ONLINE LEARNING FOR LATENT YULE-SIMON PROCESSES

Asher A. Hensley and Petar M. Djurić

Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, 11794 NY

ABSTRACT

Yule-Simon processes are one of the most commonly occurring processes in Nature. These processes generate power laws using a preferential attachment mechanism which can describe a variety of data distributions such as word frequencies, scientific citations, journal publications, income, node connections in complex networks, biological genera, and bosons in quantum states. Much of the work in this area has focused on modeling the properties of observable quantities such as these. In this work we focus on learning the properties of *unobservable* Yule-Simon processes which control the dynamics of sequential sensor measurements. This is motivated by the fact that Yule-Simon processes have a varying memory length which offer a more general framework for data modeling than hidden Markov models. In this paper we present an approximate online learning procedure based on multiple hypothesis pruning which is shown to reach 0.5dB of the posterior Cramer-Rao lower bound.

***Index Terms*—** Bayesian filtering, regime switching, time series, state estimation, power law.

1. INTRODUCTION

In this work we present a real-time learning procedure for sequentially arriving sensor measurements whose properties are controlled by a hidden Yule-Simon process. Yule-Simon processes have a varying memory length and therefore offer a more general framework for data modeling than Markov processes which have trouble modeling waiting times within a given state [1]. In particular, Yule-Simon processes have a preferential attachment mechanism [2], where the probability of state transition is inversely proportional to the time within the current state. A common example of this appears in finance where price volatility increases the probability of future price volatility, a phenomenon known as volatility clustering [3].

The Yule-Simon model was first proposed by Yule in 1925 [4] to explain the power law distribution observed in biological genera. The same result was developed by Simon in 1955 using a different mathematical approach to explain word frequency distributions in texts [5]. The Yule-Simon model has since been applied and/or rediscovered in a variety of other areas to develop scientific models in network theory [6], astrophysics [7], and quantum mechanics [8]. The history of the Yule-Simon model and several closely related processes, such as branching processes [9] are surveyed exhaustively in [10].

Prior studies more aligned with this work have focused primarily on inference problems for observable Yule-Simon processes. Garcia proposes a fixed point algorithm based on maximum likelihood in [11], Roberts describes an expectation maximization approach in [12], and Leisen provides Markov Chain Monte Carlo (MCMC) sampling methods for Bayesian inference in [13, 14]. However, the literature in this area is surprisingly sparse. This work differs from these studies by focusing on the inference problem for *unobservable* Yule-Simon processes and extends prior work on batch inference using MCMC techniques in [15].

2. PRELIMINARIES

The Yule-Simon probability distribution is based on the Geometric-Exponential model given by,

$$w|\alpha \sim \text{Exponential}(\alpha), \tag{1}$$

$$k|w \sim \text{Geometric}(e^{-w}). \tag{2}$$

The Yule-Simon probability distribution is then defined as the marginal distribution of k,

$$p(k|\alpha) = \int_0^\infty p(k|w)p(w|\alpha)dw = \alpha B(k, \alpha+1), \tag{3}$$

where $B(x, y)$ is the beta function. Examples of the Yule-Simon distribution with different values of α are shown in Fig. 1 for both linear and logarithmic scales. The contrast between these plots is rather dramatic and helps to give a sense of how "heavy" the tails are. As k becomes large, the Yule-Simon distribution approaches

$$p(k|\alpha) \propto k^{-(\alpha+1)}, \tag{4}$$

where the α parameter can be interpreted as the power law exponent which controls the tail behavior.

The Yule-Simon distribution can also be viewed as a stochastic process where a sequence of i.i.d. values of k are generated sequentially. Thus, rather than drawing each k

The authors thank the support of NSF under Award 2021002.

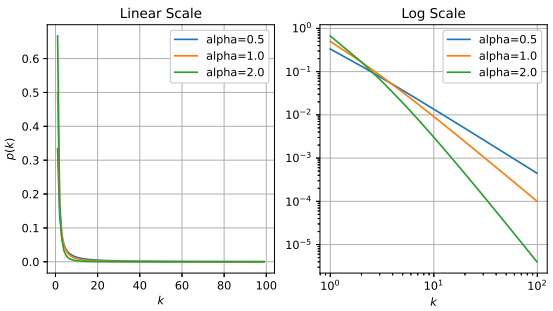

Fig. 1. Yule-Simon distribution examples.

from the Geometric distribution conditioned on w, we can use the definition of the Yule-Simon distribution directly. In particular, we can view this as a special type of sequential Polya urn scheme [16].

To see how this works, assume we have an urn with one white ball and α black balls. The process proceeds by sequentially drawing balls from the urn until the first black ball is drawn. Each time a white ball is drawn, it is replaced in the urn and an additional white ball is added. The number of draws required to observe the first black ball is a Yule-Simon random variable. We can generate a sequence of i.i.d. Yule-Simon random variables by simulating this process and resetting the urn to its initial state each time a black ball is observed. The state transition diagram of the Yule-Simon process is shown in Fig. 2, where the state variable is the number of white balls in the urn.

3. GENERATIVE MODEL

The sequential Yule-Simon process can be formulated as a generative model where we do not observe the Yule-Simon process directly. Instead, we observe another process which is being controlled by the Yule-Simon process. Specifically, the properties of the observable process change each time the Yule-Simon process is reset to its initial state.

For this formulation, it is convenient to define a Yule-Simon counting process which can be written as a Bernoulli process with memory,

$$s_t \sim \text{Bernoulli}(\alpha(n_t + \alpha)^{-1}), \tag{5}$$
$$x_t = x_{t-1} + s_t, \tag{6}$$
$$n_t = (n_{t-1} + 1)^{1-s_t}, \tag{7}$$

where the x_t counts how many times the process has reset itself (i.e., number of black balls drawn) and n_t counts how many white balls have been drawn in a row. The complete generative model can be written as follows:

$$s_t \sim \text{Bernoulli}(\alpha(n_t + \alpha)^{-1}) \tag{8}$$
$$x_t = x_{t-1} + s_t, \tag{9}$$
$$n_t = (n_{t-1} + 1)^{1-s_t}, \tag{10}$$
$$\theta_k \sim \pi(\theta), \tag{11}$$
$$z_t \sim p(z|\theta_{x_t}). \tag{12}$$

The observable process is based on a measurement distribution $p(z|\theta)$ where the parameter θ controls the properties of the observations z. The process proceeds by first randomly drawing an infinite set of parameters $\{\theta_k\}_{k=1}^{\infty}$ from the prior $\pi(\theta)$. The state of the Yule-Simon counting process x_t is then updated on each discrete time sample t and used to index into the infinite set of parameters to sample the measurement z_t from $p(z|\theta_{x_t})$.

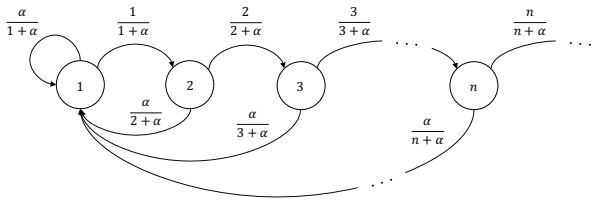

Fig. 2. The Yule-Simon state transition diagram.

4. INFERENCE ALGORITHM

4.1. Full Bayesian Solution

The full Bayesian solution is based on a simple idea: compute the posterior distribution over all possible state trajectories, and then select the one with the highest probability:

$$\hat{\mathbf{x}}_T = \underset{\mathbf{x}_T}{\text{argmax}}\ p(\mathbf{x}_T|\mathbf{z}_T), \tag{13}$$

where $\mathbf{x}_T$ and $\mathbf{z}_T$ denote the set all values of x_t and z_t from 1 to T. However, the problem with this approach is the number of possible state trajectories is 2^T which becomes intractable very quickly. Therefore, we turn to an approximate inference strategy based on multiple hypothesis pruning.

4.2. Multiple Hypothesis Solution

The multiple hypothesis solution approximates the full solution by setting a maximum number of active branches m and pruning branches with low probability using a predict/update recursion. We begin by defining a set of vectors specific to a computer based implementation: $\mathbf{p}_t, \mathbf{q}_t, \boldsymbol{\phi}_t$, and $\mathbf{n}_t$.

The $\mathbf{p}_t$ vector denotes the set of m active path probabilities and $\mathbf{q}_t$ represents the set of $2m$ predicted path probabilities (recall the number of possible state trajectories doubles on each measurement). The symbol $\boldsymbol{\phi}_t$ stands for the hyper-parameters of the distributions of θ for each of the $2m$

predicted paths. Moving forward, we will assume that the distribution of θ is conjugate to the distribution of z. In doing so we can maintain the posterior distributions of θ for each of the active and predicted paths as more data are received which will be required to compute the state posterior marginals. Finally, the vector $\mathbf{n}_t$ denotes the number of samples received since the last partition reset (i.e., the number of white balls drawn in a row from the Yule-Simon urn) for each of the $2m$ predicted paths.

4.3. Initialization

We begin by first specifying m, α, and ϕ_0 and then initializing $\mathbf{p}_t$, $\mathbf{q}_t$, $\boldsymbol{\phi}_t$, and $\mathbf{n}_t$ to their respective sizes (m, $2m$, $2m$, and $2m$ respectively) with all zero values. Next we observe the first measurement and set the initial state:

$$p_1[1] = n_1[1] = 1, \quad \phi_1[1] = h(z_1, \phi_0). \tag{14}$$

The Yule-Simon process always starts in state 1 with probability 1 based on a single measurement, which is why $p_1[1] = n_1[1] = 1$. The function h refers to the update equation for the θ posterior hyper-parameters, which will depend on the choice of measurement and prior distributions. We give an example of this in the analysis section for the Gaussian process model. Here we use ϕ_0 as an initial guess.

4.4. Prediction

The prediction step computes the splitting probabilities for all active branches:

$$q_t[k] = \frac{n_{t-1}[k]}{(n_{t-1}[k] + \alpha)} p_{t-1}[k], \tag{15}$$

$$q_t[k+m] = \frac{\alpha}{(n_{t-1}[k] + \alpha)} p_{t-1}[k]. \tag{16}$$

This can be done in a *for loop* from $k = 1$ to m.

4.5. Update

The update step computes the posterior probabilities for each branch with the corresponding θ hyper parameters given the most recent measurement. The θ hyper parameters are updated as follows:

$$\boldsymbol{\phi}_t[I] = h(z_t, \boldsymbol{\phi}_{t-1}[I]), \tag{17}$$

where $I = \mathbf{q}_t > 0$ is a logical mask used to only update active branches. The predicted path probabilities are then updated with the marginal likelihoods in another for loop from $k = 1$ to m:

$$q_t[k] = F(z_t, \phi_t[k]) q_t[k], \tag{18}$$

$$q_t[k+m] = F(z_t, \phi_t[k+m]) q_t[k+m], \tag{19}$$

where

$$F(z_t, \phi_t[k]) = \int p(z_t|\theta) p(\theta|\phi_t[k]) d\theta, \tag{20}$$

and then normalized upon completion, or

$$q_t[j] = \frac{q_t[j]}{\sum_{k=1}^{2m} q_t[k]} \tag{21}$$

for $j = 1$ to $2m$. Finally, before proceeding to the hypothesis pruning step, we update the state counters by

$$\mathbf{n}_t[I] = \mathbf{n}_{t-1}[I] + 1. \tag{22}$$

4.6. Hypothesis Pruning

Hypothesis pruning proceeds by sorting $\mathbf{q}_t$ in descending order and reordering $\boldsymbol{\phi}_t$ and $\mathbf{n}_t$ accordingly:

$$\mathbf{q}_t = \mathbf{q}_t[J], \quad \boldsymbol{\phi}_t = \boldsymbol{\phi}_t[J], \quad \mathbf{n}_t = \mathbf{n}_t[J], \tag{23}$$

where $J = \text{argsort}(\mathbf{q}_t)$. The posterior branch probabilities are then updated with the first m values of $\mathbf{q}_t$ according to

$$\mathbf{p}_t = \mathbf{q}_t[1:m] \tag{24}$$

and re-normalized. The discarded branches are then reinitialized in preparation for the next iteration, that is,

$$\boldsymbol{\phi}_t[m+1:2m] = \phi_0, \tag{25}$$

$$\mathbf{n}_t[m+1:2m] = 0. \tag{26}$$

4.7. Parameter Estimation

After the predict and update steps, posterior estimates of θ can be made based on the most likely trajectory. This consists of taking the expected value of θ given $\phi_t[1]$.

5. ANALYSIS

5.1. Gaussian Process Case Study

As a case study, we consider a zero mean Gaussian process model given by:

$$s_t \sim \text{Bernoulli}(\alpha(n_t + \alpha)^{-1}), \tag{27}$$

$$x_t = x_{t-1} + s_t, \tag{28}$$

$$n_t = (n_{t-1} + 1)^{1-s_t}, \tag{29}$$

$$\lambda_k \sim \text{Gamma}(\lambda|a, b), \tag{30}$$

$$z_t \sim \text{Normal}(z|0, \lambda_{x_t}^{-1}). \tag{31}$$

Here $\theta = \lambda$, which is the precision (or inverse variance) of the Normal distribution, the hyper parameters are given by $\phi = [a, b]$, and the function h is simply:

$$h(z_t, [a, b]) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + 0.5 \begin{bmatrix} 1 \\ z_t^2 \end{bmatrix}. \tag{32}$$

To get an idea of what this process looks like, several sample realizations are shown in Fig. 3 for the case of $a = b = \alpha = 1$.

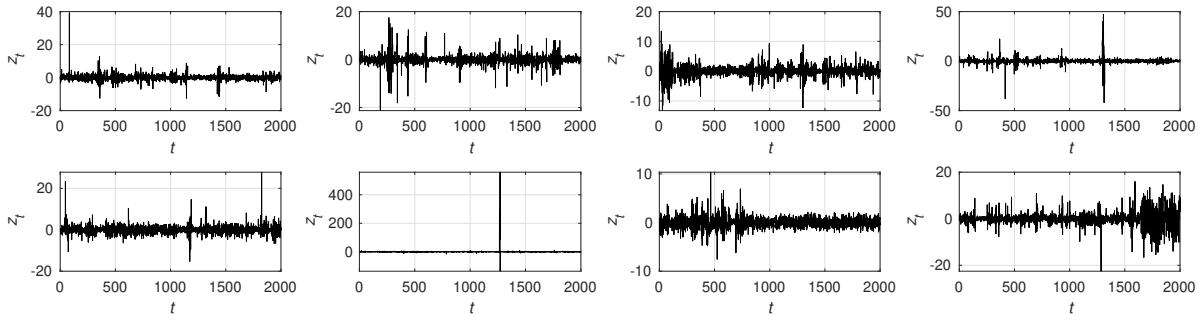

Fig. 3. Examples of Yule-Simon Gaussian process realizations used for performance evaluation.

5.2. Evaluation Strategy

A scalar linear Gaussian state space model is given by the following process and measurement equations:

$$\mu_t = F\mu_{t-1} + v_t, \quad v_t \sim \text{Normal}(0, Q_t), \tag{33}$$

$$y_t = H\mu_t + w_t, \quad w_t \sim \text{Normal}(0, R_t). \tag{34}$$

The posterior Cramer Rao Lower Bound (CRLB) [17] of the mean square error for the state variable μ_t is given by the variance update equation for the Kalman filtering algorithm [18]. The implication of this being the Kalman filter is the optimal estimation algorithm for the linear Gaussian system when Q_t and R_t are known for all t. To evaluate the multiple hypothesis inference algorithm, we note that the Yule-Simon Gaussian process can be interpreted as a trivial case of the linear Gaussian model where $F = H = 1$ and $Q_t = \mu_t = 0$:

$$\mu_t = \mu_{t-1}, \tag{35}$$

$$y_t = \mu_t + z_t. \tag{36}$$

Therefore, the strategy is to simulate sample realizations, run the multiple hypothesis algorithm, and then use the inferred values of $R_t = \lambda_{x_t}^{-1}$ to run the Kalman filtering algorithm. The results are then compared to the posterior CRLB which is computed with the true values of R_t.

5.3. Simulation Results

For evaluation, we generated 1000 realizations of the Yule-Simon Gaussian process with a duration of 2000 samples each with $a = b = \alpha = 1$. We then ran the multiple hypothesis inference algorithm with $\log_2 m = 2, 4, 6, 8$, and 10 on each realization and used the inferred values of λ to run the Kalman filtering algorithm as described above. The squared error between each Kalman filter update and the true mean (in this case 0) was computed for each time sample and then averaged across all 1000 realizations. The hyperparameters a, b, and α were assumed to be known. In addition, we computed the posterior CRLB with the true values of λ for each time sample and averaged the results across all 1000 realizations.

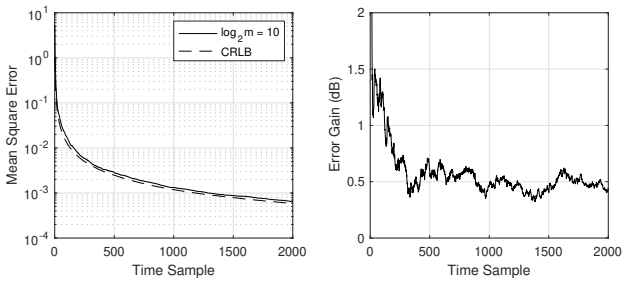

Fig. 4. MSE (left) and error gain (right) for $\log_2 m = 10$.

$\log_2 m$	Error Gain
2	1.19 dB
4	0.66 dB
6	0.56 dB
8	0.53 dB
10	0.50 dB

Table 1. Average error gain results.

To analyze the results we defined the error gain as the following metric:

$$G = 10\log_{10}(\text{MSE}/\text{CRLB}). \tag{37}$$

The average error gain as a function of $\log_2 m$ ($t > 500$) is given in Table 1. Results for the best case of $\log_2 m = 10$ are given in Fig. 4.

5.4. Source Code

`https://github.com/AsherHensley/ICASSP_2022_Supplemental_Material`

6. CONCLUSION AND FUTURE WORK

The simulation results suggest the multiple hypothesis solution converges to the CRLB in the limit as $\log_2 m$ approaches ∞. Future work will focus on showing this analytically, investigating sensitivity to mismatched hyperparameters, and developing additional measurement models.

7. REFERENCES

[1] Yves Boussemart and Mary L. Cummings, "Predictive models of human supervisory control behavioral patterns using hidden semi-markov models," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1252–1262, 2011.

[2] Mark E.J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, pp. 025102, 2001.

[3] Benoit Mandelbrot, "The variation of certain speculative prices," *The Journal of Business*, vol. 36, no. 4, pp. 394–419, 1963.

[4] George Udny Yule, "A mathematical theory of evolution, based on the conclusions of Dr. J.C. Willis, FR S," *Philosophical Transactions of the Royal Society of London. Series B*, vol. 213, no. 402-410, pp. 21–87, 1925.

[5] Herbert A. Simon, "On a class of skew distribution functions," *Biometrika*, vol. 42, no. 3/4, pp. 425–440, 1955.

[6] Albert-László Barabási and Réka Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[7] Enrico Fermi, "On the origin of the cosmic radiation," *Physical Review*, vol. 75, no. 8, pp. 1169, 1949.

[8] Mikhail V. Simkin and E.G.D. Cohen, "Magnetic properties of a Bose-Einstein condensate," *Physical Review A*, vol. 59, no. 2, pp. 1528, 1999.

[9] Theodore Edward Harris et al., *The theory of branching processes*, vol. 6, Springer Berlin, 1963.

[10] Mikhail V. Simkin and Vwani P. Roychowdhury, "Reinventing Willis," *Physics Reports*, vol. 502, no. 1, pp. 1–35, 2011.

[11] Juan Manuel Garcia Garcia, "A fixed-point algorithm to estimate the Yule–Simon distribution parameter," *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8560–8566, 2011.

[12] Lucas Roberts and Denisa Roberts, "An expectation maximization framework for Yule-Simon preferential attachment models," *arXiv preprint arXiv:1710.08511*, 2017.

[13] Fabrizio Leisen, Luca Rossini, and Cristiano Villa, "A note on the posterior inference for the Yule–Simon distribution," *Journal of statistical computation and simulation*, vol. 87, no. 6, pp. 1179–1188, 2017.

[14] Fabrizio Leisen, Luca Rossini, and Cristiano Villa, "Objective Bayesian analysis of the Yule–Simon distribution with applications," *Computational Statistics*, vol. 33, no. 1, pp. 99–126, 2018.

[15] Asher A. Hensley and Petar M. Djurić, "Nonparametric learning for hidden Markov models with preferential attachment dynamics," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3854–3858.

[16] Norman Lloyd Johnson and Samuel Kotz, "Urn models and their application; an approach to modern discrete probability theory," 1977.

[17] Petr Tichavsky, Carlos H. Muravchik, and Arye Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[18] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*, Artech house, 2003.