with a star topology at the root and the $M'(C)$ subMETs attached to these edges. Also, because every $M'(C)$ is the union of a number of the subMETs of M'', M'' can be obtained from M' by cutting some edges that lie in the subMETs of M'. By definition, $|\mathrm{cov}(x, y)| \leq (3\epsilon_2/4)$ holds for every pair of leaves x and y whose connecting path contains one of these edges. Again, applying Lemma 2.49 inductively, and using an argument similar to the one we gave for $\mathrm{V}(M, M'')$, to show that $\mathrm{V}(M'', M') \leq \epsilon/4$. Therefore $\mathrm{V}(M, M') \leq \epsilon/2$. $\square$

Lemma 2.51 *Suppose that for every set C of related leaves, there is some $M'(C)$ that generates $M(C)$, and that every parameter of $\widehat{M}(C)$ is within additive error ϵ_1 of the corresponding parameter in $M'(C)$. Let M' be the product of the $M'(C)$ subMETs, and let $\widehat{M}$ be the product of the $\widehat{M}(C)$ subMETs. Then*

$$\mathrm{var}(M', \widehat{M}) \leq \epsilon/2.$$

Proof: Let $\widehat{M}$ be defined exactly as we described at the beginning of this section. Assume without loss of generality that M' is realized by a root r, an edge $e[C] = (r \to r_C)$ from r to the root of component $M'(C)$ for every related set C. Also assume without loss of generality (see Observation 2.9, Observation 2.11 and Observation 2.14) that each $M'(C)$ is the particular Two-State MET that generates the distribution $M(C)$ and that is close (all parameters are within additive error ϵ_1) to $\widehat{M}(C)$. The edges adjacent to the root of M' are labelled by $e[C]_0 = \Pr(r_C = 1)$ and $e[C]_1 = \Pr(r_C = 0)$.

There are at most $2n$ edges in any tree on any tree on n leaves, so therefore, we have at most $4n + 1$ parameters (two parameters for each edge, and one root parameter). We will now show that changing a single parameter of a MET by at most $\pm\epsilon_1$ yields at MET whose variation distance from the original is at most

$2\epsilon_1$. This implies that $\mathrm{var}(M', \widehat{M}) \leq (4n+1)2\epsilon_1 = \epsilon/2$. Suppose that e is an edge from u to v and that we change e_0 by ϵ_1. The probability that the output has string s_1 on the leaves below v and string s_2 on the remaining leaves is

$$\begin{aligned} & \Pr(u=0)\Pr(s_2 \mid u=0)(e_0 \Pr(s_1 \mid v=1) + (1-e_0)\Pr(s_1 \mid v=0)) \\ + \quad & \Pr(u=1)\Pr(s_2 \mid u=1)(e_1 \Pr(s_1 \mid v=0) + (1-e_1)\Pr(s_1 \mid v=1)). \end{aligned}$$

Thus, the variation distance between a Two-State MET and another Two-State MET obtained by changing the value of e_0 (within $\pm\epsilon_1$) is at most

$$\begin{aligned} & \epsilon_1 \sum_s \sum_{s'} \Pr(u=0)\Pr(s' \mid u=0)(\Pr(s \mid v=1) + \Pr(s \mid v=0)) \\ \leq \quad & \epsilon_1 \Pr(u=0) \left(\sum_{s'} \Pr(s' \mid u=0)\right) \left((\sum_s \Pr(s \mid v=1)) + (\sum_s \Pr(s \mid v=0))\right) \end{aligned}$$

which is at most $2\epsilon_1$. Similarly, if ρ_1 is the root probability of a MET then the probability of having output s is

$$\rho_1 \Pr(s \mid r=1) + (1-\rho_1)\Pr(s \mid r=0).$$

So the variation distance between the original MET and one in which ρ_1 is changed within $\pm\epsilon_1$ is at most

$$\sum_s \epsilon_1(\Pr(s \mid r=1) + \Pr(s \mid r=0)) \leq 2\epsilon_1.$$

$\square$

2.6.1 Extension to KL-distance

Now we consider the relationship between PAC-learnability in variation distance and PAC-learnability in KL-distance. In Subsection 1.3.2, we mentioned that Cover and Thomas [13] show that $\mathrm{V}(D_1, D_2) \leq \sqrt{(2\ln 2)\mathrm{KL}(D_1, D_2)}$, for any two distributions D_1 and D_2. This implies that every class of distributions that can be learned in KL-distance can also be learned in variation distance, with exactly the same hypothesis class.

In this Subsection we will prove a Lemma that shows that if the hypothesis class for a PAC-learning problem is sufficiently general, then a class of distributions that can be learned in variation distance can also be learned in KL-distance. This result is not as strong as the result in the opposite direction, because the proof depends on altering the original hypothesis returned by the learning algorithm for variation distance. The Lemma is proved using a method related to the ϵ-Bayesian shift of Abe and Warmuth [1].

Lemma 2.52 *When the hypothesis class for a learning problem is general enough, a class of probability distributions over the domain $\{0,1\}^n$ that is PAC-learnable under the variation distance metric is also PAC-learnable under the KL-distance measure.*

Proof: Let K be a polynomial in three inputs and let A be an algorithm which takes as input $K(n, 1/\epsilon, 1/\delta)$ samples from a distribution $\mathcal{D}$ from the class of distributions and, with probability at least $1-\delta$, returns a distribution $\mathcal{D}'$ such that $\mathrm{V}(\mathcal{D}, \mathcal{D}') \leq \epsilon$. Without loss of generality, we can assume that ϵ is sufficiently small. For example, it will suffice to have $\epsilon \leq 2/15$.

Define algorithm A' as follows. Let $\xi = \epsilon^2/(12n)$. Run A with sample size $K(n, 1/\xi, 1/\delta)$ (note that the sample size is polynomial in n, $1/\epsilon$, and $1/\delta$). Let $\mathcal{D}'$ be the distribution returned by A, let U denote the uniform distribution on $\{0,1\}^n$ and let $\mathcal{D}''$ be the distribution defined by

$$\mathcal{D}''(s) = (1-\xi)\mathcal{D}'(s) + \xi\, U(s).$$

With probability at least $1-\delta$, $\mathrm{V}(\mathcal{D}, \mathcal{D}') \leq \xi$, and by definition, $\mathrm{V}(\mathcal{D}', \mathcal{D}'') \leq 2\xi$. Then, with probability at least $1-\delta$, $\mathrm{V}(\mathcal{D}, \mathcal{D}'') < 3\xi$. For all s, $\mathcal{D}''(s) \geq \xi\, 2^{-n}$. We define S as the set of all output strings s such that $\mathcal{D}''(s) < \mathcal{D}(s)$. S contains all the strings which contribute positively to the KL-distance from $\mathcal{D}$ to $\mathcal{D}''$. Then

$$\begin{aligned}\mathrm{KL}(\mathcal{D},\mathcal{D}'') &\leq \sum_{s\in S}\mathcal{D}(s)(\log\mathcal{D}(s)-\log\mathcal{D}''(s))\\ &= \sum_{s\in S}(\mathcal{D}(s)-\mathcal{D}''(s))(\log\mathcal{D}(s)-\log\mathcal{D}''(s))\\ &\quad+\sum_{s\in S}\mathcal{D}''(s)(\log\mathcal{D}(s)-\log\mathcal{D}''(s)).\end{aligned}$$

We have seen that $\mathrm{V}(\mathcal{D},\mathcal{D}'') \leq 3\xi$. Then $\sum_{s\in S}(\mathcal{D}(s)-\mathcal{D}''(s)) \leq 3\xi$. So, the first term is at most

$$\begin{aligned}&\max_{s\in S}(\log\mathcal{D}(s)-\log\mathcal{D}''(s))\sum_{s\in S}(\mathcal{D}(s)-\mathcal{D}''(s))\\ \leq\ & 3\xi\ \max_{s\in S}(\log\mathcal{D}(s)-\log\mathcal{D}''(s))\\ \leq\ & 3\xi\ \max_{s\in S}(-\log\mathcal{D}''(s))\\ \leq\ & 3\xi(-\log(\xi\, 2^{-n}))\\ =\ & 3\xi(n-\log(\xi)).\end{aligned}$$

Furthermore, the second term is at most

$$\begin{aligned}&\sum_{s\in S}\mathcal{D}''(s)(\log\mathcal{D}(s)-\log\mathcal{D}''(s))\\ =\ & \sum_{s\in S}\mathcal{D}''(s)(\log(\mathcal{D}''(s)+h_s)-\log\mathcal{D}''(s)),\end{aligned}$$

where $h_s = \mathcal{D}(s)-\mathcal{D}''(s)$, which is a positive quantity for $s \in S$. By concavity of the logarithm function, the above quantity is at most

$$\sum_{s\in S}\mathcal{D}''(s)h_s\Big[\frac{d}{dx}(\log(x))\Big]_{x=\mathcal{D}''(s)} = \sum_{s\in S}h_s \leq 3\xi.$$

Thus, $\mathrm{KL}(\mathcal{D},\mathcal{D}'') \leq 3\xi(1+n-\log\xi)$. This quantity is at most ϵ for all $n \geq 1$ by the definition of ξ. □

The method described in Lemma 2.52 converts a hypothesis distribution which is close to the original distribution in variation distance to a hypothesis distribution which is close to the target distribution in KL-distance. However, if the original hypothesis is a Two-State MET, then the modified hypothesis will be the weighted sum of the distribution of the Two-State MET and the uniform distribution on $\{0,1\}^n$. This weighted sum can be represented by a 3-State MET in the following way: if $\widehat{M}$ is the hypothesis Two-State MET, let ρ be the root of $\widehat{M}$. Now define a 3-State MET M' with the same topology as $\widehat{M}$ and define $\rho'_0 = \rho_0 * (1-\xi)$, $\rho'_1 = \rho_1 * (1-\xi)$ and $\rho'_2 = \xi$. The transition matrices for $\widehat{M}$ are defined as follows: for every internal edge e,

$$M'_e[i,i'] = \begin{cases} M_e[i,i'], & \text{if } i,i' \in \{0,1\} \\ 0, & \text{for } [i,i'] \in \{[0,2],[1,2],[2,0],[2,1]\} \\ 1, & \text{if } i = i' = 2. \end{cases}$$

For every leaf edge e, we define

$$M'_e[i,i'] = \begin{cases} M_e[i,i'], & \text{if } i,i' \in \{0,1\} \\ 0, & \text{for } [i,i'] \in \{[0,2],[1,2],[2,2]\} \\ 1/2, & \text{for } [i,i'] \in \{[2,0],[2,1]\}. \end{cases}$$

The key to understanding why this new MET generates the weighted sum of the original hypothesis and the uniform distribution on $\{0,1\}^n$ is the fact that on any broadcast from the tree, the root is labelled 2 if and only if all of the internal nodes are labelled 2. Then the leaf transition matrices ensure that the distribution of the 3-State MET, conditioned on the root being labelled 2, is the uniform distribution.

We now show that is also possible to modify the distribution of the hypothesis Two-State MET $\widehat{M}$ to obtain a new Two-State MET that is close to the original MET in KL-distance.

As before, we run our PAC-learning algorithm with the accuracy parameter $\xi = \epsilon^2/(12n^3)$ to obtain a MET $\widehat{M}$. The new Two-State MET M'' is obtained as follows: For each edge $e = (u, \ell)$ of $\widehat{M}$ where ℓ is a leaf, define e_0'' and e_1'' as follows: If $\widehat{e}_0 < \xi$ then set $e_0'' = \xi$ and if $\widehat{e}_0 > 1 - \xi$ then set $e_0'' = 1 - \xi$. Otherwise define $e_0'' = \widehat{e}_0$. We define e_1'' in the same way. By the proof of Lemma 2.51, $\mathrm{V}(\widehat{M}, M'') \leq 4n\xi$, since $2n$ parameters have each been changed by at most ξ, and therefore with probability at least $1 - \delta$, $\mathrm{V}(M, M'') \leq (1 + 4n)\xi$.

For each string $s \in \{0,1\}^n$, $M''(s) \geq \xi^n$. Using a similar argument to the proof of lemma 2.52,

$$\mathrm{KL}(M, M'') \leq (1 + 4n)\xi(1 - \log(\xi^n)) = (1 + 4n)\xi(1 - n \log \xi)$$

$$= (1 + 4n)\frac{\epsilon^2}{12n^3}(1 - n(2\log\epsilon - 3\log n - \log 12))$$

which as before is at most ϵ for all $n \geq 1$.

Chapter 3

Approximation results for some tree problems

3.1 Introduction

This Chapter of the thesis contains results on two character-based problems that involve labelling a known topology in order to minimize some function:

Definition 3.1 *A* character *c, which is sometimes also called a* monomorphic character*, is a function from a species set S to some set R_c of* character states. *A* polymorphic character *is a function from S to the power set of R_c (excluding the empty set). We will use the notation $c : S \to 2^{R_c} - \{\}$ to represent polymorphic characters.*

As we explained in Chapter 1, we usually think about a character as specifying morphological properties such as egg colour or the ability to fly. Many of the problems that have been previously studied assume that data is available in the form of monomorphic characters, so it is assumed that every species only exhibits a single state for each character. However, it has been pointed out that polymorphism is a quite common phenomenon in biological species and especially in comparative linguistics, where the subject of study is the evolutionary relationships between natural languages (see Bonet et al. [8]).

Many optimization problems related to evolutionary tree construction are described in terms of a set of species and a set of characters on those species. Then, if the set of characters is $c_1, \ldots, c_k$, any vector from $R_{c_1} \times \ldots \times R_{c_k}$ represents a hypothetical species that may be an extinct ancestor or even a species that never existed (in the polymorphic case, the ith position in this vector will be an element of $2^{R_{c_i}} - \{\}$). Any tree T whose leaves are bijectively labelled by the species in S and whose internal nodes are labelled by hypothetical species represents a hypothetical evolutionary tree (also called a hypothetical *phylogeny*) for that species set. Most character-based problems involve finding a phylogeny that minimizes some function on the tree for the original set of species and characters. In other cases, we may already be given a *fixed-topology* T as part of the input, and we may simply want to label the internal nodes of this tree with hypothetical ancestral species to minimize some quantity.

In Sections 3.3 and 3.4 of this chapter, we present approximation algorithms for two different $\mathcal{NP}$-hard fixed-topology problems. The next section of this thesis is devoted to providing some background about related research.

3.2 Previous research on Character-Based problems

The three character-based phylogeny problems that have been studied most widely in the theoretical science community are the *ℓ-phylogeny* problem, the *parsimony* problem and the *compatibility* problem. The original version of each of these problems assumes that the data on a set of species consists of a collection of monomorphic characters.

The ℓ-phylogeny metric was introduced by Goldberg, Goldberg, Phillips, Sweedyk and Warnow [32] and is a generalization of an older problem called the *perfect phylogeny* problem. Given a phylogenetic tree T, a character c_i and a state $j \in R_{c_i}$, let ℓ_{ij} be the number of connected components in ${c_i}^{-1}(j)$ (the subtree induced by the species with state j in character i). Then we say that this phylogeny is an ℓ-phylogeny if and only if $\max_{c_i, j \in R_{c_i}} \ell_{ij} \leq \ell$. The ℓ-phylogeny problem is to determine if an input consisting of a species set S and a set of characters $c_1, \ldots, c_k$ has an ℓ-phylogeny, and the *phylogenetic number* problem is to determine the minimum value of ℓ for which an ℓ-phylogeny exists. A 1-phylogeny is called a *perfect phylogeny*.

The parsimony problem is to find a phylogeny that minimizes $\sum_{c_i, j \in R_{c_i}} \ell_{ij}$, and compatibility aims to maximize $|\{c_i : \ell_{ij} = 1 \text{ for all } j \in R_{c_i}\}|$.

Although parsimony, ℓ-phylogeny, and compatibility all allow states of a character to evolve multiple times, they have different ways of distributing these changes among the character states. When the collection of characters for a given species set has a perfect phylogeny, then this phylogeny is also the optimum tree under parsimony and compatibility. In general, parsimony is the least constrained of the criteria, because it minimizes the *total* number of evolutionary changes in the tree. Compatibility allows some characters to evolve many times, to obtain a perfect labelling for as many characters as possible. The ℓ-phylogeny metric requires *balanced* evolution, so that no character state evolves too many times. Thus, ℓ-phylogeny may be a better measure than parsimony or compatibility in biological situations in which all characters are believed to evolve slowly.

It is well-known that the *perfect phylogeny* problem is $\mathcal{NP}$-complete (shown independently by Bodlaender, Fellows and Warnow [7] and Steel [57]). If the maximum number of states allowed for any character is bounded by a constant, then the problem can be solved in polynomial time (see Gusfield [35] for details on the binary character case, Kannan and Warnow for sets of characters with at most four states [42, 41], and Agarwala and Fernández-Baca [3] for the general case). More recently, Goldberg et al. [32] proved that the ℓ-phylogeny problem is $\mathcal{NP}$-complete for $\ell \geq 2$. It has also been shown that parsimony, and some variations of the classic parsimony problem, are $\mathcal{NP}$-hard (See Day [20], Day, Johnson and Sankoff [21] and Graham and Foulds [33] [1]). The decision version of the compatibility problem was shown to be $\mathcal{NP}$-complete by Day and Sankoff [22] (See also Steel [57]).

In Section 3.3 we will consider the *fixed-topology* phylogenetic number problem, when we are given a set of species S, a collection of monomorphic characters $c_1, \ldots, c_k$ and a tree topology T whose leaves are bijectively labelled by the elements of S. The problem is to find a labelling for the internal nodes of T that minimizes the phylogenetic number of the tree. This problem was introduced in the paper by Goldberg et al [32]. The motivation for considering this problem is to define a *filter* for hypothetical evolutionary trees that have been produced by different algorithms. If there are a number of different topologies that are considered to be equally likely under some other measure, one way of distinguishing between these hypotheses is to see which topologies also have low phylogenetic number. In the paper by Goldberg et al. [32], it was shown that the fixed-topology ℓ-phylogeny problem can be solved in polynomial time for $\ell \leq 2$, but is $\mathcal{NP}$-complete for $\ell \geq 3$ even for degree-3 trees in which no state labels more than $\ell + 1$ leaves (and therefore there is a trivial $\ell + 1$ phylogeny). This is in contrast to the fixed-topology parsimony problem and the fixed-topology compatibility problem, both of which can be solved in polynomial

[1] Gusfield [35] and Wareham [64] both correct minor errors in the reductions used by Graham and Foulds [33] and Day [20] respectively.

time using dynamic programming techniques (see Fitch[29] and Hartigan [36]). Goldberg et al. also showed that if the value of $|R_{c_i}|$, for every i, is bounded above by a constant then the fixed-topology phylogenetic number problem can be solved in polynomial time.

In Section 3.3 we give a 2-approximation algorithm for the general fixed-topology phylogenetic number problem. We first show that it is possible to write any instance of this problem as an Integer Program whose solution gives an optimal labelling for the tree. However, Integer linear programming is $\mathcal{NP}$-hard in general (see Karp [43], Borosh and Treybig [9] and Garey and Johnson [31]), so we cannot expect to solve an Integer Program in polynomial time (In particular, this would allow us to solve the fixed-topology ℓ-phylogeny problem in polynomial-time, and we know this is $\mathcal{NP}$-hard for $\ell \geq 3$.) However, Linear Programs can be solved in polynomial-time, so we solve the linear relaxation of the original Integer Program. We show that by rounding the values in the solution to this Linear Program, we can approximate the phylogenetic number within a factor of 2.

First we will explain why the fixed-topology phylogenetic number problem cannot be solved by adapting recent techniques for similar fixed-topology problems. The research that we are referring to is the research of Jiang, Lawler, and Wang [39], who considered the fixed-topology tree-alignment problem, where species are represented as biomolecular sequences and the cost of an edge in a labelled tree is the edit distance between the sequences at its endpoints. Edit distance is a weighted measure of the number of mutations, deletions and insertions needed to transform one sequence into another; the algorithm of Jiang et al. only relies on the fact that edit distance satisfies the triangle inequality and that the edit distance between two sequences can be computed efficiently. The optimal solution to the tree alignment problem is the labelling that minimizes the sum of the edit distances over all edges.

Jiang et al. showed that the tree alignment problem is $\mathcal{NP}$-hard, but gave a 2-approximation for bounded-degree input topologies and extended this to obtain a polynomial-time approximation scheme (PTAS). In Lemma 3 of [39], they showed that the best lifted tree (each internal node is labelled by one of its children) is within a factor of 2 of the best tree with arbitrary labels. The proof only uses the fact that edit distance satisfies the triangle inequality; therefore the result holds for many other cost measures, including ℓ-phylogeny and parsimony. It also holds for the variant of ℓ-phylogeny in which a different ℓ_i is specified for each character c_i. This variant was introduced by Goldberg et al. [32] and is called the *generalized ℓ-phylogeny problem*. In fact, Lemma 3 of [39] holds for the fixed-topology problem with *arbitrary* input topologies, though the authors do not state this fact since they do not use it.

It might be hoped that Lemma 3 of the paper by Jiang et al. [39] could be used to obtain an approximation algorithm for the fixed-topology phylogenetic number problem. However, Jiang et al. use dynamic programming to find a minimum-cost lifted tree. Suppose that the tree has n leaves. When calculating the labelling for the subtree $T(u)$ rooted at u, there are at most n possible labellings for u. Also, because the cost is summed over edges of the tree, it is only necessary to find one labelling of $T(u)$ such that the labelling has minimum cost for a given root label for u. However, this dynamic programming approach does not seem to extend to the more global metric of ℓ-phylogeny. The phylogenetic number of a tree depends upon how many times each state is broken for a given character. Therefore, instead of maintaining a single optimal tree for each root label, it is necessary to maintain all trees whose cost (represented as a vector of components for each state) is undominated. This number can be exponential in r, the number of states, even for bounded-degree input trees. Therefore, we cannot use this approach to approximate the phylogenetic number of a fixed-topology.

Gusfield and Wang [62] improved the results of Jiang et al. by proving that the best uniform lifted tree (ULT) is within a factor of 2 of the best arbitrarily-labelled tree. In a uniform lifted tree on each level, the internal nodes on a particular level are labelled by the same child (e.g. all nodes at level one take the label of their leftmost child). Again, this proof extends to the ℓ-phylogeny metric. If the input tree is a complete binary tree, then there are only n ULTs, and exhaustive search gives a 2-approximation algorithm for our problem. However, when the input tree is not complete, Gusfield and Wang use dynamic programming to find the minimum-cost ULT, so their method cannot be adapted for the ℓ-phylogeny problem. Wang, Jiang, and Gusfield later improved the efficiency of their PTAS for tree alignment [63], but still use dynamic programming.

In Section 3.3 we present an approximation algorithm for the fixed-topology phylogenetic number problem, that is based on linear programming and that has an approximation ratio of 2 for any input topology.

In section 3.4, we extend our linear-programming techniques to find an approximation algorithm for a different fixed-topology problem. This input to this problem consists of a fixed-topology T whose leaves are bijectively labelled by species, a collection of polymorphic characters on the leaves of the tree, and a *load* ℓ that denotes the maximum number of states allowed to label a species or internal node, for any character c. Bonet et al [8] previously considered a generalization of the perfect phylogeny problem for polymorphic characters: given a set of species S and a set of polymorphic characters, determine the minimum load such that there is a phylogeny with this load whose leaves are bijectively labelled by the elements of S and every state of every character forms a single connected component in the phylogeny. They proved that this problem is $\mathcal{NP}$-complete, and gave polynomial-time algorithms for some special cases of the problem. They also introduced a problem called the *load* problem, and showed that this problem is $\mathcal{NP}$-hard, even when the input includes a fixed-topology. The load problem is the problem that we consider in Section 3.4.

3.3 Approximating the Fixed-Topology Phylogenetic Number

Two facts will be useful. First, when we aim to construct a phylogeny on a fixed-topology, the labelling of nodes for one character does not influence the choice of labels for any other character. Therefore in this section we will consider each character separately. Also, it will sometimes be convenient to allow a node to remain unlabelled in a fixed topology. If we adopt the convention that an unlabelled node disagrees with all other states of any character, then any labelling can be extended to one in which every node is labelled, without increasing the number of connected components for any state: for any connected component of nodes that are not labelled, we simply choose the state of some neighbouring node and label the entire component with this neighbour's state. This does not introduce any extra component for any state, nor does it break any components of the original labelling.

Let $c : S \to \{1, \ldots, r\}$ be a character and let T be a tree with root ρ whose leaves are labelled by states from $1, \ldots, r$. For each state i, T_i is the subtree of T induced by the leaves labelled i. Let $L(T_i)$ be the set of leaves of T_i, and let rt_i, the root of T_i, be the node of T_i closest to ρ. The *important nodes* of T_i are the leaf nodes and the nodes of degree greater than 2. An *i-path* p of T_i is a sequence of edges of T_i that connects two important nodes of T_i, but does not pass through any other important nodes. The two important nodes are referred to as the *endpoints* of p, and the other nodes along the i-path are said to be *on* p (an i-path need not have any nodes on it). Although the edges of T are undirected, we will sometimes use the notation $(u \to v)$ for an edge or i-path with endpoints u and v, to indicate that u is nearer to the root of T than v (u is the *higher* endpoint and v is the *lower* endpoint); otherwise we will write edges and i-paths as (u,v). If the label of the upper endpoint u or some node on the i-path $p = (u \to v)$ differs from the label for the lower endpoint v, then we say that p *breaks* state i.

We begin by giving an expression that counts the number of components induced by the nodes labelled i, for any state i, whenever the nodes of T are labelled by elements of $\{1, \ldots, r\}$. Since the tree is rooted, we can assume that each connected component has a root, namely the node closest to the root of T. Then a node labelled i is the root of its component if its label differs from that of its parent in T. Also, the root ρ, which has no parent, is the root of its component. Therefore we have the following:

Observation 3.1 *Let T be a tree with its leaves and internal nodes labelled by elements of $\{1, \ldots, r\}$. Let ρ be the root of T. Then the number of connected components induced by the nodes labelled i is $|\{e = (u \to v) : c(u) \neq i,\ c(v) = i\}| + Y_i$, where $Y_i = 1$ if ρ is labelled i and 0 otherwise.*

We now define an Integer Linear Program (ILP) which solves the fixed-topology phylogenetic number problem. The Linear Program (LP) obtained by relaxing this ILP is the key to our 2-approximation algorithm. The Integer Linear Program $\mathcal{I}$ uses the variables $X_{x,i}$, for each state $i \in \{1, \ldots, r\}$, and each node x in the tree T, and the variables $X_{p,i}$ and $cost_{p,v,i}$, for each state i, each i-path p of T_i, and each lower endpoint v of the path p. These variables have the following interpretation:

$$
\begin{aligned}
X_{x,i} &= \begin{cases} 1 & \text{if node } x \text{ is labelled } i \\ 0 & \text{otherwise} \end{cases} \\
X_{p,i} &= \begin{cases} 1 & \text{if all nodes on } p \text{ are labelled } i \\ 0 & \text{otherwise} \end{cases} \\
cost_{p,v,i} &= \begin{cases} 1 & \text{if lower endpoint } v \text{ of } p \text{ is the root of a component of state } i \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

ILP $\mathcal{I}$ is defined as follows:

$$\text{minimize } \ell$$

subject to

$$
\begin{aligned}
X_{x,i} &= 1 && \text{for each leaf } x \in T_i && (3.1) \\
X_{x,i} &= 0 && \text{if } x \notin T_i && (3.2) \\
\sum_{i=1}^{r} X_{x,i} &\leq 1 && \forall x \in T && (3.3) \\
X_{p,i} &= X_{x,i} && \forall p \in T_i,\ \forall x \text{ on } p && (3.4) \\
X_{p,i} &\leq X_{x,i} && \forall p \in T_i,\ \text{each endpoint } x \in p && (3.5) \\
cost_{p,v,i} &\geq X_{v,i} - X_{p,i} && \forall p \in T_i,\ \text{lower endpoint } v \in p && (3.6) \\
\sum_{p,v} cost_{p,v,i} + X_{rt_i,i} &\leq \ell && && (3.7) \\
X_{x,i}, X_{p,i}, cost_{p,v,i} &\in \{0,1\} && && (3.8)
\end{aligned}
$$

For each set of constraints in $\mathcal{I}$, except Constraint (3.3), we add the set of constraints for *every* $i \in \{1, \ldots, r\}$.

Constraint (3.8) assures that the cost ($cost_{p,v,i}$), i-path ($X_{p,i}$), and vertex ($X_{x,i}$) variables serve as indicator variables in accordance with their interpretation. Constraint (3.1) labels the leaves in accordance with the input. Constraint (3.2) prohibits labelling a node x with a state i when x is not in T_i (the number of components labelled i could not possibly be reduced by this labelling). Constraint (3.3) ensures that each internal node will have no more than one label. Constraints (3.4) and (3.5) are not essential for the Integer program formulation of the fixed-topology ℓ-phylogeny problem. They are included because they will be useful when we use the linear relaxation of the Integer program as the basis for our 2-approximation algorithm. Together, Constraints (3.4) and (3.5) ensure that for each tree T_i, nodes on paths are taken all-or-none; if any node on an i-path p (including endpoints) is lost to a state i, then it does no good to have any of the other nodes on the path (though it may be beneficial to maintain one or both endpoints). Constraint (3.6) computes the path costs, and Constraint (3.7) ensures that each state has no more than ℓ connected components. This is an implementation of Observation 3.1. Since there is no i-path in T_i with rt_i as its lower endpoint, we must explicitly check the root of each tree T_i, just as we checked the global root in Observation 3.1.

Integer program $\mathcal{I}$ solves the fixed-topology ℓ-phylogeny problem. We will now show that the optimal value of ℓ given by $\mathcal{I}$ is a lower bound on the phylogenetic number of tree T with the given leaf labelling.

Proposition 3.2 *If there exists an ℓ-phylogeny for tree T with a given leaf labelling, then there is a feasible solution for the integer linear program for this value of ℓ.*

Proof: Suppose there exists an ℓ-phylogeny on the tree T with leaves and internal nodes labelled from $\{1, \ldots, r\}$. Consider one particular ℓ-phylogeny, and assume without loss of generality that all node labels are useful for connectivity (i.e. changing the label of node x from i to something else will increase the number of components labelled i). This may require some nodes to be unlabelled. We obtain a feasible solution to $\mathcal{I}$ as follows. Set variable $X_{x,i}$ to 1 if node x is labelled i in this phylogeny and 0 otherwise. Set $X_{p,i}$ to 1 if both endpoints and all internal nodes of i-path p are labelled i and 0 otherwise. Set $cost_{p,v,i} = 1$ if lower endpoint v of p is labelled i and the i-path is not, and set $cost_{p,v,i} = 0$ otherwise. We now show this assignment is a solution to $\mathcal{I}$.

The $X_{x,i}$, $X_{p,i}$, and $cost_{p,v,i}$ variables are binary by construction, thus satisfying Constraint (3.8). By construction, Constraint (3.1) will be satisfied by our assignment. Constraint (3.2) will also be satisfied, because it is never useful to label nodes outside T_i with i, and we have assumed all the labels on nodes are useful for connectivity. Constraint (3.3) is also satisfied because each node of the phylogeny will be labelled with at most one state. Constraints (3.4) and (3.5) are satisfied because the condition that all labelled nodes are necessary for connectivity ensures that a node on an i-path will only be labelled i if all the nodes and endpoints of the i-path are labelled i. Constraint (3.6) is satisfied by construction.

To show that Constraint (3.7) is satisfied, consider the connected components for i; by assumption, these all lie in T_i. Let $\gamma = \{e = (u \to v) : c(u) \neq i,\ c(v) = i\}$. By Observation 3.1 we have $|\gamma| + X_{\rho,i} \leq \ell$, where ρ is the root of T. To calculate $(\sum_{p,v} cost_{p,v,i}) + X_{rt_i,i}$, note that $cost_{p,v,i} = 1$ if and only if $X_{v,i} = 1$ for lower endpoint v and $X_{p,i} = 0$ and otherwise $cost_{p,v,i}$ is 0. By our definitions above, $X_{p,i} = 0$ and $X_{v,i} = 1$ iff the edge $(u, v) \in T$ from v's parent (on the i-path p or its upper endpoint) into v has $c(u) \neq i$ and $c(v) = i$. Furthermore, this is the only edge on the i-path with this property (every other edge costs 0), unless path p passes through a degree-2 root and both its endpoints have breaks. In the latter case there is a second lower endpoint v' such that $cost_{p,v',i} = 1$. Since the i-paths partition T_i, each i-path p and lower endpoint v with $cost_{p,v,i} = 1$ contains one element of γ which is unique to that i-path. Thus $(\sum_{p,v} cost_{p,v,i}) \leq |\gamma| \leq \ell$. If rt_i is the node ρ, then $X_{rt_i,i} = X_{\rho,i}$ and $(\sum_{p,v} cost_{p,v,i}) + X_{rt_i,i} \leq |\gamma| + X_{\rho,i} \leq \ell$. Otherwise, by our assumption that only useful nodes of T are labelled with i, the ancestor node a_i of rt_i is not labelled i. Then, if $X_{rt_i,i} = 1$ the edge $e = (a_i \to rt_i)$ contributes 1 to $|\gamma|$, and therefore $(\sum_{p,v} cost_{p,v,i}) + X_{rt_i,i} \leq |\gamma| + X_{\rho,i} \leq \ell$. Hence Constraints (3.7) are satisfied and we have a solution for the integer program $\mathcal{I}$. □

Integer linear programming in $\mathcal{NP}$-hard in general [9, 31, 43], so we cannot expect to solve it directly in polynomial time. However, we can solve the linear-programming relaxation $\mathcal{L}$ of $\mathcal{I}$, which consists of all the constraints of $\mathcal{I}$ except that Constraint (3.8) is replaced by

$$0 \quad \leq \quad X_{x,i}, X_{p,i}, cost_{p,v,i} \quad \leq \quad 1 \qquad\qquad (3.8').$$

Theorem 3.3 *If there is a solution for the linear program $\mathcal{L}$ for a fixed topology T with leaves labelled with states from $\{1, \ldots, r\}$, then we can assign states to the internal nodes of T such that no state $i \in \{1, \ldots, r\}$ has more than 2ℓ components.*

Proof: The 2ℓ phylogeny for the character $c : S \to \{1, \ldots, r\}$ on T is constructed by assigning states to the nodes of each tree T_i based on the $X_{x,i}$ values. For each state $i \in \{1, \ldots, r\}$, consider each internal node u of T_i. A node u is labelled i if and only if $X_{u,i} > 1/2$, and there is a path $u, v_1, v_2, \ldots, v_k, v^*$ through tree T_i to a leaf v^* of T_i where $X_{v_j,i} > 1/2$ for all $j = 1, \ldots, k$. If $X_{u,i} > 1/2$, but there is no such path, then node u is *isolated*, and by our procedure remains unlabelled. A node u also remains unlabelled if $X_{u,i} \leq 1/2$ for all states i.

To show that the labelling is a 2ℓ-phylogeny, we show that each component of state i adds at least $1/2$ to the sum $(\sum_{p,v} cost_{p,v,i}) + X_{rt_i,i}$. From Observation 3.1, the number of connected components for the state i is $|\{e = (u \to v) :$

$c(u) \neq i,\ c(v) = i\}| + Y_i$, where Y_i is 1 if ρ has state i (and therefore $\rho = rt_i$) and 0 otherwise. Constraints (3.5) and (3.4) ensure that if the edge $e = (u \rightarrow v)$ has $c(u) \neq i$ and $c(v) = i$ then either v is the root of T_i, or v must be an endpoint node with $X_{v,i} > 1/2$, and that either $X_{u,i} \leq 1/2$ or u is isolated. However, since v is labelled i, v must not be isolated, and therefore u would not be isolated if $X_{u,i}$ was greater than 1/2. So $X_{u,i} \leq 1/2$, and $X_{p,i} \leq 1/2$ for the i-path p with lower endpoint v. Therefore we need only calculate the number of i-paths p with lower endpoint v such that $X_{p,i} \leq 1/2$, $X_{v,i} > 1/2$, and v is not isolated.

Suppose $p = (u \rightarrow v)$ is such an i-path. Since v is not isolated and the node above v is not labelled i, there is a sequence $p_1 = (v \rightarrow v_1)$, $p_2 = (v_1 \rightarrow v_2), \ldots, p_j = (v_{j-1} \rightarrow v_j)$ of i-paths of T_i such that $X_{p,i} > 1/2$ for every $p \in \{p_1, \ldots, p_j\}$ and $X_{x,i} > 1/2$ for every $x \in \{v_1, \ldots, v_j\}$, and v_j is a leaf of T_i. Calculating $cost_{p,v,i} + cost_{p_1,v_1,i} + \ldots + cost_{p_j,v_j,i} = (X_{v,i} - X_{p,i}) + (X_{v_1,i} - X_{p_1,i}) + \ldots + (X_{v_j,i} - X_{p_j}) = -X_{p,i} + (X_{v,i} - X_{p_1,i}) + (X_{v_1,i} - X_{p_2}) + \ldots + (X_{v_{j-1},i} - X_{p_j,i}) + X_{v_j,i}$, we know by Constraints (3.5) that $X_{v,i} - X_{p_1,i} \geq 0$, $X_{v_1,i} - X_{p_2,i} \geq 0$, $\ldots$, $X_{v_{j-1},i} - X_{p_j,i} \geq 0$. So $cost_{p,v,i} + cost_{p_1,v_1,i} + \ldots + cost_{p_j,v_j,i} \geq X_{v_j,i} - X_{p,i} = 1 - X_{p,i} \geq 1/2$.

Note also that for any two i-paths $p = (u \rightarrow v)$ and $p' = (u' \rightarrow v')$ which break i, the i-labelled paths to leaves are disjoint (because they are in separate components of i). Therefore each i-path p with lower endpoint v which breaks i in our construction contributes at least 1/2 to the sum $(\sum_{p,v} cost_{p,v,i})$. If rt_i is labelled i (and hence is the root of a component of i), then $X_{rt_i,i} > 1/2$ (corresponding to an edge $(u \rightarrow rt_i)$ in T or to the case $Y_i = 1$). So $2 \times ((\sum_{p,v} cost_{p,v,i}) + X_{rt_i,i}) \geq |\{e = (u \rightarrow v) : c(u) \neq i,\ c(v) = i\}| + Y_i$, and therefore $2\ell \geq |\{e = (u \rightarrow v) : c(u) \neq i,\ c(v) = i\}| + Y_i$. □

It is possible to show that Theorem 3.3 is tight by the following example: let the input topology be a star graph with an even number n of leaves, and suppose that $n/2$ leaves are labelled i_1 and $n/2$ leaves are labelled i_2. Then $\ell = (n/2+1)/2$ by Constraint (3.7). The optimal solution has $\ell = n/2$, arbitrarily close to twice the LP bound. In this example, however, the LP bound is loose, so our analysis of the approximation quality of the algorithm may not be tight.

Theorem 3.3 shows that for any character of an input to the fixed-topology phylogenetic number problem, we can obtain a 2-approximation of the phylogenetic number of that character. Doing this for each character of the input, we obtain a 2-approximation of the phylogenetic number of the input. Also, the same algorithm can be used for the generalized ℓ-phylogeny problem. In particular, we have the following theorem.

Theorem 3.4 *There is an approximation algorithm with approximation ratio* 2 *for the generalized ℓ-phylogeny problem.*

Finally, the paper by Cryan, Goldberg and Phillips [14] contains a proof that there is a polynomial-time algorithm that constructs a 4-phylogeny for any input which has a 3-phylogeny. Since the fixed topology 3-phylogeny problem is known to be $\mathcal{NP}$-complete (see [32]), this is an optimal approximation algorithm for this problem (assuming that $\mathcal{P} \neq \mathcal{NP}$).

3.4 Approximating Polymorphism

We have already mentioned that "language data" in comparative linguistics often comes in the form of polymorphic characters (see Bonet et al. [8], Warnow et al. [65]).

The evolution of polymorphic characters from parent to child can be modelled in terms of mutations, losses and duplications of states between species (see Nei[54]). A *mutation* changes one state into another; a *loss* drops a state from a polymorphic character from parent to child; and a *duplication* replicates a state which subsequently mutates. We associate a cost with each mutation, duplication and loss between a pair of species. In the state-independent model, which we will consider, a loss costs c_l, a mutation costs c_m and a duplication costs c_d, regardless of which states are involved. Following Bonet et al. [8], we insist $c_l \leq c_m \leq c_d$. Let $s_1, s_2 \in S$ and assume s_1 is the parent of s_2. Define $X = c(s_1) - c(s_2)$, and $Y = c(s_2) - c(s_1)$. Then the cost for the character c from s_1 to s_2 is $c_m * |X|$ if $|X| = |Y|$, and is $c_l * [|X| - |Y|] + c_m * |Y|$ if $|X| > |Y|$ and is $c_d * [|Y| - |X|] + c_m * |X|$ if $|Y| > |X|$.

As input we are given a fixed-topology T which has a unique species from S associated with each of its leaves, and label the leaf associated with $s \in S$ with the set of states $c(s)$. The *parsimony problem* is the problem of extending the function c to the internal nodes of T so that the sum of the costs over all edges of T is minimised. In the monomorphic case, as discussed earlier, this problem can be solved in polynomial time [29], though the problem of finding a minimum cost labelling is $\mathcal{NP}$-hard if the input does not include a topology [33]. We will consider the *load problem*, introduced in [8]: calculate a labelling of the internal nodes of a fixed topology T with load at most ℓ and cost at most p. This problem was shown to be $\mathcal{NP}$-hard in [8], even when $c_l = 0$ and the topology T is a binary tree.

An (α, β)-approximation algorithm for the load problem computes a phylogeny with load at most $\alpha\ell$ and cost at most βp provided there is a load-ℓ cost-p phylogeny. This could be called a pseudoapproximation algorithm, since the cost of the best $\alpha\ell$-load phylogeny may be significantly lower than the cost of the best ℓ-load phylogeny. In this section of the chapter, we consider the load problem when $c_l = 0$ and the topology is arbitrary. We extend the results of section 3.3 to obtain, for any $\alpha > 1$, an $(\alpha, \frac{\alpha}{\alpha-1})$-approximation algorithm for the problem. (Note that taking $\alpha = 2$ gives a $(2, 2)$-approximation algorithm.) It is also possible to use the results of Jiang et al. [39] to obtain a $(1, 2)$-approximation algorithm for the load problem with $c_l = 0$, if $c(s)$ contains exactly ℓ states for every leaf species s and every character c in the input; we will explain how to do this at the end of this section.

We first quote the following observation, which was first noted in [8]:

Observation 3.5 *If $c_l = 0$, then if there is a labelling for the topology T which has load ℓ and cost p, then there is also a labelling for T with load ℓ and cost p such that each internal node contains all the states in the subtree rooted at it or else has load ℓ. Therefore, there is a labelling which has load ℓ and cost p, and which contains no duplications.*

Therefore to approximate the load we only need to consider the labellings where each internal node contains all the states in the subtree rooted at it or else has load ℓ. We begin by presenting an ILP which provides an exact solution for the load problem. We then use the solution to the linear-programming relaxation of this ILP to compute an $(\alpha, \frac{\alpha}{\alpha-1})$-approximation for the problem. The integer program $\mathcal{P}$ uses the variables $X_{x,i}$, for each node x of the fixed-topology T and each state $i \in \{1, \ldots, r\}$, cost variables $cost_{e,i}$ for each edge $e \in E(T)$ and each state $i \in \{1, \ldots, r\}$ and the total cost variable $cost_e$ for each edge e. These variables have the following interpretation:

$$
\begin{aligned}
X_{x,i} &= \begin{cases} 1 & \text{if state } i \text{ is in } c(x) \\ 0 & \text{otherwise} \end{cases} \\
cost_{e,i} &= \begin{cases} 1 & \text{if } i \in c(v) \text{ and } i \notin c(u), \text{ for } e = (u \rightarrow v) \\ 0 & \text{otherwise} \end{cases} \\
cost_e &= \textstyle\sum_{i=1}^{r} cost_{e,i}
\end{aligned}
$$

The ILP $\mathcal{P}$ is then defined as:

$$\text{minimize } p$$

subject to

$$
\begin{aligned}
X_{x,i} &= 1 && \text{for each leaf } x \in V(T),\ \forall i \in c(x) && (3.9)\\
\sum_{i=1}^{r} X_{x,i} &\leq \ell && \forall x \in V(T) && (3.10)\\
cost_{e,i} &\geq 0 && \forall e \in E(T),\ i = 1, \ldots, r && (3.11)\\
cost_{e,i} &\geq X_{v,i} - X_{u,i} && \forall e = (u \to v) \in E(T),\ i = 1, \ldots, r && (3.12)\\
cost_e &= \sum_{i=1}^{r} cost_{e,i} && \forall e = (u \to v) \in E(T) && (3.13)\\
\sum_{e \in E(T)} cost_e &\leq p/c_m && && (3.14)\\
X_{x,i}, cost_{e,i} &\in \{0, 1\} && && (3.15)
\end{aligned}
$$

The integer program $\mathcal{P}$ solves the load problem. Since we are interested in using the linear relaxation of this program to obtain an approximation algorithm, we will restrict ourselves to showing that when we solve $\mathcal{P}$ with parameter ℓ, the optimal value of p is less than or equal to the cost of the best load-ℓ solution to the fixed topology problem.

Lemma 3.6 *Let S be a species set, T be a fixed topology and $c : S \to (2^{\{1,\ldots,r\}} - \{\})$ be a polymorphic character on S. If the internal nodes of T can be labelled with subsets of $\{1, \ldots, r\}$ to create a phylogeny for c with load ℓ and cost p, then there is a feasible solution for the Integer program $\mathcal{P}$ for this value of ℓ and p.*

Proof: Because of Observation 3.5, we can assume that in the load-ℓ, cost c phylogeny, for each internal node u in $V(T)$, either $c(v) \subset c(u)$ for every child v of u, or else $|c(u)| = \ell$. Therefore the cost of this phylogeny is $\sum_{(u \to v) \in E(T)} (c_m * |c(v) - c(u)|) = p$. Assign values to the $X_{u,i}$ variable for each internal node u and to the $cost_{e,i}$ variable for each edge $e = (u \to v)$ as follows:

$$X_{u,i} = \left\{ \begin{array}{ll} 1 & \text{if } i \in c(u) \\ 0 & \text{otherwise} \end{array} \right\}$$
$$cost_{e,i} = \left\{ \begin{array}{ll} 1 & \text{if } i \in c(v) - c(u) \\ 0 & \text{otherwise} \end{array} \right\}$$

This assignment satisfies Constraints (3.15), (3.10), (3.11) and (3.12) of $\mathcal{P}$. Constraint (3.9) is automatically satisfied, and Constraint (3.13) is definitional. Also, $c_m * cost_e = c_m * |c(v) - c(u)|$ for every $e = (u \to v)$ by definition of the $cost_{e,i}$, and therefore $\sum_{e \in E(T)} c_m * cost_e = p$, and Constraint (3.14) is satisfied. $\square$

Once again, since integer linear programming is $\mathcal{NP}$-hard, we solve the linear-programming relaxation $\mathcal{LP}$ of $\mathcal{P}$, which consists of all the constraints of $\mathcal{P}$ except that Constraint (3.15) is replaced with

$$0 \leq X_{x,i}, cost_{e,i} \leq 1 \qquad (3.15').$$

Theorem 3.7 *Suppose there is a solution for the linear program $\mathcal{LP}$. Then we can assign states to the internal nodes of input tree T such that the resulting phylogeny for c has load $\alpha\ell$ and cost no more than $\left(\frac{\alpha}{\alpha-1}\right)p$.*

Proof: We assign states to the internal nodes of the fixed topology from the leaves upwards. For each internal node $u \in V(T) - L(T)$, consider the set $R(u) = \cup_{(u \to v) \in E(T)} c(v)$. If $|R(u)| \leq \alpha\ell$, then define $c(u) = R(u)$. If $|R(u)| > \alpha\ell$ then choose the $\alpha\ell$ states i of $R(u)$ which have the greatest $X_{u,i}$ values. By definition, this assignment of states to the internal nodes of T has load $\alpha\ell$.

To show that the cost of this assignment is no more than $\left(\frac{\alpha}{\alpha-1}\right)p$, note that the cost on an edge $e = (u \to v) \in E(T)$ is $c_m * |c(v) - c(u)|$, as $|c(v) - c(u)|$ is the number of mutations on e. Our assignment guarantees that if $|c(u)| < \alpha\ell$ then $c(u) \supset c(v)$, which implies $c_m * |c(v) - c(u)| = 0$, so we need only consider edges whose upper endpoint has full load. Suppose $|c(u)| = \alpha\ell$ and $i \in c(v) - c(u)$. Then, by construction of the phylogeny, there is a downwards path from v to some leaf w which has $i \in c(v')$ at every node along the path, including w. Suppose this path is $e_1 = (v \to v_1)$, $e_2 = (v_1 \to v_2)$, ..., $e_j = (v_{j-1} \to w)$. By the constraints of the linear program, $cost_{e,i} + cost_{e_1,i} + \ldots + cost_{e_j,i} \geq (X_{v,i} - X_{u,i}) + (X_{v_1,i} - X_{v,i}) + \ldots + (X_{w,i} - X_{v_{j-1},i}) = X_{w,i} - X_{u,i}$, and as w is a leaf and $i \in c(w)$, this is $1 - X_{u,i}$. Then, since $i \notin c(u)$, and the $\alpha\ell$ states in $c(u)$ were chosen to have the greatest $X_{u,i}$ values, we know $X_{u,i} \leq \ell/(\alpha\ell + 1)$. Therefore $cost_{e,i} + cost_{e_1,i} + \ldots + cost_{e_j,i} \geq ((\alpha - 1)\ell + 1)/(\alpha\ell + 1)$. Furthermore, the costs $cost_{e,i}$, $cost_{e_1,i}$, $\ldots cost_{e_j,i}$ will not be allocated to any other mutation to i, because any mutation occurring above u will not have an unbroken path in i intersecting with any of the edges $e, e_1, \ldots, e_j$. So every mutation along an edge $e = (u \to v) \in T$ with $|c(u)| = \alpha\ell$ contributes at least $((\alpha - 1)\ell + 1)/(\alpha\ell + 1)$ to the sum $\sum_{e \in E(T)} cost_e$ in our linear program. Hence $p/c_m \geq \sum_{e \in E(T)} cost_e \geq \sum_{e=(u \to v) \in E(T)} |c(v) - c(u)| \left(\frac{(\alpha-1)\ell+1}{\alpha\ell+1}\right)$, so the cost $\sum_{e=(u \to v) \in E(T)} c_m * |c(v) - c(u)| \leq (\alpha/(\alpha - 1)) * p$. □

Finally, we will explain why the results of Jiang et al. [39] can be used to obtain a $(1, 2)$-approximation algorithm for instances of this problem that satisfy $|c(s)| = \ell$ for every leaf s. Denote the cost of an optimal load-ℓ labelling by p. By Observation 3.5, we can assume that this optimal labelling assigns exactly ℓ states to every internal node of the fixed-topology. Then, for every edge $e = (u \to v)$ in such a labelling, the cost on e is exactly $c_m * |c(u) - c(v)|$. The triangle inequality holds for this cost measure when every node in the tree is labelled by exactly ℓ states. Then Lemma 3 from [39] implies that every fixed-topology has some labelling in which every internal node u has the label of one if its child nodes (called a lifted tree), such that the cost of this labelling is at most $2p$. Also, the cost measure for the load problem with $c_l = 0$ allows us to use dynamic programming to calculate the optimal lifted tree in polynomial time. We use the same algorithm as Jiang et al. Let u be an internal node. Assume that for every child v of u, and every leaf s in the subtree rooted at v, we have *one* labelling of $T(v)$ that labels v with $c(s)$ and that is optimal, given that $c(s)$ must label v. Let $cost(v, s)$ be the cost of this optimal labelling, summed over the edges in the subtree rooted at v. Then for every $c(s)$ that labels a leaf of $T(u)$, we can calculate a labelling for $T(u)$ that is optimal, given that $c(s)$ must label u. We find the labelling by defining the labelling $T(v)$ for each child v by choosing s' so that

$$c_m|c(s) - c(s')| + cost(v, s')$$

is minimized, where this minimum is taken over all leaves s' in $T(v)$. The labelling constructed by this algorithm will either be the optimal lifted tree, or will have cost less than the optimal lifted tree. Therefore, we have a $(1, 2)$-approximation algorithm for this special case of the load problem with $c_l = 0$.

The reason that the argument does not generalize when the leaves are allowed to have load less than ℓ is because the triangle inequality does not really hold for the cost measure. This is because of the asymmetry of losses and duplications. For example, suppose we are given an instance of the problem for load 4, and the topology is a tree with a root and two leaves x and y, and that $c(x) = \{a, b\}$ and $c(y) = \{c, d\}$. Then the optimal lifted tree has cost $2c_m$, even though labelling the root with $\{a, b, c, d\}$ gives cost 0. Although the solution is obvious in this case, there does not seem to be an obvious way of adapting the algorithm of Jiang et al. to solve the problem for more interesting inputs.

Chapter 4

Conclusions and Further Work

The first point that should be made is that although the research presented in Chapter 2 gives a positive result for the problem of PAC-learning the distribution of Two-State Markov Evolutionary Trees, it also provides new lower bounds on the number of samples required to reconstruct the topology of a Two-State MET. Let M be a Two-State MET that satisfies Steel's conditions, and remember that in the Two-State case these conditions are

(i) $\rho_0 \in (0, 1)$, where ρ_0 is the probability that the root is 0;

(ii) $0 < |1 - e_0 - e_1| < 1$ for every edge e.

Steel has shown that under these conditions, all leaf-pair covariances are non-zero, and that given the exact values of these covariances, the topology of M can be recovered (See Steel [58]).

Now define $\alpha' > 0$ to be the maximum value such that the leaf connectivity graph is connected for the threshold α', when the exact covariances are used. Then Lemma 2.50 shows that there must be some edge e in the topology of M such that $|\text{cov}(x, y)| \leq \alpha'$ for *every* pair of leaves x and y that are connected through e (otherwise the leaf connectivity graph would be connected for some threshold greater than α'). Then by Lemma 2.49, there is another MET M' in which e is replaced by a "cut edge" such that the variation distance between M and M' is at most $\alpha' n^2$. Remember that the location of a cut edge in the topology cannot be inferred from the distribution of M'. The argument of Farach and Kannan given in Subsection 1.3.2 implies that we would need to take at least $\Omega(1/(\alpha' n^2))$ samples from the distribution of M to distinguish between M and M'. Therefore it is not possible to determine the location of e in the topology of M without taking $\Omega(1/(\alpha' n^2))$ samples.

As further work on this problem, it would be interesting to obtain a PAC-learning result for the class of j-State METs, for $j > 2$. Steel has already shown that the topology of any j-State MET can be inferred from the exact distribution, as long as $\Lambda(e) \in (0, 1)$ for every edge e in the tree. However, we do not even know how to use the exact distribution of a j-State MET to reconstruct transition probabilities for the edges of a j-State MET, even when the multiplicative weights are non-zero. The problem of learning j-State METs seems to be considerably more difficult than the Two-State case. Another interesting question that might be simpler than the PAC-learnability of j-State METs is the problem of learning a linear mixture of j product distributions that was described in Subsection 2.1.4.

There are also some open problems related to the research presented in Chapter 3 that are of interest. One question that is still open is whether or not there is a polynomial-time algorithm with a constant approximation ratio for the phylogenetic number problem, when the input does not include a fixed topology.

Bibliography

[1] N. Abe, J. Takeuchi, and N. K. Warmuth. "Polynomial Learnability of Probabilistic Concepts with Respect to the Kullback-Leibler Divergence". In *Proceedings of the fourth Annual Workshop on Computational Learning Theory*, pages 277–289, (1991).

[2] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. "On the Approximability of Numerical Taxonomy (Fitting Distances by Tree Metrics)". *SIAM Journal on Computing*, **28**(3):1073–1085, (1999).

[3] R. Agarwala and D. Fernández-Baca. "A Polynomial-Time Algorithm for the Perfect Phylogeny Problem when the number of Character States is Fixed". *SIAM Journal on Computing*, **23**(6):1216–1224, (1994).

[4] A. Ambainis, R. Desper, M. Farach, and S. Kannan. "Nearly Tight Bounds on the Learnability of Evolution". In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS '97)*, pages 524–533, Miami Beach, Florida, (1997).

[5] M. Anthony and N. Biggs. *"Computational Learning Theory"*, volume 30 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, (1992).

[6] H. J. Bandelt and A. Dress. "Reconstructing the shape of a Tree from Observed Dissimilarity Data". *Adv. Appl. Math.*, **7**:309–343, (1987).

[7] H. Bodlaender, M. Fellows, and T. Warnow. "Two Strikes Against Perfect Phylogeny". In *Proceedings of the 19th International Congress on Automata, Lan-*

guages and Programming, Springer-Verlag Lecture Notes in Computer Science, pages 273–287, (1992).

[8] M. Bonet, C. Phillips, T. Warnow, and S. Yooseph. "Constructing Evolutionary Trees in the Presence of Polymorphic Characters". *SIAM Journal on Computing*, **29**(1):103–131, (1999).

[9] I. Borosh and L. Treybig. "Bounds on Positive Integral Solutions of Linear Diophantine Equations". In *Proceedings of the American Mathematical Society*, volume **55**, (1976).

[10] J. C. Brown. "What the Heck is a Gene?". Available from `http://falcon.cc.ukans.edu/~jbrown/gene.html`, (1995).

[11] P. Buneman. "The Recovery of Trees from Measures of Dissimilarity". In F. R. Hodson, D. G. Kendall, and P. Tautu, editors, *Mathematics in the Archeological and Historical Sciences*, pages 387–395, (1971).

[12] J. A. Cavender. "Taxonomy with Confidence". *Math. Biosci.*, **40**:271–280, (1978).

[13] T. H. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, (1991).

[14] M. Cryan, L. A. Goldberg, and C. A.Phillips. "Approximation Algorithms for the Fixed-Topology Phylogenetic Number Problem". *Algorithmica*, **25**(2):311–329, (1999).

[15] M. Cryan, L. A. Goldberg, and P. W. Goldberg. "Evolutionary Trees can be Learned in Polynomial-Time in the Two-State General Markov Model". In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '98)*, pages 436–445, Palo Alto, California, (1998).

[16] M. Csűrös and M.-Y. Kao. "Fast Recovery of Evolutionary Trees through Harmonic Greedy Triplets". Available from `http://www.cs.yale.edu/~kao-ming-yang/papers.html`. An earlier version appeared as "Recovering evolutionary trees through Harmonic Greedy Triplets" in *Proceedings of the 10th An-*

nual ACM-SIAM Symposium on Discrete Algorithms (SODA '99), pages 261–270, (1999).

[17] M. Csűrös and M.-Y. Kao. "Reconstructing Evolutionary Trees in a General Markov Model". Available from `http://www.cs.yale.edu/~csuros-miklos/papers.html`, (1999).

[18] I. Csiszár. "Information Type Measures of Difference of Probability Distributions and Indirect Observations". *Studia Sci. Math. Hungar.*, **2**:229–318, (1967).

[19] N. L. David Haussler, Michael Kearns and M. K. Warmuth. "Equivalence of Models for Polynomial Learnability". *Information and Computation*, **95**(2):129–161, (1991).

[20] W. Day. "Computationally Difficult Parsimony Problems in Phylogenetic Systematics". *Journal of Theoretical Biology*, **103**, (1983).

[21] W. Day, D. Johnson, and D. Sankoff. "The Computational Complexity of Inferring Phylogenies by Parsimony". *Mathematical biosciences*, **81**, (1986).

[22] W. Day and D. Sankoff. "Computational Complexity of Inferring Phylogenies by Compatibility". *Systematic Zoology*, **35(2)**:224–229, (1986).

[23] P. L. Erdös, M. A. Steel, L. A. Székely, and T. J. Warnow. "A Few Logs Suffice to Build (Almost) all Trees (I)". *Random Structures and Algorithms*, **14**(2):153–184, (1999).

[24] P. L. Erdös, M. A. Steel, L. A. Székely, and T. J. Warnow. "A Few Logs Suffice to Build (Almost) all Trees (II)". *Theoretical Computer Science*, **221**(1–2):77–118, (1999).

[25] M. Farach and S. Kannan. "Efficient Algorithms for Inverting Evolution". In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 230–236, Philadelphia, Pennsylvania, (1996).

[26] J. S. Farris. "A Probability Model for Inferring Evolutionary Trees". *Systematic Zoology*, **22**:250–256, (1973).

[27] J. Felsenstein. "Cases in which Parsimony or Compatibility methods will be Positively Misleading". *Systematic Zoology*, **22**:240–249, (1978).

[28] J. Felsenstein. "Statistical Inference of Phylogenies". *Journal of the Royal Statistical Society*, **146**(Part 3):246–272, (1983).

[29] W. Fitch. "Towards Defining the Course of Evolution: Minimum Change for a Specified Tree Topology". *Systematic Zoology*, 20, (1971).

[30] Y. Freund and Y. Mansour. "Estimating a Mixture of Two Product Distributions". In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 53–62, (1999).

[31] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, (1979).

[32] L. A. Goldberg, P. Goldberg, C. Phillips, E. Sweedyk, and T. Warnow. "Minimizing Phylogenetic Number to find Good Evolutionary Trees". *Discrete Applied Mathematics*, **71**(1–3):111–136, (1996).

[33] R. L. Graham and L. R. Foulds. "Unlikelihood that Minimal Phylogenies for a Realistic Biological Study can be Constructed in Reasonable Computational Time". *Mathematical Biosciences*, **60**:133–142, (1982).

[34] G. R. Grimmett and D. R. Stirzaker. *"Probability and Random Processes"*. Oxford University Press, (1992).

[35] D. Gusfield. "The Steiner Tree Problem in Phylogeny". Technical Report 332, Department of Computer Science, Yale university, (1984).

[36] J. A. Hartigan. "Minimum Mutation Fits to a Given Tree". *Biometrics*, **29**:53–65, (1973).

[37] N. Hawkes. "Computer Tale gets Closer to Chaucer". *The Times*, 27th August, 1998.

[38] M. Hendy, D. Penny, and M. Steel. "A Discrete Fourier Analysis for Evolutionary Trees". *Proceedings of the National Academy of Sciences USA*, **91**:3339–3343, (1994).

[39] T. Jiang, E. Lawler, and L. Wang. "Aligning Sequences via an Evolutionary Tree: Complexity and Approximation". In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC '94)*, pages 760–769, Montréal, Québec, Canada, (1994). (A later version of this paper appeared as: L. Wang, T. Jiang and E.L. Lawler. "Approximation Algorithms for Tree Alignment with a Given Phylogeny". *Algorithmica*, **16**:302–315, (1996)).

[40] T. H. Jukes and C. R. Cantor. *"Mammalian Protein Metabolism"*, volume **3**, chapter "Evolution of Protein Molecules", pages 21–132. "Academic Press", (1969).

[41] S. K. Kannan and T. J. Warnow. "Triangulating Three-Colored Graphs". *SIAM Journal on Discrete Mathematics*, **5**(2):249–258, (1992).

[42] S. K. Kannan and T. J. Warnow. "Inferring Evolutionary History from DNA Sequences". *SIAM Journal on Computing*, **23**(4):713–737, (1994).

[43] R. Karp. "Reducibility among Combinatorial Problems". In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, (1972).

[44] R. Karp. "Mapping the Gemone: Some Combinatorial Problems Arising in Molecular Biology". In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC '93)*, pages 278–285, San Diego, California, (1993).

[45] K. Kashyap and S. Subas. "Statistical Estimation of Parameters in a Phylogenetic Tree Using a Dynamic Model of the Substitutional Process". *Journal of Theoretical Biology*, **47**:74–101, (1974).

[46] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. "On the Learnability of Discrete Distributions". In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pages 273–282, (1994).

[47] M. Kearns and U. Vazirani. *"An Introduction to Computational Learning Theory"*. MIT Press, (1994).

[48] J. H. B. Kemperman. *On the Optimum Rate of Transmitting Information*. Springer-Verlag, New York, (1967).

[49] M. Kimura. "Estimation of Evolutionary Distances between Homologous Nucleotide Sequences". *Proceedings of the National Academy of Sciences, USA*, **78**(1):454–458, (1981).

[50] L. Kou, G. Markowsky, and L. Berman. "A Fast Algorithm for Steiner Trees". *Acta Informatica*, 15, (1981).

[51] S. Kullback. "A Lower Bound for Discrimination in terms of Variation". *IEEE Trans. Infor. Theory.*, IT-13:126–127, (1967).

[52] C. McDiarmid. "On the Method of Bounded Differences". *London Mathematical Society Lecture Note Series 141*, pages 148–188, (1989).

[53] B. L. Monroe and C. G. Sibley. *"A World Checklist of Birds"*. Yale University Press, New Haven, (1993).

[54] M. Nei. *"Molecular Evolutionary Genetics"*. Columbia University Press, New York, (1987).

[55] J. Neyman. "Molecular Studies of Evolution: a source of Novel Statistical Problems". *Statistical Decision Theory and Related Topics*, pages 1–27, (1971).

[56] P. A. Pevzner and M. S. Waterman. "Open Combinatorial Problems in Computational Molecular Biology". In *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems*, pages 158–172. IEEE Computer Society Press, (1995).

[57] M. Steel. "The Complexity of Reconstructing Trees from Qualitative Characters and Subtrees". *Journal of Classification*, **9**:91–116, (1992).

[58] M. Steel. "Recovering a Tree from the Leaf Colourations it Generates under a Markov Model". *Appl. Math. Lett.*, **7**(2):19–24, (1994).

[59] D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis. "Phylogeny Reconstruction". In D. Hillis and C. Moritz, editors, *Molecular Systematics*, pages 407–514. Sinaur Associates Inc., (1990).

[60] L. G. Valiant. "A Theory of the Learnable". *Communications of the ACM*, pages 1134–1142, November (1984).

[61] L. G. Valiant. "Learning Disjunctions of Conjunctions". In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 560–566, (1985).

[62] L. Wang and D. Gusfield. "Improved Approximation Algorithms for Tree Alignment". *Journal of Algorithms*, **25**(2):255–273, (1997).

[63] L. Wang, T. Jiang, and D. Gusfield. "A more Efficient Approximation Scheme for Tree Alignment". In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, (1997).

[64] T. H. Wareham. "On the Computational Complexity of Inferring Evolutionary Trees". Technical Report 9301, Department of Computer Science, Memorial University of Newfoundland, (1993).

[65] T. Warnow, D. Ringe, and A. Taylor. "A Character Based Method for Reconstructing Evolutionary History for Natural Languages". In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '96)*, Atlanta, Georgia, (1996).

[66] M. Waterman, T. Smith, M. Singh, and W. Beyer. "Additive Evolutionary Trees". *Journal of Theoretical Biology*, **64**:199–213, (1977).