Explanation of the Frog-Star II Algorithm

Feng Cheng

In the self-similar Poisson(1) frog model $\mathrm{SFM}(p,d)$, let V be the total number of visits to the root. Then V has the following RDE:

$$V \overset{\mathrm{D}}{=} \mathrm{Pois}(p_d^*) + \mathrm{Binom}(V^{(1)}, \hat{p}) + \sum_{i=2}^{d} A_i \,\mathrm{Binom}(V^{(i)}, \hat{p}), \tag{1}$$

where $A_i := \mathbf{1}_{\{\text{subtree at } v_i \text{ is activated}\}}$, $V^{(1)}, \dots, V^{(d)}$ are independent copies of V, $p_d^* = \frac{p(d-1)}{d-(d+1)p}$, and $\hat{p} = \frac{p}{1-p}$. Note that we will be abusing our notation here: the Poisson and Binomial's above refer to their independent random variables.

Since the root visits have $V_{\mathrm{SFM}(d,p)} \preceq V_{\mathrm{nbFM}(d,p)} \preceq V_{\mathrm{FM}(d,p)}$, proving $\mathrm{P}(V_{\mathrm{SFM}(d,p)} = \infty) = 1$ implies that $\mathrm{P}(V_{\mathrm{FM}(d,p)} = \infty) = 1$. Our idea is to show that for any $\lambda > 0$, given $V \succeq \mathrm{Pois}(\lambda)$, it follows that $V \succeq \mathrm{Pois}(\lambda + \epsilon)$ for a fixed $\epsilon > 0$. Since from (1) we have $V \succeq \mathrm{Pois}(p_d^*)$, by repeated application of what we mentioned we can show that $\mathrm{P}(V_{\mathrm{SFM}(d,p)} = \infty) = 1$.

Assume $V \succeq \mathrm{Pois}(\lambda)$, then we may replace independent $V^{(1)}, \dots, V^{(d)}$ in (1) by independent $\mathrm{Pois}(\lambda)$ random variables and get

$$\begin{aligned} V &\succeq \mathrm{Pois}(p_d^*) + \mathrm{Binom}(\mathrm{Pois}(\lambda), \hat{p}) + \sum_{i=2}^{d} A_i \,\mathrm{Binom}(\mathrm{Pois}(\lambda), \hat{p}), \\ &= \mathrm{Pois}(p_d^*) + \mathrm{Pois}(\lambda\hat{p}) + \sum_{i=2}^{d} A_i \,\mathrm{Pois}(\lambda\hat{p}). \end{aligned} \tag{2}$$

Now consider the auxilliary model *frog-star ii* below[1], which takes out the first three layers of the tree and replaces the distribution of each $v_j \to \varnothing'$ by $\mathrm{Pois}(\lambda) \preceq V$. Now define W to be the number of activated vertices among $v_2, \dots, v_d$ in this frog-star ii. *Note that the p.m.f. of W is now computable because we are in a finite model.* By our setup we also have $W \preceq \sum_{i=2}^{d} A_i$. Hence from (2) onward we have

$$\begin{aligned} V &\succeq \mathrm{Pois}(p_d^*) + \mathrm{Pois}(\lambda\hat{p}) + \sum_{i=2}^{d} A_i \,\mathrm{Pois}(\lambda\hat{p}) \\ &\succeq \mathrm{Pois}(p_d^*) + \mathrm{Pois}(\lambda\hat{p}) + W \cdot \mathrm{Pois}(\lambda\hat{p}) \\ &= \mathrm{Pois}\big(p_d^* + \lambda\hat{p}(1+W)\big). \end{aligned}$$

We now need to find under what λ is $\mathrm{Pois}\big(p_d^* + \lambda\hat{p}(1+W)\big) \succeq \mathrm{Pois}(\lambda + \epsilon)$ for some fixed $\epsilon > 0$.

[1] To be specific, it has Poisson(1) frogs at $\varnothing'$ and Poisson(λ) frogs at each of the leaves $v_1, \dots, v_d$. $\varnothing'$ and v_1 are activated at the beginning. The initial frogs at $\varnothing'$ move to $\varnothing$ with probability p_d^*, and to the leaves uniformly with total probability $1 - p_d^*$. Active frogs at leaf v_j moves to $\varnothing'$ with probability 1, and then to $\varnothing$ with probability $\hat{p}$, and to $v_1, \dots, v_{j-1}, v_{j+1}, \dots, v_d$ uniformly with total probability $1 - \hat{p}$.

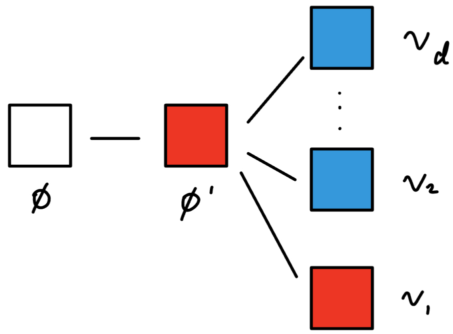


Figure 1: frog-star ii model

This is equivalent to proving that $\mathbf{E}\exp\big(p_d^* + \lambda\hat{p}(1+W)\big) \le \exp(-\lambda-\epsilon)$. Simplification gives us

$$\begin{aligned}\mathbf{E}\exp\big(-p_d^* - \lambda\hat{p}(1+W)\big) &= e^{-p_d^*}\sum_{w=0}^{d-1}\exp\big(-\lambda\hat{p}(1+w)\big)\,\mathrm{P}(W=w)\\ &= e^{-p_d^*-\lambda}\sum_{w=0}^{d-1}\exp\big(\lambda(1-\hat{p}(1+w))\big)\,\mathrm{P}(W=w),\end{aligned}$$

and thus

$$e^{-p_d^*-\lambda}\sum_{w=0}^{d-1}\exp\big(\lambda(1-\hat{p}(1+w))\big)\,\mathrm{P}(W=w) \le \exp(-\lambda-\epsilon)$$

$$\Downarrow$$

$$e^{-p_d^*}\sum_{w=0}^{d-1}\exp\big(\lambda(1-\hat{p}(1+w))\big)\,\mathrm{P}(W=w) \le e^{-\epsilon},\ \text{ where } e^{-\epsilon} \ge 1-\epsilon.$$

To remain consistent with the literature, let $m = d$ and $\tilde{p} = p_d^* = \frac{p(m-1)}{m-(m+1)p}$. Under a fixed m, we want to numerically compute the minimal p such that

$$h(\lambda) = e^{-\tilde{p}}\sum_{w=0}^{m-1}\exp\big(\lambda(1-\hat{p}(1+w))\big)\,\mathrm{P}(W=w) \le 1-\epsilon \tag{3}$$

holds for all $\lambda > 0$.

The p.m.f. of W can be computed via the following recursion algorithm[2]. Define φ to be the probability that 0 of the initial frogs at $\varnothing'$ moves to v_2, and v to be the probability that 0 of the initial frogs at v_1 moves to v_2. It is clear that $\varphi = \exp\big(-\frac{1-\tilde{p}}{m}\big)$ and $v = \exp\big(-\frac{\lambda(1-\hat{p})}{m-1}\big)$. Now let $p_{m,k} = \mathrm{P}_m(W=k)$ be the probability of $W=k$ when the frog-star ii has m leaves. With the initial

[2] c.f. Eric Yu

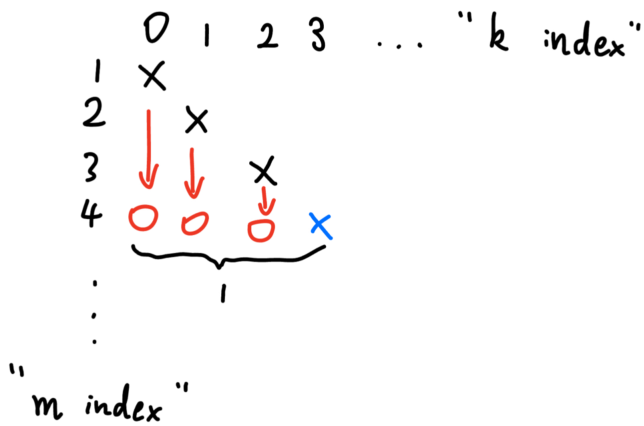
condition $p_{1,0} = 1$, the recursion is established by the following two equations:

$$p_{m,k} = \binom{m-1}{k}(\varphi v^{k+1})^{m-1-k} \cdot p_{k+1,k} \text{ when } 0 \le k \le m-2, \text{ and} \tag{4}$$

$$p_{m,m-1} = 1 - \sum_{i=0}^{m-2} p_{m,i}. \tag{5}$$

To visualize how to find the p.m.f. of W, consider the following table. Here the entry in the m-th row

Figure 2: recursion table



and k-th column corresponds to $p_{m,k}$. Take $m = 4$ for example; $p_{4,0}, p_{4,1}$, and $p_{4,2}$ are all determined by the diagonal elements determined above [by equation (4)]. Since every row sums up to 1, the new diagonal element $p_{4,3}$ can be computed by $1 - \sum_{i=0}^{2} p_{4,i}$ [equation (5)].

Hence it makes sense to implement the p.m.f. computation using a symbolic matrix. Be aware that Python-based SageMath is 0-indexed, and hence we have to shift the row index m here back by 1 (replaced by j in our code).

For numerical purposes we let p be an irreducible fraction $\frac{a}{b} < \frac{1}{2}$, and then

$$\hat{p} = \frac{a}{b-a},\ 1 - \hat{p} = \frac{b-2a}{b-a},\ \tilde{p} = \frac{am-a}{bm-am-a}, \text{ and } 1 - \tilde{p} = \frac{bm-2am}{bm-am-a}. \tag{6}$$

The way we want to approach equation (3) is that we want to convert $h(\lambda)$ into a polynomial $g(y)$ by mapping $\lambda \mapsto -c \cdot \log(y)$ for some constant c. Through this change of variable it suffices to consider that

$$\sup_{\lambda \in (0,\infty)} h(\lambda) = \sup_{y \in (0,1)} g(y) = \max_{y \in [0,1]} g(y) \le 1 - \epsilon.$$

It turns out a working constant c can be chosen in general given m and $p = \frac{a}{b}$: let

$$c = (b-a)(m-1).$$

Now we verify our choice. In the p.m.f. of W, we have terms $\varphi = \exp\left(\frac{2a-b}{bm-am-a}\right)$ and $v = y^{b-2a}$ by (6). Now we rewrite our objective function $h(\lambda)$ from (3) as follows:

$$\begin{aligned} g(y) &= \exp\left(\frac{a-am}{bm-am-a}\right) \sum_{k=0}^{m-1} y^{[-(b-a)(m-1)]\cdot\left(\frac{b-2a}{b-a} - \frac{a}{b-a}k\right)} \cdot \mathrm{P}(W=k) \\ &= \exp\left(\frac{a-am}{bm-am-a}\right) \sum_{k=0}^{m-1} y^{(m-1)[(k+2)a-b]} \cdot \mathrm{P}(W=k). \end{aligned} \tag{7}$$

It remains to show that the exponents of y in (7) are indeed nonnegative integers. It is easy to show by induction that all $p_{m,k}$'s in the lower triangle of the recursion table only have nonnegative integer powers of y.

Then for $0 \le k \le m-2$, we know from (4) that the exponents of y in $\mathrm{P}(W=k)$ are all at least $(b-2a)(k+1)(m-1-k)$. Thus in (7), the exponents in each summand from $k=0$ to $m-2$ are at least

$$\begin{aligned} &(m-1)ak - (m-1)(b-2a) + (b-2a)(k+1)(m-1-k) \\ =&(b-2a)(m-2-k)k + (m-1)ak \ge (m-1)ak \ge 0. \end{aligned}$$

We now consider the summand when $k = m-1$, i.e., $y^{(m-1)[(m+1)a-b]} \cdot \mathrm{P}(W=m-1)$. By (5) it suffices to show that $(m+1)a - b \ge 0$, i.e., $p = \frac{a}{b} \ge \frac{1}{m+1}$. This lower bound to the critical drift p_m was mentioned briefly in [BFJ+19]; basically we want to show that

> assuming all frogs are initially awake in the original one-per-site model [meaning that this is a more recurrent model], if $p < 1/(m+1)$, then there are only finite expected visits in total to the root.

Eric Yu wrote a proof of this in the general Discord channel early in June. The overall idea is that we wish to write the expected number of visits into a infinite sum of root visits E_n from vertices *at each layer n to the root*. It is quite clear that these random variables have their internal recursive relation, from which we can derive an explicit formula for these E_n's. The summation will then become a geometric series. Since a more recurrent model is transient for $p < 1/(m+1)$, $p_m \ge 1/(m+1)$ as desired.

It is really nice to prove that our substitution constant *always* gives a polynomial $g(y)$, but ultimately we can use the computation algorithm below only up to a finite m. Showing that this substitution gives integer exponents and visually checking that $g(y)$ is indeed a polynomial might be an alternative (albeit more tedious) approach.

Now we start explaining our algorithm. The only thing that needs to be adjusted is $p = a/b$ based on our choice of m. The symbolic expression of the polynomial $g(y)$ will then be computed accordingly. As preliminary exploration we shall visualize the graph of $g(y)$ over the interval $[0, 1]$.

It is almost always the case by visual inspection that $g(y)$ attains its only peak on the interval $(0,1)$ when y is somewhere close to 1, or equivalently, the original λ is not too far from 0.

The `CountRoots` method specifically from Mathematica is of great importance to us. In our context we use the Mathematica code `CountRoots[dg[y],{y,0,1}]` (which we imported into SageMath). This method uses Sturm's algorithm to count the number of roots (including multiplicities) on the designated closed interval $[0,1]$. Call it `count`. By our previous visual inspection we hope that this `count` is only 1 more than the multiplicity of the root 0. This multiplicity can be found easily by factoring $g'(y)$ in SageMath, which gives

$$(\text{some polynomial with nonzero constant}) \cdot y^t \cdot (\text{some constant}).$$

Hence this t is the multiplicity. If `count` $= t+1$, then there is only one root of $g'(y)$ in $(0,1]$, which can be found numerically using `diff(g(y)).find_root(0.9,1)`. This finds a root of $g'(y)$ on the closed interval $[0.9,1]$[3] using Brent's method, which in our case should be the only root y_0 in $[0.9,1] \subseteq (0,1]$.

Now there are two ways to check whether $y_0 = \arg\max_{y\in[0,1]} g(y)$ that I can think of. One approach is to use the second derivative test at y_0: just verify $g''(y_0) < 0$ numerically. I am more in favor of the second approach, which is precise and avoids using g'' or g'. We know $g(y)$ attains its maximum on the closed interval $[0,1]$. If we could check that $g(y_0) > g(0)$ and $g(y_0) > g(1)$, then the maximum of $g(y)$ is attained in the open interval $(0,1)$. By Fermat's theorem[4] we know that this maximum must be attained at $y = y_0$, the only point in $(0,1)$ such that $g'(y) = 0$.

We are only left to check if $g(y_0) = \max_{y\in(0,1)} g(y) \leq 1-\epsilon$. If this is correct, then our choice of $p = a/b$ gives an upper bound to the critical drift $p_m = \inf\{p : \mathrm{FM}(m,p) \text{ is recurrent}\}$.

On the next page I will give a list of optimal upper bounds p that can be achieved numerically by the steps above. I only included m between 2 and 13. The Brent's method used in `find_root` starts to give error when $m \geq 14$ even for p with relatively small a and b. The length of the polynomial $g(y)$ also becomes extremely long, and I doubt if we can go any further than $m = 15$.

[3]As I mentioned earlier, by visual inspection we know that $\max_{y\in[0,1]} g(y)$ should be achieved somewhere close to 1. Thus I choose 0.9 here.

[4]https://en.wikipedia.org/wiki/Fermat%27s_theorem_(stationary_points)

m	upper bound $p =$
2	$55/159 < 0.3460$
3	$31/107 < 0.2898$
4	$17/65 < 0.2616$
5	$23/94 < 0.2447$
6	$46/197 < 0.2336$
7	$23/102 < 0.2255$
8	$29/132 < 0.2197$
9	$20/93 < 0.2151$
10	$11/52 < 0.2116$
11	$5/24 < 0.2084$
12	$7/34 < 0.2059$
13	$11/54 < 0.2038$

References

[BFJ+19] Erin Beckman, Natalie Frank, Yufeng Jiang, Matthew Junge, and Si Tang. "The frog model on trees with drift". In: *Electronic Communications in Probability* 24.26 (2019), pp. 1–10. DOI: 10.1214/19-ECP235.