Journal of Theoretical Biology 262 (2010) 413–424

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

Phylogenetic estimation with partial likelihood tensors

J.G. Sumner [a,b,*], M.A. Charleston [a,c,d]

[a] School of Information Technologies, University of Sydney, NSW 2006, Australia
[b] School of Mathematics and Physics, University of Tasmania, TAS 7001, Australia
[c] Centre for Mathematical Biology, University of Sydney, NSW 2006, Australia
[d] Sydney Bioinformatics, University of Sydney, NSW 2006, Australia

ARTICLE INFO

Article history:
Received 27 January 2009
Received in revised form
30 September 2009
Accepted 30 September 2009
Available online 12 October 2009

Keywords:
Phylogenetics
Maximum likelihood
Computational complexity
Tensors

ABSTRACT

We present an alternative method for calculating likelihoods in molecular phylogenetics. Our method is based on partial likelihood tensors, which are generalizations of partial likelihood vectors, as used in Felsenstein's approach. Exploiting a lexicographic sorting and partial likelihood tensors, it is possible to obtain significant computational savings. We show this on a range of simulated data by enumerating all numerical calculations that are required by our method and the standard approach.

1. Introduction

In his landmark paper, Felsenstein (1981) popularized the method of maximum likelihood for phylogenetic estimation. This method takes as input a set of aligned molecular sequences and returns the "most likely" evolutionary tree that explains this data. This is done by taking a candidate tree, together with a Markov model that gives the probabilities for molecular state changes along each edge of the tree conditional upon some unknown parameters (such as edge lengths). An optimization routine then obtains the MLE (maximum likelihood estimator) for each of the unknown parameters by maximizing the probability of sampling the observed sequences under this model. Crucial to practical computational implementation of this approach was the introduction of a "pruning algorithm", which, under the assumptions of a reversible Markov process on a tree, allows for efficient computation of the likelihood of observed molecular sequences. Since this time, there has been an explosion in the use of the maximum likelihood method in phylogenetic studies. There have also been numerous algorithmic developments including computation with more general models (Boussau and Gouy, 2006; Yang, 1997) and heuristics speedups acting at the likelihood step (Guindon and Gascuel, 2003; Stamatakis, 2006) or the tree search level (Guindon and Gascuel, 2003; Whelan, 2007). However, at the likelihood step, the basic algorithmic implementation still proceeds by applying Felsenstein's original recursive formula.

In this article, we present an alternative to the pruning algorithm based on "partial likelihood tensors". Our approach depends upon a conceptual shift in the way that we perceive the intermediate quantities that must be calculated. This point of view relies heavily on noticing that these intermediate quantities can be brought to the fore as tensors (or multi-dimensional arrays) with the rank of the tensors fluctuating as the algorithm proceeds. This observation gives a fresh perspective on the likelihood calculation and presents new possibility and fluidity into the likelihood computation. By way of presenting this conceptual shift, we will show that these tensors can be applied in a practical context in giving a more efficient algorithm for likelihood computation (under certain caveats that we discuss). We emphasize that our method returns exactly the same likelihood values as the standard approach; we simply achieve this final result via an alternative route.

The particulars of the transition matrices do not concern us, as our results are independent of the model of sequence evolution (provided it is Markov). Hence, we will assume these matrices are given, and concentrate on computational complexity at the likelihood step. This is well justified, as, aside from the tree search, the likelihood step is the most intensive part of maximum likelihood estimation (Bryant et al., 2005). Given the task of calculating the likelihood of a single site in a sequence alignment, the method we present can actually be more cost intensive than applying Felsenstein's recursive formula. As we will show in Section 2, the effectiveness of our method becomes apparent by considering that in practice one is never calculating the likelihood

* Corresponding author at: School of Mathematics and Physics, University of Tasmania, TAS 7001, Australia. Tel.: +61 3 6226 1765.
E-mail address: jsumner@utas.edu.au (J.G. Sumner).

0022-5193/$ - see front matter Crown Copyright © 2009 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2009.09.037

of just a single site, but must calculate a likelihood value for each site in the sequence alignment. As we will show in Section 3.1, straightforward application of our method offers significant computational savings for up to 16 taxa. For data sets with >16 taxa, the structural properties of our approach are such that performance becomes significantly degraded. However, our method stands as a novel approach to this problem and has significant promise for the development of sophisticated algorithms. In particular, we will show that the high performance our method has for smaller numbers of taxa can be applied to larger data sets. Due to the flexibility of our approach, it is possible to do this in at least two different ways, and we discuss this in Section 4. We present results for simulated data only, as the performance of our method is dependant on the quality of the alignment and in real-world cases this is obviously dependent on the alignment algorithms used.

As noted by Felsenstein, the most obvious cost-saving measure follows by observing that many site patterns in an alignment occur multiple times and there is no need to recalculate the likelihood each time. The process of identifying these common sites is called "aliasing" (Felsenstein, 2004). Aliasing aside, the basic premise of our method is that a large number of sites in an alignment are often very similar to each other (two sites have most states in common and differ on only 1 or 2 of the sequences). This is certainly true for realistic data sets, as they have evolved from a common ancestor in the not-too-distant past (at least by hypothesis if not in fact). We define *partial likelihood tensors* (PLTs) as multi-dimensional arrays that generalize Felsenstein's partial likelihood vectors (PLVs). Using these PLTs, it becomes advantageous to sort the sites lexicographically and retain a few of these PLTs as the likelihood of each site is calculated. These PLTs can then be returned to when it comes to computing the likelihood of the next site, resulting in a minimization of the total number of calculations required.

An important aspect of our approach is that the lexicographic ordering can be computed exactly and efficiently using an $\mathcal{O}(N(m+k))$ radix sort (Cormen et al., 2001), where N is the number of unique site patterns in the alignment, m is the number of sequences and k is the number of possible character states. This should be compared to the related "column sorting" approach of Pond and Muse (2004), where, as the calculation moves through the alignment, each PLV from the present site is retained. Depending on how the next site differs from the present site, some of the PLVs required for the next site will be identical to those retained, and hence some superfluous computation can be avoided. This approach relies upon a (heuristic) solution of a travelling salesman problem (TSP) to find an ordering of sites that maximizes the saving. The solution of the TSP used by Pond and Muse is $\mathcal{O}(N^2)$, so we can expect that their technique works best for shorter sequences. This is the direct converse of the approach we present here, which is at its greatest power, relative to other approaches, when the sequence length is long, such that the data is very heterogeneous.

Another related approach is implemented by Stamatakis et al. (2005) using "subtree equality vectors" (SEVs). This approach extends the idea of aliasing to the subtree level: faced with the need to calculate the likelihood on a given subtree, a sweep through the corresponding sequences in the alignment is performed, counting occurrences of *homogeneous* subpatterns. (A homogeneous pattern is one in which each character state is identical.) Only the homogeneous patterns are accounted for as a general count would not amortize well with large data sets, and the SEVs must be recomputed for each alternative tree. Stamatakis et al. are primarily interested in the problem of likelihood computations for data sets with very many taxa (10^3 and above).

2. Methods

Here we present our method, `retroML`, for computing the likelihood of molecular sequence data under the assumption of a Markov model on a rooted binary tree. We do this with the aid of an example for a septet tree (Fig. 1). In Appendix A we give an example for a tree with 16 leaves and in Appendix B we give a generic presentation, valid for trees of any size. We will also discuss the computational complexity of `retroML` as judged against Felsenstein's (1981) approach $\mathcal{F}$.

We consider an alignment of m sequences, with no gaps. (It is straightforward to modify our results for when there are gaps, depending on how they are to be dealt with.) A "pattern" will be the (ordered) sequence of states that occur at a given site in a sequence alignment. The actual numeric values of the Markov model parameters will be of no concern to us: our results depend only on the number m of leaves of the tree, its topology, the number k of states, and the number N of unique patterns in the alignment. Thus, we take the transition matrices defining the Markov process on the tree as given (with one matrix for every vertex excluding the root) and consider the complexity of computing partial likelihood vectors at the root, conditioned on the patterns observed at the leaves. Rather than present empirical timing results, which are dependent on computer hardware and/or programming language, we will compare an *exact* count of numerical operations required of `retroML` to that of $\mathcal{F}$. For this purpose, we define a "cost" of a computation as a pair representing the numbers of multiplications and additions required, respectively: $s(\cdot)=[s^*(\cdot),s^+(\cdot)]$ for `retroML` and $f(\cdot)=[f^*(\cdot),f^+(\cdot)]$ for the standard approach $\mathcal{F}$.

2.1. Partial likelihood tensors

In Felsenstein's method $\mathcal{F}$, a *partial likelihood vector* (PLV) at a vertex represents the likelihood of observing each of the k possible states at that vertex, conditional upon a pattern of states at the leaves of the subtree subtended by that vertex. For the i th unique pattern, these PLVs are computed recursively by implementing the formula

$$L_i^v(a)=\left(\sum_{b=1}^{k} M_{ab}^{u_1} L_i^{u_1}(b)\right)\cdot\left(\sum_{b'=1}^{k} M_{ab}^{u_2} L_i^{u_2}(b')\right),$$

where v is an internal vertex with children u_1 and u_2, and $M_{ab}^{u_j}$ is the probability of a transition from state a to b along the edge connecting v and u_j. For a PLV at a leaf, the entry that corresponds to the state present at that leaf is set to 1, whilst the other entries are set to 0. This recursive formula plays a vital role in the efficient implementation of all likelihood calculations in molecular phylogenetics, and it is exactly this recursion that we aim to supplant with our approach.

The basic components underlying `retroML` are *partial likelihood tensors* (PLTs), which can be thought of as generalizations of partial likelihood vectors. A PLT represents the likelihood of

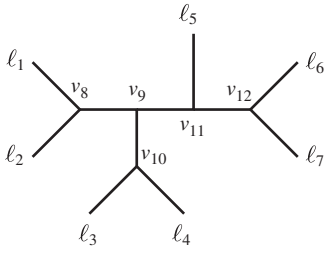

Fig. 1. Septet tree.