An algorithm for solving systems of quadratic equations in branching processes

Federico Poloni[*]

This document contains a short survey of the talk presented by the author during the XIX Conference of the Italian Mathematical Union in 2011; it summarizes and recalls the results described there.

1 Introduction — Markovian binary trees and classical algorithms

We are interested in studying Markovian binary trees [1, 4, 3, 6, 9], i.e., a family of branching processes characterized by the following laws.

1. At each time instant, a finite number of entities, called "individuals", exist.
2. Each individual can be in any one of N different states (these model, for instance, age classes or difference features in a population).
3. Each individual evolves independently from the others. Depending on its state i, it has a fixed probability b_{ijk} of being replaced by two new individuals ("children") in states j and k respectively, and a fixed probability a_i of dying without producing any offspring.

The characteristics of the population are determined therefore by the vector $a \in \mathbb{R}_+^N$ and the 3-way tensor $B \in \mathbb{R}_+^{N\times N\times N}$, where we denote by $\mathbb{R}_+$ the set of nonnegative reals. Instead of B, it is useful to think in terms of the bilinear map $b : \mathbb{R}_+^N \times \mathbb{R}_+^N \to \mathbb{R}_+^N$ given by

$$[b(u,v)]_i := \sum_{j,k=1}^{N} b_{ijk} u_j v_k.$$

We denote by e the vector, of dimensions that are clear by the context, all of whose components are 1. With this notation, a necessary compatibility condition for our model is that $e = a + b(e,e)$, that is, for each i we require the probabilities of all the possible events that can happen to an individual in state i to sum to 1.

Markovian binary trees are used not only for population dynamics, but also for instance for modelling computer and networking systems [5]. A natural

[*] Università di Pisa, Dipartimento di Informatica; Largo Pontecorvo, 3; 56128 Pisa (Italy). fpoloni@di.unipi.it

question is computing the extinction probabilities, i.e., the vector $x \in \mathbb{R}^N_+$ such that a colony starting from a single individual in state i becomes extinct in a finite time with probability x_i. One can prove [1, 3] that x is the minimal (in the componentwise order) nonnegative solution x_* of the equation

$$x = a + b(x, x).$$

As noted above, $x = e$ is a solution for this equation, although not necessarily the minimal one. In fact, one can define two different families of processes:

- *subcritical* processes, in which $x_* = e$ is the minimal solution, and thus the colony becomes extinct with probability 1. This corresponds, in a suitable sense, to the case in which the average number of children per individual is less than 1.

- *supercritical* processes, in which $x_* \leq e$ and equality does not hold for all components. In this case, there is a nonzero probability that the number of individuals grows indefinitely; this corresponds to the case in which the average number of children pre individual is more than 1.

There is of course an intermediate case, in which the average number of children is *exactly* 1. In this case, the colony becomes extinct with probability 1 but requires on average an infinite time to do that.

There are several algorithms for computing the extinction probabilities x for supercritical processes. Several fixed-point iterations have been studied in [1, 4, 3], e.g.,

$$x_{k+1} = a + b(x_k, x_k) \tag{1}$$

or

$$x_{k+1} = a + b(x_{k+1}, x_k). \tag{2}$$

The latter iteration can be computed explicitly since the map $b(\cdot, x_k)$ is a linear map from $\mathbb{R}^N$ to itself and thus is associated to a matrix, and the stochasticity properties ensure that

$$I - b(\cdot, x_k)$$

is a nonsingular M-matrix.

With this notation, the Newton method takes a similar form, namely

$$x_{k+1} = \big(I - b(\cdot, x_k) - b(x_k, \cdot)\big)^{-1} a. \tag{3}$$

A similar strategy has been suggested in [6], i.e., applying Newton's method to a slightly different fixed-point equation.

These fixed-point iterations share several properties, the most interesting one being that they converge monotonically to the minimal solution x_* when started from $x_0 = 0$, i.e.,

$$0 = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \ \leq \ldots,$$

and $x_k \to x_*$.

All of them have probabilitistic explanations, in that x_k can be seen as the probability of the colony becoming extinct under additional restrictions depending on k. The probabilistic interpretations are of great interest to the applied queuing theory community, as they provide better insight on the properties of the method.

The Newton method (3) has quadratic convergence for supercritical processes, but this convergence degrades to linear in the critical case; similarly, (1) and (2) converge linearly, but the convergence speed degrades to sublinear for critical processes. This makes sense intuitively in view of the probabilistic interpretation, since we need to "simulate" many more iterations to determine the asymptotical behavior of a critical process. In particular, it is a common feature of all these algorithms that the convergence is slower for critical or close-to-critical processes.

2 Perron vector-based algorithms

We wish to propose here an algorithm with a completely different origin and numerical behavior [7, 2, 8]. First of all, for ease of explanation, we make a change of variable by setting $y = e - x$. In this way, y_i is the complement of the extinction probability, i.e., the *survival probability* starting from a single individual. The equation becomes

$$y = b(e - y, y) + b(y, e), \tag{4}$$

or, setting $P_y := b(e - y, \cdot) + b(\cdot, e) \in \mathbb{R}^{N \times N}$,

$$y = P_y y. \tag{5}$$

Since this is a linear change of variable, there are little changes if we apply the traditional fixed point iterations to this reformulation. However, (5) can be interpreted as saying that y is the Perron vector of a nonnegative matrix depending on y itself. This suggests setting up the following fixed-point iteration:

$$y_{k+1} = \text{Perron vector of } P_{y_k}. \tag{6}$$

Perron vectors are determined up to a multiplicative constant; therefore, we have the additional issue of choosing a normalization for y_{k+1}. We make here the choice of imposing that the residual

$$y_{k+1} - P_{k+1} y_{k+1}$$

be orthogonal to a fixed nonnegative vector w. In principle, any $w > 0$ could work. However, the following result shows that there is a clear best candidate for w.

Theorem 1 ([7]). *Ley b_t, $t \leq 1$ define a smoothly varying family of supercritical Markovian binary trees that converge to a critical problem for $t \to 1$; Let v_t, y_t be the left and right Perron eigenvectors of*

$$R_t := b_t(e, \cdot) + b_t(\cdot, e), \tag{7}$$

which we suppose to be irreducible for t sufficiently close to 1. Then, the Jacobian J_t of the iteration map is such that

$$\lim_{t\to 1} \rho(J_t) = \left| 1 - \frac{v_1^T b_1(y_1, y_1) \cdot w^T y_1}{w^T b_1(y_1, y_1) \cdot v_1^T y_1} \right|$$

In particular, choosing $w = v_1$ the right-hand side vanishes and thus we have superlinear convergence in the limit $t \to 1$.

The work [2] suggests a deflation algorithm to deal with cases in which R_1 is not irreducible; essentially, we can split the problem into two smaller ones which can be solved one after the other. It is proved in the same paper that, when the algorithm converges, it always converges to the minimal solution rather than spurious ones.

Moreover, it is possible to construct a Newton-type algorithm based on the same fixed-point equation (6).

The previous result works whenever R_t as in (7) is irreducible in a neighbourhood of $t = 1$ (or, equivalently, for $t = 1$). Problems with a reducible matrix $R = b(e, \cdot) + b(\cdot, e)$ can be dealt with by reducing to two smaller problems, with a strategy which is analogue to (block) back-substitution for block triangular systems.

3 Numerical experiments

Numerical experiments reveal that, unlike the previous algorithm, the convergence is not in general monotone; indeed, for problems which are very far from the critical case (and thus "easy to solve" in the framework of the classical algorithms), this new method can fail to converge. Nevertheless, in close-to-critical problems, convergence is surprisingly fast. We report numerical experiments for two parameter-dependent problems in Figures 1 and 2. The upward spikes in the classical Newton algorithm (red line) correspond to values of λ for which the problem is critical; in these cases, the algorithm is slower as expected. However, the Perron-based algorithms (the fixed-point iteration (6) and its Newton-based variant) show the opposite behavior and become faster instead when the problem is close-to-critical.

Overall, the proposed algorithm is faster than the classical algorithms for close-to-critical problems, and is thus the recommended choice in these cases.

References

[1] Nigel G. Bean, Nectarios Kontoleon, and Peter G. Taylor. "Markovian trees: properties and algorithms". In: *Ann. Oper. Res.* 160 (2008), pp. 31–50. ISSN: 0254-5330.

[2] Dario A. Bini, Beatrice Meini, and Federico Poloni. "On the solution of a quadratic vector equation arising in Markovian binary trees". In: *Numer. Linear Algebra Appl.* 18.6 (2011), pp. 981–991. ISSN: 1070-5325. DOI: `10.1002/nla.809`.

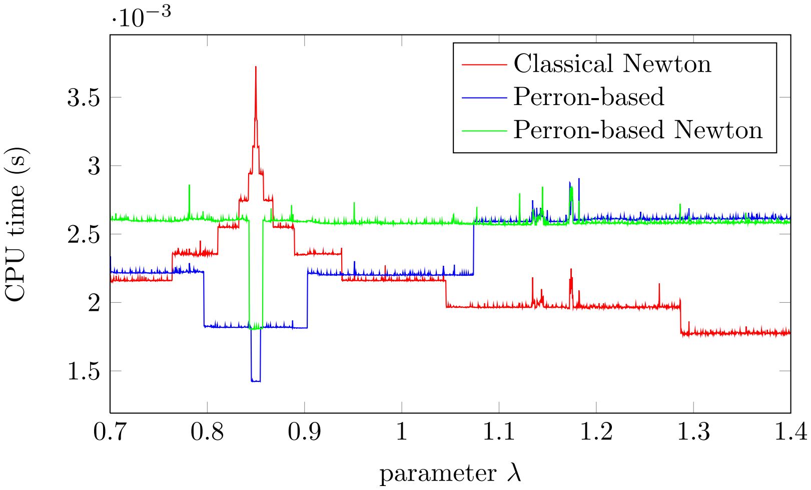

Fig. 1: Computational times vs. parameter for a parameter-dependent problem [4, Example 1]

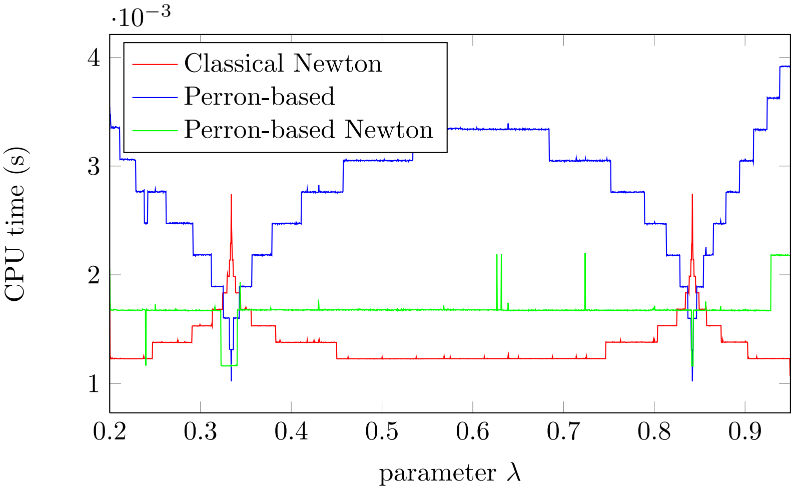

Fig. 2: Computational times vs. parameter for a parameter-dependent problem [4, Example 1]

[3] Sophie Hautphenne, Guy Latouche, and Marie-Ange Remiche. "Algorithmic Approach to the Extinction Probability of Branching Processes". In: *Methodology and Computing in Applied Probability* (2010).

[4] Sophie Hautphenne, Guy Latouche, and Marie-Ange Remiche. "Newton's iteration for the extinction probability of a Markovian binary tree". In: *Linear Algebra Appl.* 428.11-12 (2008), pp. 2791–2804. ISSN: 0024-3795.

[5] Sophie Hautphenne, Kenji Leibnitz, and Marie-Ange Remiche. "Modeling of P2P file sharing with a level-dependent QBD process". In: *Advances in queueing theory and network applications*. New York: Springer, 2009, pp. 247–263. DOI: 10.1007/978-0-387-09703-9_14.

[6] Sophie Hautphenne and Benny Van Houdt. "On the link between Markovian trees and tree-structured Markov chains". In: *European Journal of Operational Research* 201.3 (2010), pp. 791 –798. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2009.03.052.

[7] Beatrice Meini and Federico Poloni. "A Perron iteration for the solution of a quadratic vector equation arising in Markovian binary trees". In: *SIAM J. Matrix Anal. Appl.* 32.1 (2011), pp. 248–261. ISSN: 0895-4798. DOI: 10.1137/100796765.

[8] Federico Poloni. *Algorithms for quadratic matrix and vector equations*. Vol. 16. Tesi. Scuola Normale Superiore di Pisa (Nuova Series) [Theses of Scuola Normale Superiore di Pisa (New Series)]. Dissertation, Scuola Normale Superiore, Pisa, 2011. Edizioni della Normale, Pisa, 2011, pp. xvi+239. ISBN: 978-88-7642-383-3. DOI: 10.1007/978-88-7642-384-0.

[9] Federico Poloni. "Quadratic vector equations". In: *Linear Algebra and its Applications* (2011). to appear in press. ISSN: 0024-3795. DOI: 10.1016/j.laa.2011.05.036.