20

Applications of Interval Methods to Phylogenetics

Raazesh Sainudiin

Ruriko Yoshida

When statistical inference is conducted in a maximum likelihood (ML) framework as discussed in Chapter 1, we are interested in the global maximum of the likelihood function over the parameter space. In practice we settle for a local optimization algorithm to numerically approximate the global solution since explicit analytical solutions for the maximum likelihood estimates (MLEs) are typically difficult to obtain for unrooted trees with three or more leaves. See Chapter 18 and the references therein for algebraic approaches to solving such ML problems. In this chapter we will take a rigorous numerical approach to this phylogenetic problem via interval methods. We accomplish this by first constructing an interval extension of the recursive formulation for the likelihood function of a Markov model of DNA evolution on unrooted phylogenetic trees with a fixed topology. Then we use an adaptation of a widely applied global optimization algorithm using interval analysis for the phylogenetic context to rigorously enclose the ML value as well as the MLEs. The MLEs are the set of branch lengths of the phylogenetic tree. The method is applied to enclose the most likely 2 and 3 taxa trees under the *Jukes-Cantor model* of DNA evolution. The method is general and can provide rigorous estimates when coupled with standard phylogenetic algorithms. Solutions obtained with such methods are equivalent to computer-aided proofs unlike solutions obtained with conventional numerical methods.

Statistical inference procedures that obtain MLEs through conventional numerical methods may suffer from several major sources of errors. To fully appreciate the sources of errors we need some understanding of a number screen. Computers can only support a finite set of numbers, usually represented as fixed-length binary floating point quantities of the form, $x = \pm m \cdot 2^e = \pm 0.m \cdot 2^e$, where $m = (m_1 m_2 \dots m_p)$ is the signed mantissa ($m_1 = 1$, $m_i \in \{0,1\}, \forall i, 1 < i \leq p$) with base 2, p is the precision, and e is the exponent ($\underline{e} \leq e \leq \overline{e}$) [IEEE Task P754, 1985]. Thus, the smallest and largest machine-representable numbers in absolute value are $\underline{x} = 0.10\dots0 \cdot 2^{\underline{e}}$ and $\overline{x} = 0.11\dots1 \cdot 2^{\overline{e}}$, respectively. Therefore, the binary floating-point system of most machines $\mathcal{R} = \mathcal{R}(2, p, \underline{e}, \overline{e})$ is said to form a screen of the real numbers in the interval $[-\overline{x}, +\overline{x}]$ with 0 uniquely represented by $0.00\dots0 \cdot 2^{\underline{e}}$. When numerical inference procedures rely on inexact computer arithmetic with a number screen

they may suffer from at least five types of errors: *roundoff error*, the difference between computed and exact result [Cuyt *et al.*, 2001, Loh and Walster, 2002]; *truncation error*, from having to truncate an infinite sequence of operations; *conversion error*, inability to machine-represent decimals with infinite binary expansion; and *ill-posed statistical experiment*, presence of unknown nonidentifiable subspaces.

The verified global optimization method [Hansen, 1980] sketched below rigorously encloses the global maximum of the likelihood function through interval analysis [Moore, 1967]. Such interval methods evaluate the likelihood function over a continuum of points including those that are not machine-representable and account for all sources of errors described earlier. In this chapter we will see that interval methods, in contrast to heuristic local search methods, can enclose the global optimum with guaranteed accuracy by exhaustive search within any compact set of the parameter space. We begin with a brief introduction to analysis in the space of all compact real intervals, our basic platform for rigorous numerics.

20.1 Brief introduction to interval analysis

Lowercase letters denote *real numbers*, e.g., $x \in \mathbb{R}$. Uppercase letters represent compact *real intervals*, e.g., $X = [\underline{x}, \overline{x}] = [\inf(X), \sup(X)]$. Any compact interval X belongs to the set of all compact real intervals $\mathbb{IR} := \{[a, b] : a \leq b,\ a, b \in \mathbb{R}\}$. The *diameter* and the *midpoint* of X are $d(X) := \overline{x} - \underline{x}$ and $m(X) := (\underline{x} + \overline{x})/2$, respectively. The *smallest* and *largest absolute value* of an interval X are the real numbers given by $\langle X \rangle := \min\{|x| : x \in X\} = \min\{|\underline{x}|, |\overline{x}|\}$, if $0 \notin X$, and 0 otherwise, and $|X| := \max\{|x| : x \in X\} = \max\{|\underline{x}|, |\overline{x}|\}$, respectively. The *absolute value* of an interval X is $|X|_{[\,]} := \{|x| : x \in X\} = [\langle X \rangle, |X|]$. The *relative diameter* of an interval X, denoted by d_{rel}, is the diameter $d(X)$ itself if $0 \in X$ and $d(X)/\langle X \rangle$ otherwise. An interval X with zero diameter is called a *thin interval* with $\underline{x} = \overline{x} = x$ and thus $\mathbb{R} \subset \mathbb{IR}$. The *hull* of two intervals is $X \underline{\cup} Y := [\min\{\underline{x}, \underline{y}\}, \min\{\overline{x}, \overline{y}\}]$. By the notation $X \Subset Y$, it is meant that X is *strictly contained* in Y, i.e., $\underline{x} > \underline{y}$ and $\overline{x} < \overline{y}$. No notational distinction is made between a real number $x \in \mathbb{R}$, or a real vector $x = (x_i, \ldots, x_n)^T \in \mathbb{R}^n$ and between a real interval X and a *real interval vector* or *box* $X = (X_1, \ldots, X_n)^T \in \mathbb{IR}^n$, i.e., $X_i = [\underline{x_i}, \overline{x_i}] = [\inf(X_i), \sup(X_i)] \in \mathbb{IR}$, where $i = 1, \ldots, n$. The diameter, relative diameter, midpoint, and hull operations for boxes are defined componentwise to yield vectors. The maximum over the components is taken to obtain the maximal diameter and the maximal relative diameter, $d_\infty(X) = \max_i d(X_i)$ and $d_{rel,\infty}(X) = \max_i d_{rel}(X_i)$, respectively, for a box X. Also $\mathbb{IR}$ under the metric $\mathfrak{h}$, given by $\mathfrak{h}(X, Y) := \max\{|\underline{x} - \underline{y}|, |\overline{x} - \overline{y}|\}$, is a complete metric space. Convergence of a sequence of intervals $\{X^{(i)}\}$ to an interval X under the metric $\mathfrak{h}$ is equivalent to the sequence $\mathfrak{h}(X^{(i)}, X)$ approaching 0 as i approaches ∞, which in turn is equivalent to both $\underline{x}^{(i)} \to \underline{x}$ and $\overline{x}^{(i)} \to \overline{x}$. Continuity and differentiability of a function $F : \mathbb{IR}^n \to \mathbb{IR}^k$ are defined in the usual

way. Let $\circ$ denote a binary operation. An interval arithmetic (IA) operation $X \circ Y := \{x \circ y : x \in X, y \in Y\}$ thus yields the set containing the result of the operation performed on every real pair $(x, y) \in (X, Y)$. Although there are uncountably many real operations to consider during an interval operation, the properties of continuity, monotonicity, and compactness imply that:

$$X + Y = [\underline{x} + \underline{y}, \overline{x} + \overline{y}],\ X \cdot Y = [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}],$$
$$X - Y = [\underline{x} - \overline{y}, \overline{x} - \underline{y}],\ X/Y = X \cdot [1/\overline{y}, 1/\underline{y}],\ 0 \notin Y.$$

This definition of IA leads to the *property of inclusion isotony* which stipulates that $X \circ Y$ contains $V \circ W$ provided $V \subseteq X$ and $W \subseteq Y$. Note that continuous functions of compact sets are necessarily inclusion isotonic. The identity elements of $+$ and $\cdot$ are the thin intervals 0 and 1, respectively. Multiplicative and additive inverses do not exist except when X is also thin. IA is commutative and associative but not distributive. However, $X \cdot (Y + Z) \subseteq (X \cdot Y) + (X \cdot Z)$. For any real function $f(x) : \mathbb{R}^n \to \mathbb{R}$ and some box $X \in \mathbb{IR}^n$, let the image of f over X be denoted by $f(X) := \{f(x) : x \in X\}$. Inclusion isotony also holds for interval evaluations that are compositions of arithmetic expressions and the elementary functions. When real constants, variables, and operations in f are replaced by their interval counterparts, we obtain $F(X) : \mathbb{IR}^n \to \mathbb{R}$, the natural interval extension of f. Guaranteed enclosures of the image of $f(X)$ are obtained by $F(X)$ due to the *inclusion property*, which states that if $x \in X$, then $f(x) \in F(X)$. The natural interval extension $F(X)$ often overestimates the image $f(X)$, but can be shown under mild conditions to linearly approach the image as the maximal diameter of the box X goes to zero, i.e., $\mathfrak{h}(F(X), f(X)) \leq \alpha \cdot d_\infty(X)$ for some $\alpha \geq 0$. This implies that a partition of X into smaller boxes $\{X^{(1)}, \ldots, X^{(m)}\}$ gives better enclosures of $f(X)$ through the union $\bigcup_{i=1}^{m} F(X^{(i)})$. This is illustrated by the gray rectangles of a given shade that enclose the image of the nonlinear function shown in Figure 20.1. The darker the shade of the image enclosure the finer the corresponding partition on the domain $[-10, 6]$.

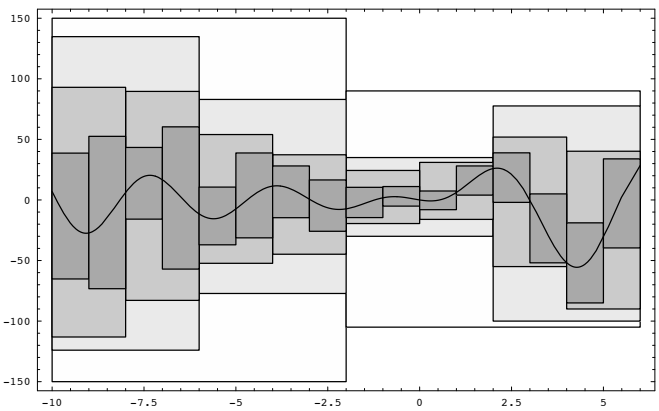

Fig. 20.1. *Image enclosure of* $-\sum_{k=1}^{5} k\,x\,\sin\left(\frac{k(x-3)}{3}\right)$ *linearly tightens with mesh*

Some interval extensions of f are better at enclosing the true image than others. Figure 20.2 exhibits three functions. These functions are equivalent as real maps but their natural interval extensions yield successively tighter range enclosures: $F^{(1)} \Subset F^{(2)} \Subset F^{(3)}$. Note that $F^{(3)} \subset F^{(2)}$ since $X^2 \subset X \cdot X$ in IA. If X appears only once in the expression and all parameters are thin intervals, it was shown by [Moore, 1979] that the natural interval extension does indeed yield a tight enclosure, i.e., $F(X) = f(X)$. In general, we can obtain tighter enclosures by minimizing the occurrence of X in the expression.

There is another way to improve the tightness of the image enclosure. Let $\nabla f(x)$ and $\nabla^2 f(x)$ denote the gradient and Hessian of f, respectively. Now let $\nabla F(x)$ and $\nabla^2 F(x)$ represent their corresponding interval extensions. A better enclosure of $f(X)$ over all $x \in X$ with a fixed center $c = m(X) \in X$ is possible for a differentiable f with the following centered form:

$$f(x) = f(c) + \nabla f(b) \cdot (x - c) \in f(c) + \nabla f(X) \cdot (x - c) \subseteq F_c(X),$$

for some $b \in X$ and where $F_c(X) := f(c) + \nabla F(X) \cdot (X - c)$. $F_c(X)$ is the interval extension of the centered form of f with center $c = m(X)$ and decays quadratically to $f(X)$ as the maximal diameter of X approaches 0. Next we introduce automatic differentiation (AD) to obtain gradients, Hessians, and their enclosures for a twice-differentiable f.

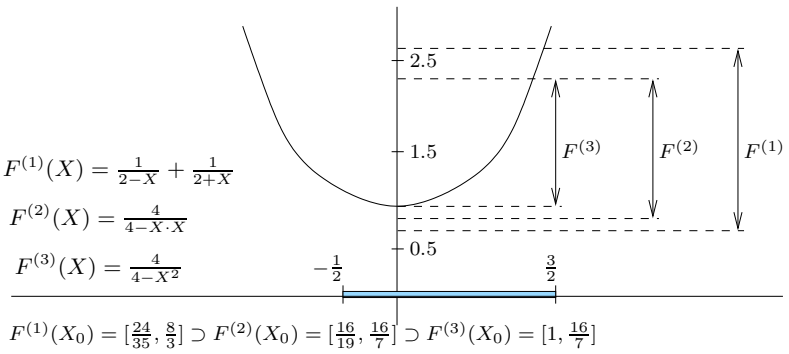

Fig. 20.2. *Extension-specific dependence of image enclosures*

When it becomes too cumbersome or impossible to explicitly compute $\nabla f(x)$ and $\nabla^2 f(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$, we may employ a Hessian differentiation arithmetic, also known as second-order AD [Rall, 1981]. This approach defines an arithmetic on a set of ordered triples. Consider a twice-continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with the gradient vector $\nabla f(x) := (\partial f(x)/\partial x_1, \ldots, \partial f(x)/\partial x_n)^T \in \mathbb{R}^n$, and Hessian matrix $\nabla^2 f(x) := ((\partial^2 f(x)/\partial x_i \partial x_j))_{i,j=\{1,\ldots,n\}} \in \mathbb{R}^{n \times n}$. For every f, consider its corresponding ordered triple $(\, f(x), \nabla f(x),\ \nabla^2 f(x)\,)$. The ordered triples corresponding to a constant function, $c(x) = c : \mathbb{R}^n \to \mathbb{R}$, and a component identifying function (or variable), $I_j(x) = x_j : \mathbb{R}^n \to \mathbb{R}$, are $(\, c,\ 0,\ 0\,)$ and $(\, x_j,\ e^{(j)},\ 0\,)$, respectively, where $e^{(j)}$ is the j-th unit vector and the 0's are additive identities in their appropriate spaces. To perform an elemen-

tary operation $\circ \in \{+, -, \cdot, /\}$ on a pair of such triples to obtain another, as in $(\, h(x),\ \nabla h(x),\ \nabla^2 h(x)\,) := (\, f(x),\ \nabla f(x),\ \nabla^2 f(x)\,) \circ (\, g(x),\ \nabla g(x),\ \nabla^2 g(x)\,)$, or to compose the triples of two elementary functions we use the chain rule of Newtonian calculus. The AD process may be extended from real functions to interval-valued functions. By replacing the real x's above by interval X's and performing all operations in the real IA with the interval extension F of f, we can rigorously enclose the components of the triple $(\, F(X),\ \nabla F(X),\ \nabla^2 F(X)\,)$ through an interval-extended Hessian differentiation arithmetic so that for every $x \in X \in \mathbb{IR}^n$, $f(x) \in F(X) \in \mathbb{IR}$, $\nabla f(x) \in \nabla F(X) \in \mathbb{IR}^n$, and $\nabla^2 f(x) \in \nabla^2 F(X) \in \mathbb{IR}^{n \times n}$. We can now apply interval AD to find the roots of nonlinear functions.

The interval version of Newton method computes an enclosure of the zero x^* of a continuously differentiable function $f(x)$ in the interval X through the following dynamical system in $\mathbb{IR}$:

$$X^{(j+1)} = \left(m(X^{(j)}) - \frac{f(m(X^{(j)}))}{F'(X^{(j)})} \right) \cap X^{(j)}, \qquad j = 0, 1, 2, \ldots$$

In this system $X^{(0)} = X$, $F'(X^{(j)})$ is the enclosure of $f'(x)$ over $X^{(j)}$, and $m(X^{(j)})$ is the mid-point of $X^{(j)}$. The interval Newton method will never diverge provided that $0 \notin F'(X^{(0)})$, or equivalently that a unique zero of f lies in $X^{(0)}$. The interval Newton method was derived by [Moore, 1967]. If there is only one root x^* of a continuously differentiable f in a compact $X^{(0)}$, then the sequence of compact sets $X^{(0)} \supseteq X^{(1)} \supseteq X^{(2)} \ldots$ can be shown to converge quadratically to x^* [Alefeld and Herzberger, 1983]. We can derive the above dynamical system in $\mathbb{IR}$ via the mean value theorem. Let $f(x)$ be continuously differentiable and $f'(x) \neq 0$ for all $x \in X$ such that x^* is the only zero of f in X. Then, by the mean value theorem, there exists $c \in (x, x^*)$ such that $f(x) - f(x^*) = f'(c)(x - x^*)$ for every x. Since $f'(c) \neq 0$ by assumption, and since $f(x^*) = 0$, it follows that:

$$x^* = x - \frac{f(x)}{f'(c)} \in x - \frac{f(x)}{F'(X)} =: N(X), \quad \forall x \in X.$$

$N(X)$ is called the Newton operator and it contains x^*. Since our root of interest lies in X, $x^* \in N(X) \cap X$. Note that the above dynamical system in $\mathbb{IR}$ is obtained by replacing x with $m(X)$ and X with $X^{(j)}$ in the previous expression. The usual Newton method lends itself to an intuitive geometric interpretation: in the jth iteration, think of shining a beam of light onto the domain from the point $(x^{(j)}, f(x^{(j)}))$ along the tangent to $f(x)$ at $x^{(j)}$. The intersection of this beam (white line in Figure 20.3) with the domain provides $x^{(j+1)}$, which is where the next iteration is resumed. In the interval Newton method, then, we shine a set of beams from the point $(x^{(j)}, f(x^{(j)}))$ along the directions of all the tangents to $f(x)$ on the entire interval X. The intersection of these beams (gray floodlight of Figure 20.3) with the domain is $N(X^{(j)})$. The iteration is resumed with the new interval $X^{(j+1)} = N(X^{(j)}) \cap X^{(j)}$. Next we extend the interval Newton method in order to allow $F'(X)$ to contain 0.

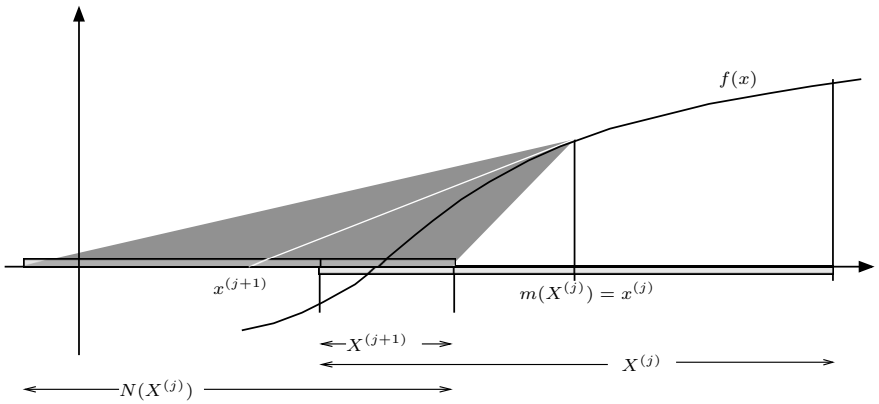

Fig. 20.3. *Geometric interpretation of the interval Newton method*

By including the points $+\infty$ and $-\infty$ to $\mathbb{R}$, it becomes possible to define extended interval arithmetic (EIA) on $\mathbb{IR}^* := \mathbb{IR} \cup \{(-\infty, \overline{x}] : \overline{x} \in \mathbb{R}\} \cup \{[\underline{x}, +\infty) : \underline{x} \in \mathbb{R}\} \cup (-\infty, +\infty)$, the set of intervals with end points in the complete lattice $\mathbb{R}^* := \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$, with respect to the ordering relation $\leq$. Let $[\,]$ denote the empty interval. Division by intervals containing 0 becomes possible with the following rules:

$$X/Y := \begin{cases} (-\infty, +\infty) & \text{if } 0 \in X, \text{or } Y = [0,0] \\ [\,] & \text{if } 0 \notin X, \text{and } Y = [0,0] \\ [\overline{x}/\underline{y}, +\infty) & \text{if } \overline{x} \leq 0, \text{ and } \overline{y} = 0 \\ [\underline{x}/\overline{y}, +\infty) & \text{if } 0 \leq \underline{x}, \text{ and } 0 = \underline{y} < \overline{y} \\ (-\infty, \overline{x}/\overline{y}] & \text{if } \overline{x} \leq 0, \text{ and } 0 = \underline{y} < \overline{y} \\ (-\infty, \underline{x}/\underline{y}] & \text{if } 0 \leq \underline{x}, \text{ and } \underline{y} < \overline{y} = 0 \\ (-\infty, \overline{x}/\overline{y}] \cup [\overline{x}/\underline{y}, +\infty) & \text{if } \overline{x} \leq 0, \text{ and } [0,0] \Subset Y \\ (-\infty, \underline{x}/\underline{y}] \cup [\underline{x}/\overline{y}, +\infty) & \text{if } 0 \leq \underline{x}, \text{ and } [0,0] \Subset Y. \end{cases}$$

When X is a thin interval with $x = \underline{x} = \overline{x}$ and Y has $+\infty$ or $-\infty$ as one of its bounds, then extended interval subtraction is also necessary for the extended interval Newton algorithm, and is defined as follows:

$$[\underline{x}, \overline{x}] - Y := \begin{cases} (-\infty, +\infty) & \text{if } Y = (-\infty, +\infty) \\ (-\infty, x - \underline{y}] & \text{if } Y = (\underline{y}, +\infty) \\ [x - \overline{y}, +\infty) & \text{if } Y = (-\infty, \overline{y}]. \end{cases}$$

The extended interval Newton method uses the EIA described above and is a variant of the method based on [Hansen and Sengupta, 1981] with Ratz's modifications [Ratz, 1992] as implemented in [Hammer *et al.*, 1995]. It can be used to enclose the roots of a continuously differentiable $f : \mathbb{R}^n \to \mathbb{R}^n$ in a given box $X \in \mathbb{IR}^n$. Let $J_f(x) := ((\partial f_i(x)/\partial x_j))_{i,j=\{1,\ldots,n\}} \in \mathbb{R}^{n \times n}$ denote the Jacobian matrix of f at x. Let $J_F(X) \supset J_f(X)$ denote the Jacobian

of the interval extension of f. The Jacobian can be computed via AD by computing the gradient of each component f_i of f. By the mean value theorem, $f(m(X))-f(x^*) = J_f(w)\cdot(m(X)-x^*)$, for some $x^* \in X, w = (w_1, w_2, \ldots, w_n)$, where $w_i \in X, \forall i \in \{1, 2, \ldots, n\}$. Setting $f(x^*) = 0$ yields the relation $x^* \in \mathcal{N}(X) \cap X$, where $\mathcal{N}(X) := m(X) - (\, J_F(X)\,)^{-1} \cdot F(m(X))$, for all $x \in X$ such that $J_F(x)$ is invertible. An iteration scheme $X^{(j+1)} := \mathcal{N}(X^{(j)}) \cap X^{(j)}$ for $j = 0, 1, \ldots$, and $X^{(0)} := X$ will enclose the zeros of f contained in X. We may relax the requirement that every matrix in $J_F(X)$ be invertible by using the inverse of the midpoint of $J_F(X)$, i.e., $(m(J_F(X)))^{-1} =: p \in \mathbb{R}^{n\times n}$, as a matrix preconditioner. The extended interval Gauss-Seidel iteration, which is also applicable to singular systems [Neumaier, 1990], is used to solve the preconditioned interval linear equation

$$\begin{aligned} p \cdot F(m(X)) &= p \cdot J_F(X) \cdot (m(X) - x^*) \\ a &= G \cdot (c - x^*), \end{aligned}$$

where $a \in A := p \cdot F(m(X)), G := p \cdot J_F(X)$, and $c := m(X)$. Thus the solution set $\mathbf{S} := \{x \in X : g \cdot (c - x) = a, \forall g \in G\}$ of the interval linear equation $a = G \cdot (c - x)$ has the componentwise solution set $\mathbf{S}_i = \{x_i \in X_i : \sum_{j=1}^{n} (\, g_{i,j} \cdot (c_j - x_j)\,) = a_i, \forall g \in G\}$, $\forall i \in \{1, \ldots, n\}$. Now set $Y = X$, and solve the ith equation for the ith variable iteratively for each i as follows:

$$\begin{aligned} y_i = c_i - \tfrac{1}{g_{i,i}} \Big(a_i + \textstyle\sum_{j=1, j\neq i}^{n} (\, g_{i,j} \cdot (y_j - c_j)\,)\Big) \\ \in \Big(c_i - \tfrac{1}{G_{i,i}} \Big(A_i + \textstyle\sum_{j=1, j\neq i}^{n} (\, G_{i,j} \cdot (Y_j - c_j)\,)\Big)\Big) \cap Y_i. \end{aligned}$$

Then $\mathcal{N}_{GS}(X)$, the set resulting from one extended interval Newton Gauss-Seidel step such that $\mathbf{S} \subseteq \mathcal{N}_{GS}(X) \subseteq X$, contains interval vector(s) Y obtained by this iteration. Thus the roots of f are enclosed by the discrete dynamical system $X^{(j)} = \mathcal{N}_{GS}(X^{(j)})$ in $\mathbb{IR}^n$. Every 0 of f that lies in X also lies in $\mathcal{N}_{GS}(X)$. If $\mathcal{N}_{GS}(X) = [\,]$, the empty interval, then f has no solution in X. If $\mathcal{N}_{GS}(X) \Subset X$, then f has a unique solution in X [Hansen, 1992]. When $G_{ii} \supset 0$, the method is applicable with EIA that allows for division by 0. In such cases, we may obtain up to two disjoint compact intervals for Y_i subsequent to EIA and intersection with the previous compact interval X_i. In such cases, the iteration is applied to each resulting sub-interval.

All the interval arithmetic demonstrated up to this point involved real intervals. However, $\mathcal{R}$, the set of floating-point numbers available on a computing machine, is finite. A *machine interval* is a real interval with bounds in $\mathcal{R}$, the set of floating-point numbers described in the introduction. We can perform IA on $\mathbb{I}\mathcal{R} = \{\{X \in \mathbb{IR} : \underline{x}, \overline{x} \in \mathcal{R}\}$, the set of all machine intervals, in a computer. In spite of the finiteness of $\mathbb{I}\mathcal{R}$, the strength of IA lies in a machine interval X being able to enclose a segment of the entire continuum of reals between its machine-representable boundaries. Operations with real intervals can be tightly enclosed by the *rounding directed* operations, provided by the IEEE arithmetic standard, with the smallest machine intervals containing them [Hammer *et al.*, 1995, Kulisch *et al.*, 2001].

20.2 Enclosing the likelihood of a compact set of trees

Let $\mathcal{D}$ denote a homologous set of distinct DNA sequences of length v from n taxa. We are interested in the branch lengths of the most likely tree under a particular topology. Let b denote the number of branches and s denote the number of nodes of a tree with topology τ. Thus, for a given unrooted topology τ with n leaves and b branches, the unknown parameter $\theta = (\theta_1, \ldots, \theta_b)$ is the real vector of branch lengths in the positive orthant ($\theta_q \in \mathbb{R}_+$). An explicit model of DNA evolution is needed to construct the likelihood function which gives the probability of observing data $\mathcal{D}$ as a function of the parameter θ. The simplest such continuous time Markov chain model (JC69) on the state space Σ is due to Jukes and Cantor [Jukes and Cantor, 1969]. We may compute $\ell^{(k)}(\theta)$, the log-likelihood at site $k \in \{1, \ldots, v\}$ through, the following post-order traversal [Felsenstein, 1981]:

(i) Associate with each node $q \in \{1, \ldots, s\}$ with m descendants, a partial likelihood vector, $\mathbf{l}_q := (\mathbf{l}_q^A, \mathbf{l}_q^C, \mathbf{l}_q^G, \mathbf{l}_q^T) \in \mathbb{R}^4$, and let the length of the branch leading to its ancestor be θ_q.

(ii) For a leaf node q with nucleotide i, set $\mathbf{l}_q^i = 1$ and $\mathbf{l}_q^j = 0$ for all $j \neq i$. For any internal node q, set $\mathbf{l}_q := (1, 1, 1, 1)$.

(iii) For an internal node q with descendants $s_1, s_2, \ldots, s_m$,

$$\mathbf{l}_q^i = \sum_{j_1, \ldots, j_m \in \Sigma} \{ \mathbf{l}_{s_1}^{j_1} \cdot P_{i,j_1}(\theta_{s_1}) \cdot \mathbf{l}_{s_2}^{j_2} \cdot P_{i,j_2}(\theta_{s_2}) \ldots \mathbf{l}_{s_m}^{j_m} \cdot P_{i,j_m}(\theta_{s_m}) \}.$$

(iv) Compute $\mathbf{l}_q$ for each sub-terminal node q, then those of their ancestors recursively to finally compute $\mathbf{l}_r$ for the root node r to obtain the log-likelihood for site k, $\ell^{(k)}(\theta) = \mathbf{l}_r = \log \sum_{i \in \Sigma} (\pi_i \cdot \mathbf{l}_r^i)$.

Assuming independence across sites we obtain $\ell(\theta) = \sum_{k=1}^{v} \ell^{(k)}(\theta)$, the natural logarithm of the likelihood function for the data $\mathcal{D}$, by multiplying the site-specific likelihoods. The problem of finding the global maximum of this likelihood function is equivalent to finding the global minimum of $l(\theta) := -\ell(\theta)$. Replacing every constant c by its corresponding constant triple $(C, 0, 0)$, every variable θ_j by its triple $(\Theta_j, e^{(j)}, 0)$, and every real operation or elementary function by its counterpart in interval-extended Hessian differentiation arithmetic in the above post-order traversal yields a rigorous enclosure of the negative log-likelihood triple $(\mathcal{L}(\Theta), \nabla\mathcal{L}(\Theta), \nabla^2\mathcal{L}(\Theta))$ of the negative log-likelihood function $l(\theta)$ over Θ.

20.3 Global Optimization

20.3.1 Branch-and-bound

The most basic strategy in global optimization through enclosure methods is to employ rigorous branch-and-bound techniques. Such techniques recursively partition (branch) the original compact space of interest into compact subspaces and discard (bound) those subspaces that are guaranteed to not contain the global optimizer(s). For the real scalar-valued multi-dimensional objective

function $l(\theta)$, the interval branch-and-bound technique can be applied to its natural interval extension $\mathcal{L}(\Theta)$ to obtain an interval enclosure $\mathcal{L}^*$ of the global minimum value l^* as well as the set of minimizer(s) to a specified accuracy ϵ. Note that this set of minimizer(s) of $\mathcal{L}(\theta)$ is the set of maximizer(s) of the likelihood function for the observed data $\mathcal{D}$. The strength of such methods arises from the algorithmic ability to discard large sub-boxes from the original search region,

$$\Theta^{(0)} = (\Theta_1^{(0)}, \ldots, \Theta_b^{(0)}) := ([\underline{\theta}_1^{(0)}, \overline{\theta}_1^{(0)}], \ldots, [\underline{\theta}_b^{(0)}, \overline{\theta}_b^{(0)}]) \subset \mathbb{IR}^b,$$

that are not candidates for global minimizer(s). Four tests that help discard sub-regions are described below. Let $\mathfrak{L}$ denote a list of ordered pairs of the form $(\Theta^{(i)}, \underline{\mathcal{L}}_{\Theta^{(i)}})$, where $\Theta^{(i)} \subseteq \Theta^{(0)}$, and $\underline{\mathcal{L}}_{\Theta^{(i)}} := \min (\mathcal{L}(\Theta^{(i)}))$ is a lower bound for the image of the negative log-likelihood function l over $\Theta^{(i)}$. Let $\tilde{l}$ be an upper bound for l^* and $\nabla\mathcal{L}(\Theta^{(i)})_k$ denote the k-th interval of the gradient box $\nabla\mathcal{L}(\Theta^{(i)})$. If no information is available for $\tilde{l}$, then $\tilde{l} = \infty$.

20.3.1.1 Midpoint cutoff test

The basic idea of the *midpoint cutoff test* is to discard sub-boxes of the search space $\Theta^{(0)}$ with the lower bound for their image enclosures above $\tilde{l}$, the current best estimate of an upper bound for l^*. Figure 20.4 shows a multi-modal l as a function of a scalar θ over $\Theta^{(0)} = \cup_{i=1}^{16} \Theta^{(i)}$. For this illustrative example, $\tilde{l}$ is set as the upper bound of the image enclosure of l over the smallest machine interval containing the midpoint of $\Theta^{(15)}$, the interval with the smallest lower bound of its image enclosure. The shaded rectangles show the image enclosures over intervals that lie strictly above $\tilde{l}$. In this example the *midpoint cutoff test* would discard all other intervals except $\Theta^{(1)}$, $\Theta^{(2)}$, and $\Theta^{(4)}$. Given a list $\mathfrak{L}$ and candidate upper bound $\tilde{l}$, the midpoint cutoff test works as follows:

- Given a list $\mathfrak{L}$ and $\tilde{l}$.
- Choose an element j of $\mathfrak{L}$, such that $j = \text{argmin}\, \underline{\mathcal{L}}_{\Theta^{(i)}}$, since $\Theta^{(j)}$ is likely to contain a minimizer.
- Find its midpoint $c = m(\Theta^{(j)})$ and let C be the smallest machine interval containing c.
- Compute a possibly improved $\tilde{l} = \min \{ \tilde{l}, \overline{\mathcal{L}}_C \}$, where $\overline{\mathcal{L}}_C := \max(\mathcal{L}(C))$.
- Discard any i-th element of $\mathfrak{L}$ for which $\underline{\mathcal{L}}_{\Theta^{(i)}} > \tilde{l} \geq l^*$.

20.3.1.2 Monotonicity test

For a continuously differentiable function $l(\theta)$, the *monotonicity test* determines whether $l(\theta)$ is strictly monotone over an entire sub-box $\Theta^{(i)} \subset \Theta^{(0)}$. If l is strictly monotone over $\Theta^{(i)}$, then a global minimizer cannot lie in the interior of $\Theta^{(i)}$. Therefore, $\Theta^{(i)}$ can only contain a global minimizer as a boundary point if this point also lies in the boundary of $\Theta^{(0)}$. Figure 20.5 illustrates the *monotonicity test* for the one-dimensional case. In this example the search space of interest, $\Theta^{(0)} = [\underline{\theta}^{(0)}, \overline{\theta}^{(0)}] = \cup_{i=1}^{8} \Theta^{(i)}$, can be reduced considerably. In the interior of $\Theta^{(0)}$, we may delete $\Theta^{(2)}$, $\Theta^{(5)}$, and $\Theta^{(7)}$, since $l(\theta)$ is monotone

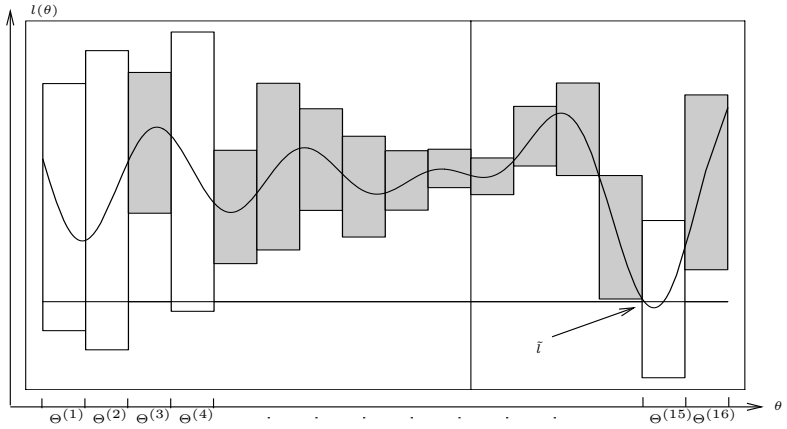

Fig. 20.4. *Midpoint cutoff test*

over them as indicated by the enclosure of the derivative $l'(\theta)$ being bounded away from 0. Since $l(\theta)$ is monotonically decreasing over $\Theta^{(1)}$ we may also deleted it, since we are only interested in minimization. $\Theta^{(8)}$ may be pruned to its right boundary point $\theta^{(8)} = \overline{\theta}^{(8)} = \overline{\theta}^{(0)}$ due to the strictly decreasing nature of $l(\theta)$ over it. Thus the *monotonicity test* has pruned $\Theta^{(0)}$ to the smaller candidate set $\{\overline{\theta}^{(0)}, \Theta^{(3)}, \Theta^{(4)}, \Theta^{(6)}\}$ for a global minimizer.

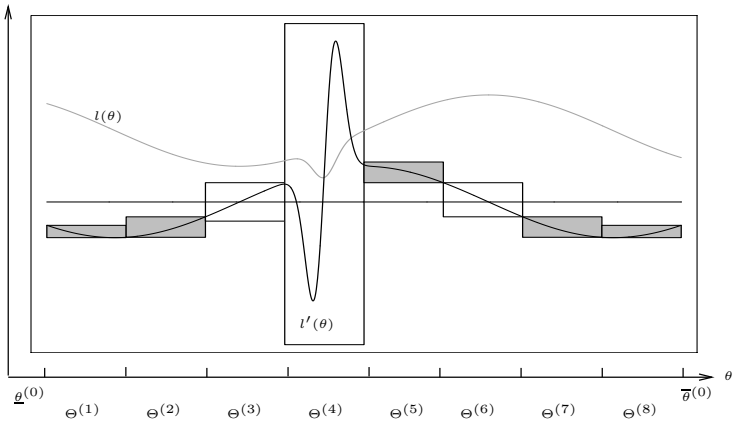

Fig. 20.5. *Monotonicity test*

- Given $\Theta^{(0)}$, $\Theta^{(i)}$, and $\nabla\mathcal{L}(\Theta^{(i)})$.
- Iterate for $k = 1, \ldots, b$
  - If $0 \in \nabla\mathcal{L}(\Theta^{(i)})_k$, then leave $\Theta_k^{(i)}$ unchanged, as it may contain a stationary point of l.
  - Otherwise, $0 \notin \nabla\mathcal{L}(\Theta^{(i)})_k$. This implies that $\Theta^{(i)}$ can be pruned, since $l^* \notin \Theta^{(i)}$ except possibly at the boundary points, as follows:
    (i) if $\min(\nabla\mathcal{L}(\Theta^{(i)})_k) > 0$ and $\underline{\theta}_k^{(0)} = \underline{\theta}_k^{(i)}$, then $\Theta_k^{(i)} = [\underline{\theta}_k^{(i)}, \underline{\theta}_k^{(i)}]$,

(ii) Else if $\max(\nabla\mathcal{L}(\Theta^{(i)})_k) < 0$ and $\overline{\theta}_k^{(0)} = \overline{\theta}_k^{(i)}$, then $\Theta_k^{(i)} = [\overline{\theta}_k^{(i)}, \overline{\theta}_k^{(i)}]$.
(iii) Else, delete the i-th element of $\mathfrak{L}$ and stop the iteration.

20.3.1.3 Concavity test

Given $\Theta^{(i)} \Subset \Theta^{(0)}$, and the diagonal elements $(\nabla^2\mathcal{L}(\Theta^{(i)}))_{kk}$ of $\nabla^2\mathcal{L}(\Theta^{(i)})$, note that if $\min((\nabla^2\mathcal{L}(\Theta^{(i)}))_{kk}) < 0$ for some k, then $\nabla^2\mathcal{L}(\Theta^{(i)})$ cannot be positive semidefinite, and therefore $l(\theta)$ cannot be convex over $\Theta^{(i)}$ and thus cannot contain a minimum in its interior. In the one-dimensional example shown in Figure 20.5, an application of the *concavity test* to the candidate set $\{\underline{\theta}^{(0)}, \Theta^{(4)}, \Theta^{(6)}\}$ for a global minimizer returned by the *monotonicity test* would result in the deletion of $\Theta^{(6)}$ due to the concavity of $l(\theta)$ over it.

- Given $\Theta^{(i)} \Subset \Theta^{(0)}$ and $\nabla^2\mathcal{L}(\Theta^{(i)})$
- If $\min((\nabla^2\mathcal{L}(\Theta^{(i)}))_{kk}) < 0$ for any $k \in \{1, \ldots, b\}$, then delete the i-th element of $\mathfrak{L}$.

20.3.1.4 Interval Newton test

Given $\Theta^{(i)} \Subset \Theta^{(0)}$, and $\nabla\mathcal{L}(\Theta^{(i)})$, we attempt to solve the system, $\nabla\mathcal{L}(\theta) = 0$ in terms of $\theta \in \Theta^{(i)}$.

- Apply one extended interval Newton Gauss-Seidel step to the linear interval equation $a = G \cdot (c - \theta)$, where $a := p \cdot \mathcal{L}(m(\Theta^{(i)}))$, $G := p \cdot \nabla^2\mathcal{L}(\Theta^{(i)})$, $c := m(\Theta^{(i)})$, and $p := (m(\nabla^2 F(X)))^{-1}$, in order to obtain $\mathcal{N}'_{GS}(\Theta^{(i)})$.
- One of the following can happen,
  (i) If $\mathcal{N}'_{GS}(\Theta^{(i)})$ is empty, then discard $\Theta^{(i)}$.
  (ii) If $\mathcal{N}'_{GS}(\Theta^{(i)}) \Subset \Theta^{(i)}$, then replace $\Theta^{(i)}$ by the contraction $\mathcal{N}'_{GS}(\Theta^{(i)}) \cap \Theta^{(i)}$.
  (iii) If $0 \in G_{jj}$, and the extended interval division splits $\Theta_j^{(i)}$ into a non-empty union of $\Theta_j^{(i),1}$ and $\Theta_j^{(i),2}$, then the iteration is continued on $\Theta_j^{(i),1}$, while $\Theta_j^{(i),2}$, if non-empty, is stored in $\mathfrak{L}$ for future processing. Thus, one extended interval Newton Gauss-Seidel step can add at most $b + 1$ sub-boxes to $\mathfrak{L}$.

20.3.2 Verification

Given a collection of sub-boxes $\{\Theta^{(1)}, \ldots, \Theta^{(n)}\}$, each of width $\leq \epsilon$, that could not be discarded by the tests in Section 20.3.1, one can attempt to verify the existence and uniqueness of a local minimizer within each sub-box $\theta^{(i)}$ by checking whether the conditions of the following two theorems are satisfied. For proof of these two theorems see [Hansen, 1992] and [Ratz, 1992].

(i) If $\mathcal{N}'_{GS}(\Theta^{(i)}) \Subset \Theta^{(i)}$, then there exists a unique stationary point of $\mathcal{L}$, i.e., a unique zero of $\nabla\mathcal{L}$ exists in $\Theta^{(i)}$.
(ii) If $(I + \frac{1}{\kappa} \cdot (\nabla^2\mathcal{L}(\Theta^{(i)}))) \cdot Z \Subset Z$, where $(\nabla^2\mathcal{L}(\Theta^{(i)}))_{d,\infty} \leq \kappa \in \mathbb{R}$ for some $Z \in \mathbb{IR}^n$ then the spectral radius $\rho(s) < 1$ for all $s \in (I - \frac{1}{\kappa} \cdot (\nabla^2\mathcal{L}(\Theta^{(i)})))$ and all symmetric matrices in $\nabla^2\mathcal{L}(\Theta^{(i)})$ are positive definite.

If the conditions of the above two theorems are satisfied by some $\Theta^{(i)}$, then a unique stationary point exists in $\Theta^{(i)}$ and this stationary point is a local minimizer. Therefore, if exactly one candidate sub-box for minimizer(s) remains after pruning the search box $\Theta^{(0)}$ with the tests in Section 20.3.1, and if this sub-box satisfies the above two conditions for the existence of a unique local minimizer within it, then we have rigorously enclosed the global minimizer in the search interval. On the other hand, if there are two or more sub-boxes in our candidate list for minimizer(s) that satisfy the above two conditions, then we may conclude that each sub-box contains a candidate for a global minimizer which may not necessarily be unique (as in the case of disconnected sub-boxes each of which contains a candidate). Observe that failure to verify the uniqueness of a local minimizer in a sub-box can occur if it contains more than one point, or even a continuum of points, that are stationary.

20.3.3 Algorithm

- *Initialization:*

  Step 1 Let the search region be a single box $\Theta^{(0)}$ or a collection of not necessarily connected, but pairwise disjoint boxes, $\Theta^{(i)}$, $i \in \{1, \ldots, r\}$.

  Step 2 Initialize the list $\mathfrak{L}$ which may just contain one element $(\Theta^{(0)}, \underline{\mathcal{L}}_{\Theta^{(0)}})$ or several elements

  $$\{ (\Theta^{(1)}, \underline{\mathcal{L}}_{\Theta^{(1)}}), (\Theta^{(2)}, \underline{\mathcal{L}}_{\Theta^{(2)}}), \ldots, (\Theta^{(r)}, \underline{\mathcal{L}}_{\Theta^{(r)}}) \}.$$

  Step 3 Let ϵ be a specified tolerance.

  Step 4 Let $\max_{\mathfrak{L}}$ be the maximal length allowed for list $\mathfrak{L}$.

  Step 5 Set the noninformative lower bound for l^*, i.e., $\tilde{l} = \infty$

- *Iteration:*

  Step1 Perform the following operations:

  Step 1.1 Improve $\tilde{l} = \min\{\tilde{l}, \max(\mathcal{L}(m(\Theta^{(j)})))\}$, $j = \operatorname{argmin}\{\underline{\mathcal{L}}_{\Theta^{(i)}}\}$.

  Step 1.2 Perform the *midpoint cutoff test* on $\mathfrak{L}$.

  Step 1.3 Set $\mathcal{L}^* = [\underline{\mathcal{L}}_{\Theta^{(j)}}, \tilde{l}]$.

  Step 2 Bisect $\Theta^{(j)}$ along its longest side k, i.e., $d(\Theta_k^{(j)}) = d_\infty(\Theta^{(j)})$, to obtain sub-boxes $\Theta^{(j_q)}$, $q \in \{1, 2\}$.

  Step 3 For each sub-box $\Theta^{(j_q)}$, evaluate $(\mathcal{L}(\Theta^{(j_q)}), \nabla\mathcal{L}(\Theta^{(j_q)}), \nabla^2\mathcal{L}(\Theta^{(j_q)}))$, and do the following:

  Step 3.1 Perform *monotonicity test* to possibly discard $\Theta^{(j_q)}$.

  Step 3.2 *Centered form cutoff test:*

  Improve the image enclosure of $\mathcal{L}(\Theta^{(j_q)})$ by replacing it with its centered form $\mathcal{L}_c(\Theta^{(j_q)}) :=$

  $$\{\mathcal{L}(m(\Theta^{(j_q)})) + \nabla\mathcal{L}(\Theta^{(j_q)}) \cdot (\Theta^{(j_q)} - m(\Theta^{(j_q)}))\} \cap \mathcal{L}\,(\Theta^{(j_q)}),$$

  and then discarding $\Theta^{(j_q)}$, if $\tilde{l} < \underline{\mathcal{L}}_{\Theta^{(jq)}}$.

  Step 3.3 Perform *concavity test* to possibly discard $\Theta^{(j_q)}$.

Table 20.1. *Machine interval MLEs of a log-likelihood function for a phylogenetic tree on three taxa:*
Chimpanzee (1), Gorilla (2), and Orangutan (3).

$\Theta^{(0)}$ and Tree	$\Theta^* \supset \theta^*$	$-\mathcal{L}(\Theta^*) \supset -l(\theta^*)$
$[1.0 \times 10^{-11}, 10.0]^{\otimes 3}$	$5.98162213842_0^2 \times 10^{-2}$	
$\tau_1 = (1,2,3)$	$5.41674167942_0^2 \times 10^{-2}$	
	$1.32990896859_8^9 \times 10^{-1}$	$-2.150318065856_6^5 \times 10^3$

Step 3.4 Apply an *extended interval Newton Gauss-Seidel step* to $\Theta^{(j_q)}$, in order to either entirely discard it or shrink it into v sub-sub-boxes, where v is at most $2s - 2$.

Step 3.5 For each one of these sub-sub-boxes $\Theta^{(j_{q,u})}$, $u \in \{1, \ldots, v\}$

Step 3.5.1 Perform *monotonicity test* to possibly discard $\Theta^{(j_{q,u})}$.

Step 3.5.2 Try to discard $\Theta^{(j_{q,u})}$ by applying the *centered form cutoff test* in **Step 3.2** to it.

Step 3.5.3 Append $(\Theta^{(j_{q,u})}, \underline{\mathcal{L}}_{\Theta^{(j_{q,u})}})$ to $\mathfrak{L}$ if $\Theta^{(j_{q,u})}$ could not be discarded by **Step 3.5.1** and **Step 3.5.2**.

- *Termination:*

Step 1 Terminate iteration if $d_{rel,\infty}(\Theta^{(j)}) < \epsilon$, or $d_{rel,\infty}(\mathcal{L}^*) < \epsilon$, or $\mathfrak{L}$ is empty, or $\text{Length}(\mathfrak{L}) > \max_{\mathfrak{L}}$.

Step 2 Verify uniqueness of minimizer(s) in the final list $\mathfrak{L}$ by applying algorithm given in Section 20.3.2 to each of its elements.

20.4 Applications to phylogenetics

By way of example, we apply our enclosure method to identifying the global maximum of the log-likelihood function for the JC69 model of DNA evolution on the three-taxa unrooted tree. The homologous sequences used were taken from the mitochondrial DNA of the chimpanzee (*Pan troglodytes*), gorilla (*Gorilla gorilla*), and orangutan (*Pongo pygmaeus*) [Brown *et al.*, 1982]. There is only one unrooted multifurcating topology for three species with all three branches emanating from the root like a star. The data set for this problem is summarized in [Sainudiin, 2004] by 29 data patterns. The sufficient statistic for this data is $(7, 100, 42, 46, 700)$. Details on obtaining this sufficient statistics can be found in Chapter 18. The parameter space is three-dimensional, corresponding to the three branch lengths of the 3-leaved star tree τ_1. The algorithm is given a large search box $\Theta^{(0)}$. The results are summarized in Table 20.1. The notation x_a^b means the interval $[xa, xb]$ (e.g., $5.98162213842_0^2 \times 10^{-2} = [5.98162213840 \times 10^{-2}, 5.98162213842 \times 10^{-2}]$). Figure 20.6 shows the the parameter space being rigorously pruned as the algorithm

Table 20.2. *Computational efficiency for four different 3 taxa trees.*

True Tree	Calls to $\mathcal{L}(\Theta^*)$	CPU time
$(1:0.01, 2:0.07, 3:0.07)$	1272 [1032, 1663]	0.55 [0.45, 0.72]
$(1:0.02, 2:0.19, 3:0.19)$	3948 [2667, 6886]	1.75 [1.17, 3.05]
$(1:0.03, 2:0.42, 3:0.42)$	20789 [12749, 35220]	9.68 [5.94, 16.34]
$(1:0.06, 2:0.84, 3:0.84)$	245464 [111901, 376450]	144.62 [64.07, 232.94]

progresses. When there are four taxa, the phylogeny estimation problem is more challenging as there are four distinct topologies to consider in addition to the branch lengths. A similar method was used to solve the most likely phylogeny of four primates with data from their mitochondria [Sainudiin, 2004].

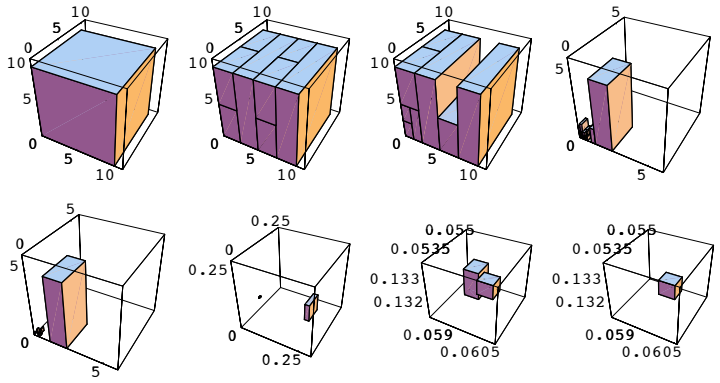

Fig. 20.6. Progress of the algorithm as it prunes $[0.001, 10.0]^{\otimes 3}$.

The running time of the global optimization algorithm depends on where the MLEs lie in the parameter space. For trees with smaller branch lengths, the running time is faster, while larger trees have a much longer running time. The Table 20.2 shows the mean and 95% confidence intervals of the number of calls to the likelihood function $\mathcal{L}$ and the CPU time in seconds for each of four trees with different weights. The results summarized in Table 20.2 are from 100 data sets, each of sequence length 1000, simulated under the JC69 model upon each one of the four trees shown in the first column.

The enclosure of an MLE by means of interval methods is equivalent to a proof of maximality. The method is robust in the presence of multiple local maxima or nonidentifiable manifolds with the same ML value. For example, when a time-reversible Markov chain, such as JC69, is superimposed on a rooted tree, only the sum of the branch lengths emanating from the root is identifiable. Identifiability is a prerequisite for statistical consistency of estimators. To demonstrate the ability of interval methods to enclose the nonidentifiable ridge along $\theta_1 + \theta_2$ in the simplest case of a two-leaved tree, we formulated a

nonidentifiable negative log-likelihood function $l(\theta_1, \theta_2)$ with its global minimizers along $\theta_1 + \theta_2 = \frac{3}{4}\log(45/17) = 0.730087$ under a fictitious dataset for which 280 out of 600 sites are polymorphic. Figure 20.7 shows the contours of

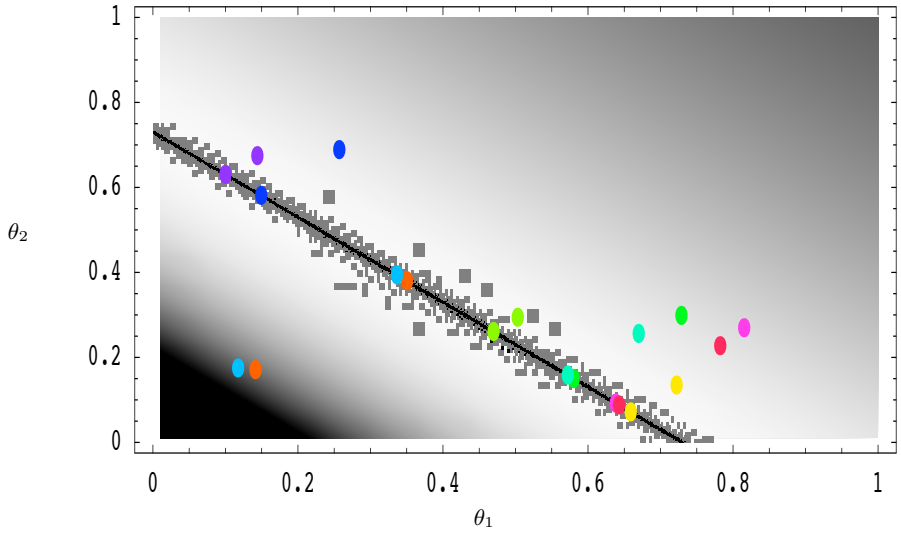

Fig. 20.7. The nonidentifiable subspace of minimizers $\theta_1 + \theta_2 = \frac{3}{4}\log(45/17)$ of $l(\theta_1, \theta_2)$ under the JC69 model evolving on a rooted two-leaved tree is enclosed by a union of up to 30,000 rectangles. The larger gray, and smaller black rectangles have tolerances of $\epsilon = 1.0 \times 10^{-4}$ and $\epsilon = 1.0 \times 10^{-6}$, respectively. The 10 pairs of colored ovals are the initial and final points of 10 local quasi-Newton searches with random initializations.

$l(\theta_1, \theta_2)$ in gray scale and the solutions of the interval method (gray and black rectangles) and those of 10 quasi-Newton searches with random initializations (10 pairs of colored ovals). Observe that the basin of attraction for each point on $\theta_1 + \theta_2 = 0.730087$ under a quasi-Newton local search algorithm is the line running orthogonal to it.

Interval methods can be slow on currently available processors that are optimized for floating-point arithmetic, especially when applied naively. Efficiency can be gained by pre-enclosing the likelihood function over a fine mesh and accessing them via hash tables. Interval methods can work efficiently when algebraic techniques are first used to reduce the data into sufficient statistics. Interval methods are particularly suited for solving a large dimensional problem by amalgamating the solutions of several lower dimensional problems. For instance, we can apply the rigorously enclosed MLEs to the generalized neighbor-joining (GNJ) method discussed in Chapter 2. We call this the *numerically rigorous generalized neighbor-joining method (NRGNJ)*. Using `fastDNAml` which implements a gradient flow algorithm with floating-point arithmetic, [Levy *et al.*, 2004] computed dissimilarity maps that are needed for the GNJ method. The NRGNJ method uses, instead, the rigorously enclosed MLEs. We applied this method to find the NJ tree for 21 *S-locus receptor kinase* (SRK) sequences [Sainudiin *et al.*, 2005] involved in the self/nonself discrimi-

nating self-incompatibility system of the mustard family [Nasrallah, 2002]. We

Δ	NRGNJ	fastDNAml	DNAml(A)	DNAml(B)	TrExML
0	0	0	2	3608	0
2	0	0	1	471	0
4	171	6	3619	5614	0
6	5687	5	463	294	5
8	4134	3987	5636	13	71
10	8	5720	269	0	3634
12	0	272	10	0	652
14	0	10	0	0	5631
16	0	0	0	0	7

Table 20.3. *Symmetric difference (Δ) between* $10,000$ *trees sampled from the likelihood function via MCMC and the trees reconstructed by 5 methods.*

sampled $10,000$ trees from a Markov chain with stationary distribution proportional to the likelihood function by means of a Markov chain Monte Carlo (MCMC) algorithm implemented in `PHYBAYES` [Aris-Brosou, 2003]. We then compared the tree topology of each tree generated by this MCMC method with that of the reconstructed trees via the NRGNJ method, `fastDNAml`, `DNAml` from `PHYLIP` package [Felsenstein, 2004], and `TrExML` [Wolf *et al.*, 2000] under their respective default settings with the JC69 model. We used `treedist` [Felsenstein, 2004] to compare two tree topologies. If the symmetric difference Δ between two topologies is 0, then the two topologies are identical. Larger Δ's are reflective of a larger distance between the two compared topologies. Table 20.3 summarizes the distance between a reconstructed tree and the MCMC samples from the normalized likelihood function. For example, the first two elements in the third row of Table 20.3 mean that 171 out of the $10,000$ MCMC sampled trees are at a symmetric difference of 4 ($\Delta = 4$) from the tree reconstructed via the NRGNJ method. DNAml was used in two ways: DNAml(A) is a basic search with no global rearrangements, whereas DNAml(B) applies a broader search with global rearrangements and 100 jumbled inputs. The fruits of the broader search are reflected by the accumulation of MCMC sampled trees over small Δ values from the DNAml(B) tree. Although the NRGNJ tree is identical to the Saito and Nei NJ tree (with pairwise distance) [Saitou and Nei, 1987] as well as the fastDNAml-based NJ tree with 3 leaves for this dataset, we now have the guarantee from the NRGNJ method that the MLEs for each triplet was enclosed.

In conclusion, we have seen a general method to rigorously enclose the likelihood function over compact boxes of branch lengths. This approach to ML estimation is equivalent to a computer-aided proof and is efficient when coupled with algebraic techniques. We can also rigorously amalgamate small trees. Interval methods can be naturally applied to other phylogenetic problems.