1

Probablistic Inference with Generating Functions for Animal Populations

Daniel Sheldon[a][b] and Kevin Winner[a] and Debora Sujono[a]

Abstract

Population sizes and demographic rates are commonly estimated by fitting probabilistic population models to observational data. However, these models expose a gap in our probabilistic inference toolkit. They often contain count variables to represent unknown population sizes, and, until recently there was no known algorithm to exactly compute

[a] College of Information and Computer Sciences, University of Massachusetts Amherst

[b] Department of Computer Science, Mount Holyoke College

the likelihood in models with latent count variables. This chapter summarizes recent advances in the AI research area of probabilistic inference motivated by this gap. We describe how a standard probabilistic inference algorithm, the forward algorithm, can be adapted to use *probability generating functions* as its internal representation of probability distributions. This leads to the first exact and efficient algorithms for these models. The new algorithms apply to a broad class of population models, are faster than existing approximate algorithms, and lead to improved behavior for estimating parameters of animal populations.

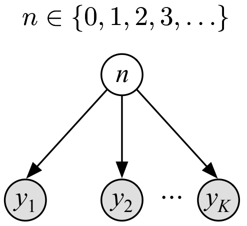
1.1 Introduction

Estimating the size and demographic parameters of animal populations is key to effective conservation. This is commonly done by using counts of animals made by human observers to estimate the parameters of a population model. When some variables from the population model are not directly observed—for example, the number of animals that are *not* detected by the observer, or the number of animals that leave a habitat patch between two consecutive surveys—an inference algorithm is required to reason about hidden events while fitting the model.

Probabilistic inference is a challenging computational problem, and a great deal of AI research over the last 20 years has been devoted to developing efficient and general probabilistic inference algorithms. Despite great advances, models can still be “hard” for several reasons. One reason is model size and complexity, for example, as measured by the number of variables and the number, structure, and type of functional relationships among variables. There is a natural trend toward more complex models in ecology as we collect large and diverse data sets through efforts such as citizen science [Sullivan et al., 2009]. Developing efficient probabilistic inference algorithms to reason about complex ecological models from growing data resources is an important research direction.

This chapter will focus on a second property of population models that can make inference difficult: the presence of count variables to represent the unknown population size. A simple example is the *N-mixture* model [Royle, 2004] illustrated in Figure 1.1. Here, the variable n is an integer representing the unknown number of animals in a patch of habitat; because this number is not directly observed, it is a *latent* variable. An observer visits the patch K times, and the variable y_k represents the number of animals she is able to detect during the the kth survey; these are the *observed* variables. This model very simple by typical measures of complexity, but, surprisingly, until very recently there was *no known* *exact inference algorithm* for this model. The computational challenge arises because the algorithm must reason about the infinite number of possible values (any non-negative integer) for the latent variable. Widely used estimation procedures for population models with latent count variables all resort to some form of approximation, usually by assuming an *a priori* upper bound on the population size. This has several drawbacks. First, it places a burden on the modeler, when we would like to let the data speak for itself. Second, it interacts poorly with estimation of the detection probability, which determines the approximate "multiplier" between the observed counts and the true population size; we demonstrate pathologies related to this later in the chapter. Finally, this method is slow, especially when reasoning about populations over time. We desire *fast* algorithms for basic building blocks like the N-mixture model, so that we may use our growing data resources to design and fit more complex models of populations over time and space.

This chapter will summarize recent AI advances that provide the first exact and efficient algorithms for models with latent count variables. The key idea is to express probability distributions over count variables using *probability generating functions* (PGFs), and then to implement traditional inference algorithms using this novel representation. PGFs



$$n \sim \text{Poisson}(\lambda)$$

$$y_k \mid n \sim \text{Binomial}(n, p)$$

Figure 1.1 The N-mixture model [Royle, 2004]. The latent variable n represents the unknown number of animals in a habitat patch. The observed (shaded) variables $y_1, \ldots, y_K$ represent the number detected by an observer during repeated surveys of the habitat patch. During each survey, each animal is detected with probability p.

allow us to compactly represent distributions of interest even though they have an infinite number of possible values. The resulting algorithms are exact, substantially faster than existing approximate approaches, and they avoid misleading statistical inferences caused by *a priori* upper bounds on the population size.

The goal of the chapter is to present an overview of the main ideas and illustrate their impact on ecological models. The readers are referred to the papers by Winner and Sheldon [2016] and Winner et al. [2017] for additional details.

1.2 Population Models and Estimation

We will introduce the ecological and computational problem in more depth using Royle's N-mixture model. Recall that n is the unobserved population size, and $y_1, \ldots, y_K$ are the counts conducted by the observer. We encode uncertainty about these values, as well as some beliefs about the mechanisms that generated them, through a probabilistic model. The model is shown in Figure 1.1. The variables $y_1, \ldots, y_K$ are shaded to indicate that they are observed. The arrows represent the dependencies of the model: the observed counts depend on the unobserved population size. The population size n is assumed to be a Poisson random variable with (unknown) mean λ. The Poisson distribution is selected as a canonical distribution for count variables; our methods are not tied to this particular choice nor do we make any particular mechanistic interpretation of it. For the observations, we assume the observer detects each animal independently with probability p, so that y_k is distributed according to the Binomial distribution with n trials (the number of animals) and success probability p (the probability of detecting each one).

The scientist wishes to answer questions like: "How many animals were present?" or "What is the probability of detecting an animal that is present?" Often, the model will be simultaneously applied to many different habitat patches, where the mean number of animals λ is either shared across patches, or modeled as $\lambda = f(x)$, where x is a vector of covariates that describe the habitat and other features of the patch. In this case the probability distribution over n is used to model variability among patches. The scientist can then answer questions such as: "What is the typical population size of patches in my district?" or "How does population size relate to measures of habitat quality?". For examples, see [Royle, 2004]. We will focus on the single-patch model because the computational considerations are the same across all of these modeling variations.

A key to answering each of the above questions is *probabilistic inference* in the model: answering queries about the probability of some variables in the model given some other variables that are observed. Suppose the observer visits the patch three times and observes $y_1 = 2, y_2 = 5, y_3 = 3$ (these are the number of animals detected in each visit). If $\lambda = 20$ and $p = 0.25$, the probability of observing these values is 0.0034; if $\lambda = 10$ and $p = 0.25$, the probability is 0.0025. Based on this, we believe the first setting of parameters is more likely. The principle of maximum likelihood is to set the unknown model parameters to the ones that maximize the probability of the observed variables.

So, we can see that calculating the likelihood $p(y_1, \ldots y_K) := p(y_{1:K})$—the probability of all observed values—is a key computational problem. Solving this problem will allow us to use numerical optimization routines to find the parameters λ and p that maximize the likelihood, and it is also a basic building block of *posterior queries* about the model, such as "What is the probability there were 5 animals in the patch given my observations?".

So, let us focus on the problem of computing the likelihood $p(y_{1:k})$ for fixed λ and p. Since our model was specified in terms of the joint probability of *all* the variables, we must apply the rules of probability to sum over all possible values of n:

$$\text{Likelihood :} \qquad p(y_{1:K}) = \sum_{n=0}^{\infty} p(n, y_{1:K})$$

Each term of the sum on the right-hand side is easily computed from the model specification. The nth term is the joint probability $p(n, y_{1:K})$, which, according to the model we have defined is equal to $p(n) \prod_{k=1}^{K} p(y_k \mid n)$, where $p(n) = \frac{\lambda^n e^{-\lambda}}{n!}$ is Poisson prior probability that there are n animals present, and $p(y_k \mid n) = \binom{n}{y_k} p^{y_k} (1-p)^{n-y_k}$ is the probability that y_k animals are observed on the kth visit given that there are actually n animals present. However, even though we can compute each term easily, we can't compute the likelihood directly because there are an infinite number of terms! More generally, we lack general computational tools to efficiently manipulate distributions over an infinite number of terms, and this prevents us from applying well known probabilistic inference algorithms.

1.3 A Change of Representation: Probability Generating Functions

The main idea of our approach is to work instead using a different representation of the probability distribution: a *probability generating function*, or PGF. The PGF is a transformation of a probability distribution $q(n)$ over non-negative integers defined as follows:

$$\underbrace{\{q(n) : n = 0, 1, 2, \ldots\}}_{\text{Probability distribution}} \qquad \Longrightarrow \qquad \underbrace{F(s) = \textstyle\sum_{n=0}^{\infty} q(n) s^n}_{\text{Probability generating function}}$$

The PGF uses the probability values as coefficients of a power series (i.e., a polynomial with an infinite number of terms) in the new variable s, and it is a function that maps s to another real number whenever the series converges. It is well known that this transformation preserves all the information about the probability distribution. That is, if we know $F(s)$, we can recover all of the original probability values.[1] At

[1] Specifically, we do this using the derivatives of F at zero: $q(n) = F^{(n)}(0)/n!$

first glance, it is not clear what this buys us. We have switched from an infinite sequence of probability values to a power series with an infinite number of terms. The important observation is that it may be possible to find a *compact* representation of the probability generating function. For example, for the Poisson distribution, for any value of s, the infinite sum on the right-hand side converges and we have:

$$\left\{p(n) = \frac{\lambda^n e^{-\lambda}}{n!} : n = 0, 1, 2, \ldots\right\} \implies F(s) = \sum_{n=0}^{\infty} \frac{\lambda^n e^{-\lambda}}{n!} s^n = e^{\lambda(s-1)}$$

This is now a simple, compact representation of the entire probability distribution. It should not be a surprise that we can do this for the Poisson distribution, since we already had a compact formula for the probability values.

We will show that it is possible to compute compact representations of PGFs for probability distributions that arise during inference algorithms. Returning to the previous N-mixture model example, consider the distribution $p(n, y_1 = 2, y_2 = 5, y_3 = 3)$. We will describe how to algorithmically compute a formula for the PGF, which in this example is:

$$\begin{aligned}F(s) =& \sum_{n=0}^{\infty} p(n, y_1 = 2, y_2 = 5, y_3 = 3)s^n \\ =&\big(0.0061s^5 + 0.1034s^6 + 0.5126s^7 \\ &+ 1.0000s^8 + 0.8023s^9 + 0.2184s^{10}\big) \\ &\times \exp(8.4375s - 15.4101)\end{aligned} \tag{1.1}$$

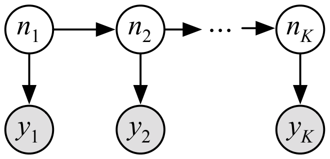
Although this expression appears somewhat complex — it is a polynomial of degree ten times an exponential function — it is a tractable and exact representation of the distribution $p(n, y_{1:K})$.

Importantly, given the PGF it is easy to solve our original problem of computing the likelihood — we simply evaluate the PGF at $s = 1$. From the series representation, we know that $F(1) = \sum_{n=0}^{\infty} p(n, y_{1:K})1^n$ is the sum over all terms in the series, which is equal to the likelihood $p(y_{1:K})$. We can compute $F(1)$ efficiently by substituting $s = 1$ in the compact representation. For example, we find that $p(y_1 = 2, y_2 = 5, y_3 = 3) = 0.0025$ by substituting $s = 1$ in the right-hand side of Equation (1.1).

We have now seen the main elements of our new approach for probabilistic inference. Given a probability model, we will devise an algorithm to compute a compact representation of the PGF for the distribution $p(n, y_{1:K})$ where n is a single latent variable and $y_{1:K}$ are all of the observed variables. Then we will compute the likelihood by evaluating $F(1)$ using our compact representation. What remains is to describe the mathematical and computational operations needed to find the compact representation of the probability generating function for models of interest. We summarize these steps in the following sections.

1.4 The PGF Forward Algorithm

Our goal is to algorithmically manipulate PGFs to compute the likelihood of population models. We would like to do this for a reasonably broad class of models that includes the N-mixture model and other models that are used in practice by ecologists. To this end, we will describe a class of models called *integer hidden Markov models (HMMs)* for partially observed populations that change over time through processes such as immigration, mortality, and reproduction. Integer HMMs map closely onto *open metapopulation models* from statistical population ecology [Dail and Madsen, 2011] and (latent) *branching processes* from applied mathematics and epidemiology [Watson and Galton, 1875, Heathcote, 1965].



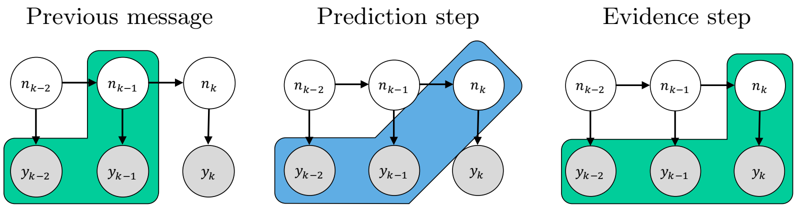
$$n_k = \sum_{i=1}^{n_{k-1}} z_{k,i} + m_k$$

$$y_k \sim \text{Binomial}(n_k, p)$$

Figure 1.2 The integer HMM model. The variable n_k represents the size of the population in the kth time period. The variable y_k represents the number of individuals counted by an observer during that time period. The population changes over time through immigration and emigration, mortality, and reproduction.

The Integer HMM Figure 1.2 illustrates the model. This extends the N-mixture model so the number of animals in the patch can change over time. The variable n_k is the size of the population at time k, which, again, is not observed. The variable y_k is the number detected by the observer. As before, we assume that each individual is detected with with probability p. The population size n_k now depends probabilistically on the population size from the previous time step as follows. First, for all i, the ith individual from the previous time step contributes $z_{k,i}$ individuals to the present time step, where $z_{k,i} \sim P_Z$ is an independent random variable drawn from the common *offspring distribution* P_Z. Here, the "offspring" can include the individual itself, to model the event that it survives from one generation to the next and remains in the patch; they can also include true offspring, to model reproduction. In this way, the modeler can model emigration, mortality, and reproduction by the appropriate choice of offspring distribution. In practice, the modeler would model these processes separately and then follow standard procedures to determine the offspring distribution P_Z. We assume only that the PGF $F(s)$ of the offspring distribution is available. In addition to the "offspring" from previous time steps, m_k new individuals enter the population, where m_k is a random variable from the *immigration distribution.* The modeler is free to select any count-valued immigration distribution; we assume only that that the PGF $G(s)$ is specified. The model can easily be extended to allow these distributions to vary over time.

The Forward Algorithm The forward algorithm is a classical algorithm to compute the likelihood in a hidden Markov model [Rabiner, 1989]. We will adapt it to use PGFs in its internal representation to compute the likelihood in integer HMMs. The algorithm is illustrated schematically in Figure 1.3. It proceeds in steps that model the joint distributions of different subsets of the variables. In the figure, the shaded boxes indicate which variables are modeled at each step. The fundamental distributions of interest are those of the form $p(n_k, y_{1:k})$—of the hidden variable n_k and the observations $y_{1:k}$ up to and including the



$$\underbrace{p(n_{k-1}, y_{1:k-1})}_{\alpha_{k-1}(n_{k-1})} \qquad \underbrace{p(n_k, y_{1:k-1})}_{\gamma_k(n_k)} \qquad \underbrace{p(n_k, y_{1:k})}_{\alpha_k(n_k)}$$

Figure 1.3 Illustration of the forward algorithm. The algorithm utilizes a recurrence to compute the current message $\alpha_k(n_k)$ (right) starting from the message $\alpha_{k-1}(n_{k-1})$ at the previous time step (left). Each message represents the joint distribution of a subset of variables that includes one hidden variable and a prefix of the observed variables—the shaded boxes show the subset corresponding to each message. The recurrence utilizes an intermediate prediction step that creates the message $\gamma_k(n_k)$ (middle).

corresponding time step—which we denote as $\alpha_k(n_k)$ and are called the *messages*. The left-most figure shows the message $\alpha_{k-1}(n_{k-1})$ for the $k-1$st time step, and the right-most plot shows the message $\alpha_k(n_k)$ for the kth time step. A recurrence is used to compute the message for the current time step from the previous one. We have split the recurrence into two steps: (1) the *prediction* step, where the observations up until time $k-1$ are used to predict n_k, resulting in the intermediate quantity $\gamma_k(n_k) := p(n_k, y_{1:k-1})$ (illustrated in the middle figure), and (2) the *evidence* step, where the observation at time k is used to update the distribution and obtain $\alpha_k(n_k)$. The two steps are defined formally as follows:

$$\text{Predict}: \qquad \gamma_k(n_k) = \sum_{n_{k-1}} \alpha_{k-1}(n_{k-1}) p(n_k \mid n_{k-1}),$$

$$\text{Evidence}: \qquad \alpha_k(n_k) = \gamma_k(n_k) p(y_k \mid n_k).$$

Starting from a base case, all α messages can be computed in a single forward pass using this recurrence. The likelihood is obtained by summing over all values of the final message.

The PGF Forward Algorithm The forward algorithm recurrence is mathematically correct even for integer HMMs, but the limits of the summation in the prediction step are infinite, and there are an infinite number of terms in each message, so it cannot be implemented directly. Instead, we will modify the algorithm to work with the PGFs of the α and γ messages, which are defined (using the corresponding capital letters) as $A_k(s_k) = \sum_{n_k=0}^{\infty} \alpha_k(n_k) s_k^{n_k}$ and $\Gamma_k(u_k) = \sum_{n_k=0}^{\infty} \gamma_k(n_k) u_k^{n_k}$. An equivalent recurrence for the PGFs is derived in [Winner et al., 2017]:

$$\text{Predict}: \qquad \Gamma_k(u_k) = A_{k-1}\big(F(u_k)\big) \cdot G(u_k)$$

$$\text{Evidence}: \qquad A_k(s_k) = \frac{(s_k \rho_k)^{y_k}}{y_k!} \cdot \Gamma_k^{(y_k)}\big(s_k(1-\rho_k)\big)$$

We do not provide details of how these formulas are derived. The formula in the prediction step follows from the model definition (see Figure 1.2) by fairly standard and elementary manipulations of PGFs, and is well known in the literature on branching processes [Heathcote, 1965]. The formula in the evidence step may appear surprising. It includes the y_kth derivative of the function Γ_k from the prediction step. This formula was derived in [Winner and Sheldon, 2016]. The derivatives are related to the selection of particular terms in the joint PGF of n_k and y_k corresponding to the observed value of y_k.

The likelihood is recovered by evaluating the final PGF at the input value one:

$$\text{Likelihood:} \qquad p(y_{1:K}) = A_K(1).$$

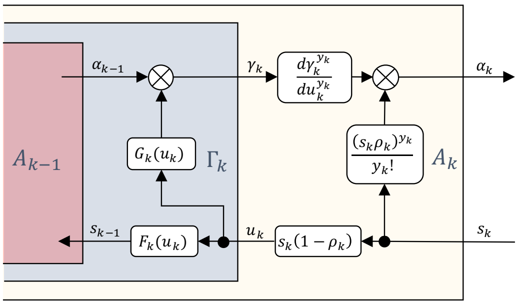
It remains to discuss how to efficiently implement the PGF recurrence.

1.5 Implementing the Recurrence: Computation with PGFs

We provide a high-level overview of the techniques to algorithmically manipulate PGFs.

$$\boxed{\begin{aligned} F(s) =&\big(0.0061s^5 + 0.1034s^6 + 0.5126s^7 \\ &+ 1.0000s^8 + 0.8023s^9 + 0.2184s^{10}\big) \\ &\times \exp(8.4375s - 15.4101) \end{aligned}}$$

(a) Symbolic representation



(b) Circuit representation

Figure 1.4 Algorithmic manipulation of PGFs. Winner and Sheldon [2016] showed how to compute compact symbolic representations of PGFs that appear in the forward algorithm for a restricted class of models with Poisson latent variables. Winner et al. [2017] showed how to use a circuit representation and automatic differentiation to evaluate the final PGF to compute the likelihood.

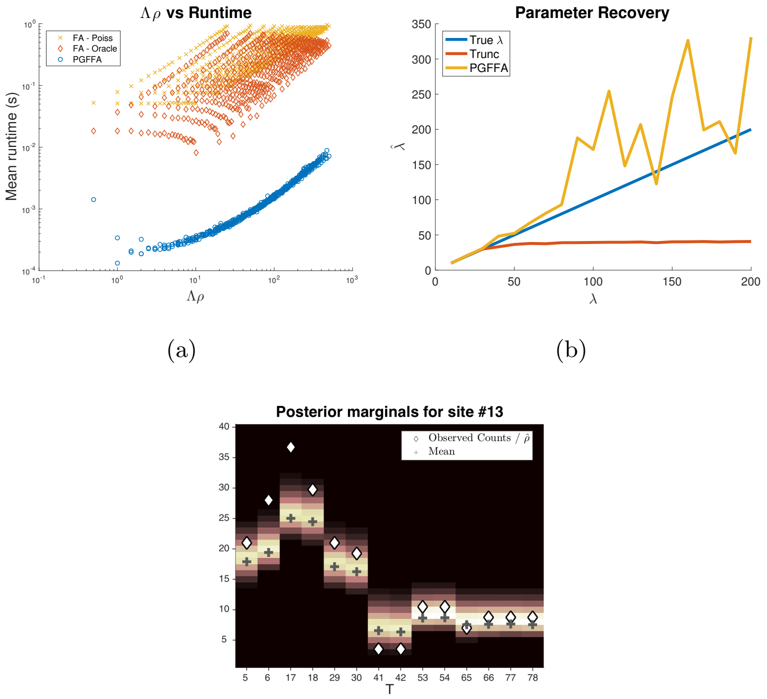
Symbolic manipulation of PGFs An obvious approach to try first is to write down the mathematical formula for the first PGF, $A_1(s_1)$, which will always have a simple form, and then observe how this formula changes when the prediction and evidence steps are repeatedly applied. In the best case, one will be able to simplify each successive PGF into a tractable mathematical expression. In [Winner and Sheldon, 2016], we successfully followed this *symbolic* approach for a restricted class of models called *Poisson HMMs*. In Poisson HMMs, the immigration distribution is Poisson and the offspring distribution is Bernoulli, which models survival but is not able to model reproduction. In this case we showed that each PGF has a form similar to the one shown in Figure 1.4(a). Specifically, each PGF can be written in the form $f(s)\exp(as+b)$ where f is a bounded degree polynomial. Thus, it can be represented compactly by the polynomial coefficients and the scalars a and b, and these can be computed efficiently from the representation of the PGF in the previous time step.

Circuit representation Although the symbolic representation is efficient, it does not seem to extend to a broader class of models, including variations commonly used by ecologists [Dail and Madsen, 2011]. For example, it is common to model immigration using the negative binomial distribution instead of the Poisson. In this and other cases, the mathematical expressions for PGFs grow rapidly more complex in each iteration and do not appear to simplify to a tractable form. In [Winner et al., 2017], we developed a much more general approach that does not attempt to represent PGFs symbolically, but instead models them using a circuit or *computation graph.* Since each PGF recursively calls the previous one in the recurrence, this circuit consists of recursively nested circuits; see Figure 1.4(b). Because the recurrence involves *derivatives* of prior PGFs, the circuit cannot be evaluated using simple arithmetic operations alone. In [Winner et al., 2017] we developed novel techniques based on automatic differentiation [Griewank and Walther, 2008] to compute the nested, higher-order, derivatives required to evaluate $A_K(1)$.

1.6 Demonstration and Experiments

So far we have given an overview of the PGF forward algorithm, a novel algorithm that leads to the first exact inference algorithms for ecological models with latent count variables. In this section, we will examine the practical capabilities of the algorithm by comparing it to the previously available approximate algorithms, and through two case studies.

Running time Our new exact algorithms are substantially faster than existing *approximate* algorithms. Figure 1.5(a) shows the running time of the symbolic version of the PGF forward algorithm for Poisson HMMs [Win-



(a)

(b)

(c)

Figure 1.5 Experimental evaluation of the PGF forward algorithm: (a) Running time PGF forward (PGFFA) versus two versions of the truncated forward algorithm (FA - Oracle, FA - Poiss) for a Poisson HMM (see text), (b) Parameter recovery via maximum-likelihood using the PGF forward algorithm avoids a pathology that affects the corresponding procedure using the truncated forward algorithm, (c) Illustration of inferences from a fitted model for the Northern Dusky Salamander. The horizontal axis represents months since the beginning of the study period; surveys are conducted only in June and July so time steps are not consecutive. The vertical axis represents the number of individuals, with crosses showing the posterior mean, and shading intensity illustrating the posterior probability. The diamonds represent a coarse estimate made by dividng the observed count by the detection probability.

ner and Sheldon, 2016] compared with two versions of an approximate algorithm that is used currently in practice [Royle, 2004, Dail and Madsen, 2011, Fiske and Chandler, 2011]. The approximate algorithm places an *a priori* upper bound $N_{\max}$ on the population size, and then uses the standard forward algorithm [Rabiner, 1989] to compute the approximate likelihood. We refer to this as the *truncated* forward algorithm, denoted FA in the figure. Our algorithm is denoted PGFFA. The running time of PGFFA grows with the magnitude of the observed counts. The running time of the truncated forward algorithm grows with the truncation parameter $N_{\max}$: smaller values are faster, but may underestimate the likelihood. Selecting $N_{\max}$ large enough to yield correct likelihoods but small enough to be fast is known to be difficult [Couturier et al., 2013, Dennis et al., 2015]. We evaluated two strategies to select $N_{\max}$: an *oracle* strategy that runs prior experiments to find an optimal setting of $N_{\max}$ ("FA – Oracle"), and a heuristic based on the prior Poisson distribution ("FA – Poiss"). We simulated data from a Poisson HMM parameterized based on a model for insect populations [Zonneveld, 1991], and then measured the time for each algorithm to compute the likelihood in this model. We varied two parameters over a range of different values: the parameter Λ controls the overall population size, and the parameter ρ is the detection probability. (For further details, see [Winner and Sheldon, 2016].) The running time is plotted relative to $\Lambda\rho$, which is the expected total number of individuals observed, on a log-log scale. We can see that the PGFFA running time indeed scales with the magnitude of the observations, and is 2–3 orders of magnitude faster than the truncated algorithms.

Avoiding Pathologies in Parameter Estimation The next experiment highlights a pathology of the truncated algorithm that is avoided by our exact algorithms. We simulated data from the N-mixture model and attempted to recover the parameter values λ (population size) and ρ (detection probability) by numerically maximizing the likelihood, using both the PGF forward algorithm and the truncated forward algorithm as subroutines to compute the likelihood. For the truncated algorithm, the modeler must select $N_{\max}$ without knowing the true values of the parameters—we assume she heuristically sets $N_{\max}$ to be approximately 5 times the average observed count based on her belief that the detection probability is not too small and this will capture most of the probability mass. We varied λ and ρ inversely proportionally to each other so that their product $\lambda\rho$, which is the expected number of observed animals, is held constant at 10. We therefore fixed $N_{\max} = 50$ to be five times this observed number.

Figure 1.5(b) shows that as the true λ approaches and surpasses $N_{\max} = 50$, the truncated method cuts off significant portions of the probability mass and severely underestimates λ. This artificially reinforces the modeler's prior belief that the true detection probability is "not too small", even when the true detection probability approaches zero! In contrast, estimation with the exact likelihood does not show this bias. It does show significantly increased variance as λ increases and $\rho \to 0$. In fact, this variance is a property of the true likelihood, but is artificially suppressed by the truncated algorithm. It is well-known in this and related models that, without enough data, it is difficult to tease apart the population size and detection probability, especially as the true detection probability goes to zero (e.g., see [Dennis et al., 2015]). The variance seen here is a byproduct of the fact that the parameters are *actually* poorly determined as $\rho \to 0$. The truncated algorithm artificially stabilizes the estimation procedure by expressing a hidden bias toward smaller population sizes.

Dusky Salamander Case Study Figure 1.5(c) shows the results from a case study to model the abundance of Northern Dusky Salamanders at

21 sites in the mid-Atlantic US using data from [Zipkin et al., 2014]. The data consists of 14 counts at each site, conducted in June and July over 7 years. Six sites were excluded because no salamanders were observed. We first fit a Poisson HMM by numerically maximizing the likelihood as computed by the PGF forward algorithm. Arrivals are modeled as a homogeneous Poisson process, and survival is modeled by assuming individual lifetimes are exponentially distributed. The fitted parameters indicated an arrival rate of 0.32 individuals per month, a mean lifetime of 14.25 months, and detection probability of 0.58.

We then investigate the posterior distribution over the number of animals at each time step. We may wish to use the model to make fine grained inferences about the population status at individual sites over time. In the figure, the horizontal axis represents time (in months) and the vertical axis is the population size. The cross represents the posterior mean for the population size at the given time step (given all observations), and the shading intensity represents the posterior probability of different values; the "spread" of this posterior distribution helps quantify our posterior uncertainty under the modeling assumptions we have made. The diamonds represent a coarse estimate of the population size at each time step made by dividing the observed count by the estimated detection probability. In contrast to the coarse estimates, the posterior distribution varies more smoothly over time, because it models the counts as being coupled through time by the processes of survival and immigration.

Computationally, querying the posterior distribution in this way requires computation of the *posterior marginals*, which are the distributions $p(n_k \mid y_{1:K})$ of each latent variable given *all* of the observed data (both preceding and following the focal time period). A variant of the PGF forward algorithm can also compute these marginals [Winner and Sheldon, 2016].

The Model Zoo A major advance of [Winner et al., 2017] was the ability to perform inference in a much wider class of models by using circuits and automatic differentiation to evaluate the PGFs. To demonstrate the advantages of this flexibility, we used this version of the algorithm within an optimization routine to compute maximum likelihood estimates (MLEs) for a variety of models with different immigration and offspring distributions. In each experiment, we generated a data set of independent samples from each model and then used a numerical optimization procedure to find the parameters that maximize the likelihood of the observations. We varied the immigration and offspring distribu-

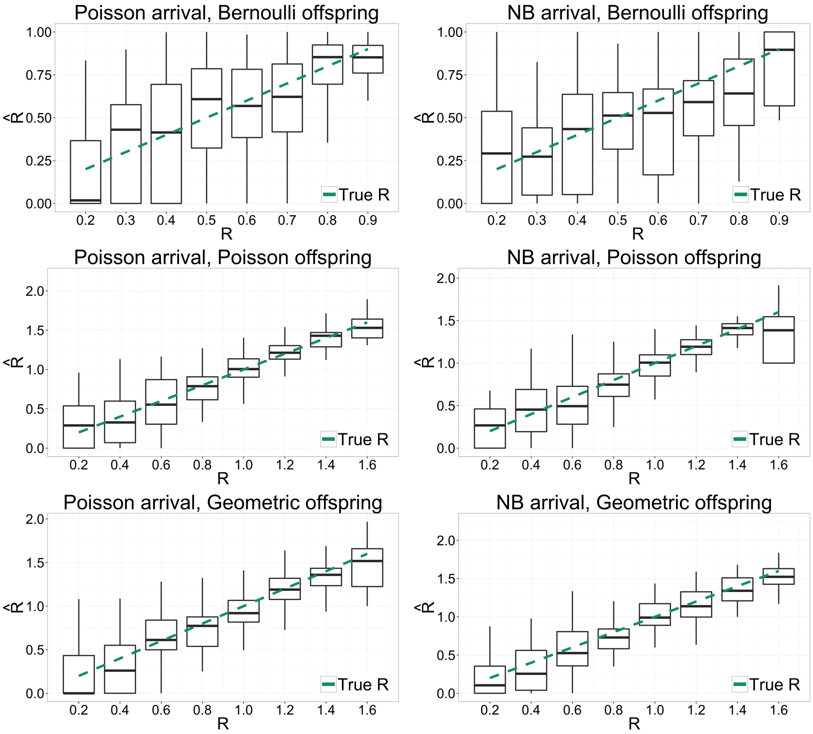


Figure 1.6 Accuracy of estimates for the mean R of the offspring distribution in different models. Rows, from top to bottom: Bernoulli offspring, Poisson offspring, geometric offspring. Columns, from left to right: Poisson immigration, negative binomial immigration. For each model combination and true value of R, box plots summarize the estimated values from 50 independent trials.

tions as well as the mean R of the offspring distribution. We fixed the mean of the immigration distribution to $\lambda = 6$ across all models, and the detection probability to $p = 0.6$. The quantity R is known as the "basic reproduction number", or the average number of offspring produced by a single individual, which is a key measure of population viability and hence important to estimate. Each panel in Figure 1.6 shows the distribution of the 50 maximum-likelihood estimates for R vs the true values of R for a different model. The estimated values closely track the true values. This shows that the PGF forward algorithm can be applied within likelihood maximization routines to successfully fit the parameters of a wide class of models.

1.7 Conclusion

Effective conservation requires effective assessment and monitoring of animal populations. Population sizes and demographic rates are commonly estimated by fitting probabilistic population models to observational data. However, these models expose a gap in our probabilistic inference toolkit. They often contain latent count variables to represent unknown population sizes, and, despite many years of research in the area of probabilistic inference, until recently there was no known algorithm to exactly compute the likelihood in models with latent count variables.

This chapter summarizes recent advances in the AI research area of probabilistic inference motivated by this gap. We described how the forward algorithm, a standard inference algorithm for hidden Markov models, can be adapted to use *probability generating functions* as its internal representation of probability distributions, which then leads to the first efficient and exact inference algorithms for this class of models.

We recommend several directions forward. From a technical standpoint, the forward algorithm is an example of a *message passing* inference algorithm [Pearl, 1986, Lauritzen and Spiegelhalter, 1988, Jensen et al., 1990, Shenoy and Shafer, 1990]. We have shown how to extend the forward algorithm to a broad class of models with latent count variables by using PGFs to represent messages. Extending this idea to more structurally complex models by doing the analogous thing for general-purpose message passing algorithms, and, more generally, exploring the potential uses of PGFs for probabilistic inference, is an interesting technical research direction. Developing techniques to compute the *gradient* of the log-likelihood in addition to the likelihood, which would facilitate learning and parameter estimation in ecological models, is another promising research direction.

From the application standpoint, we envision extending models such as the N-mixture model and the integer HMM, which assume independence across sites, by using them as the basic building blocks of spatio-temporal models that also model the interdependence among sites. Our hypothesis is that increasing volumes of data available from citizen science projects and other technological advances provide evidence about spatio-temporal patterns and interactions among populations. These models will have many more variables and interactions, and will present increasingly difficult challenges in the area of probabilistic inference. We recommend continued interactions among ecologists and AI researchers to design these models together with efficient algorithms to reason about them.

References

Brian L. Sullivan, Christopher L. Wood, Marshall J. Iliff, Rick E. Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282 – 2292, 2009.

J. A. Royle. N-Mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60(1):108–115, 2004.

K. Winner and D. Sheldon. Probabilistic inference with generating functions for Poisson latent variable models. In *Advances in Neural Information Processing Systems 29*, 2016.

Kevin Winner, Debora Sujono, and Daniel Sheldon. Exact inference for integer latent-variable models. In *Proc. of the 34th International Conference on Machine Learning (ICML)*, pages 3761–3770, 2017.

D. Dail and L. Madsen. Models for estimating abundance from repeated counts of an open metapopulation. *Biometrics*, 67(2):577–587, 2011.

H. W. Watson and F. Galton. On the probability of the extinction of families. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 4:138–144, 1875.

C. R. Heathcote. A branching process allowing immigration. *Journal of the Royal Statistical Society. Series B (Methodological)*, 27(1):138–143, 1965.

L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, feb 1989.

A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* SIAM, 2008.

I. J. Fiske and R. B. Chandler. unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43:1–23, 2011.

Thibaut Couturier, Marc Cheylan, Albert Bertolero, Guillelme Astruc, and Aurelien Besnard. Estimating abundance and population trends when detection is low and highly variable: A comparison of three methods for the Hermann's tortoise. *Journal of Wildlife Management*, 77(3):454–462, 2013. ISSN 0022541X. doi: 10.1002/jwmg.499.

Emily B. Dennis, Byron J.T. Morgan, and Martin S. Ridout. Computational aspects of n-mixture models. *Biometrics*, 71(1):237–

246, 2015. ISSN 1541-0420. doi: 10.1111/biom.12246. URL `http://dx.doi.org/10.1111/biom.12246`.

C. Zonneveld. Estimating death rates from transect counts. *Ecological Entomology*, 16(1):115–121, 1991.

E. F. Zipkin, J. T. Thorson, K. See, H. J. Lynch, E. H. C. Grant, Y. Kanno, R. B. Chandler, B. H. Letcher, and J. A. Royle. Modeling structured population dynamics using data from unmarked individuals. *Ecology*, 95 (1):22–29, 2014.

J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288, 1986.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational statistics quarterly*, 1990.

P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In *Uncertainty in Artificial Intelligence*, 1990.