Bayes Optimal Cardinality Filters for Streaming Count Data

R. L. Streit
Metron, 1818 Library St., Suite 600, Reston, VA 20190 USA
r.streit@ieee.org *and* streit@metsci.com

***Abstract*—A time sequence of counts of the number of sensor detections is the sum of the number of detections of objects and the number of false alarms. We model the count sequence as an insertion-deletion process, and model the time-varying number of objects as a birth-death process. Under these modeling assumptions, we derive the optimal recursive Bayesian posterior distribution for the number of objects conditioned only on the count sequence. The method is potentially applicable in management science to detect changes in demand in decision-independent observed data streams and in social media to estimate the number of users who abuse hashtags. A maximum a posteriori (MAP) algorithm for estimating the parameters of the birth-death and insertion-deletion processes is presented.**

Keywords— Cardinality filter, streaming count data, generating function, z-transform, lagged filter, Bayes optimal filter, saddlepoint approximation

I. Introduction

In multiple object tracking problems sensor measurement sets are typically finite sets of points in some Euclidean space, but in this paper we are concerned only with the *number* of points in these sets. We derive an optimal recursive Bayesian filter to estimate the number of objects, or signals, present in a data stream that comprises only the observed cardinalities of the sequence of sensor measurement sets.

The only input to cardinal filters, as we will call them, is a stream of nonnegative integers. Their output is the Bayes optimal posterior mass function (pmf) of the number of objects conditioned only on the input count stream.

One reason to consider such problems is that they altogether avoid using object motion models. The objects can move independently and arbitrarily along any trajectory whatsoever without altering the cardinal filter performance. This paper studies the price to be paid for object motion independence.

The problem may have applications in management science for decision-independent data streams that compress data to its cardinality [1] [2]. For example, if the streaming data is customer demand observed on a weekly basis, the inventory manager may seek only to detect changes in demand, in which case the data stream is decision-independent. (The data stream is decision-dependent if the manager makes decisions to increase demand.) In the context of social media, the number of abusive hashtag (#) users is analogous to the number of objects. For many such applications, it may be necessary to "fit" the parameters of the statistical model to the data. We propose a MAP (maximum *a posteriori*) parameter estimation algorithm.

The natural, intuitive approach to computing the pmf of the cardinal filter is to enumerate and compute the probabilities of all the ways that the numbers of objects and false alarms can account for the observed counts. This paper takes an alternative approach. It encodes the myriad combinations into a succinct, easily computed generating function (GF). The pmf of the *exact* cardinal filter is a normalized mixed derivative of the GF, where the order of the derivative is determined by the observed counts. This expression is both concise and elegant. The advantage of the GF approach is that principled methods are available to approximate the derivatives. We approximate the filter using the saddlepoint method [3], an established and widely used technique in physics and statistics.

The GF approach to cardinal filters was first proposed in [4]. Using a much improved notation, this paper shows that the joint GF can be used for Bayesian inference of any set of variables in the problem, not just object number. It uses the saddlepoint method to derive low computational complexity approximations to Bayes optimal estimators, with particular attention to cardinal filters. Several numerical examples are given. To make the cardinal filter adaptive, we propose an approximate MAP algorithm to estimate filter parameters from archived and streaming data.

Section II provides further background discussion and gives an overview of the paper and points out the significance of the saddlepoint approximate filters. Section III explicitly states the statistical modeling assumptions. Section IV gives the backward recursion for the GF of the general model. Similar recursions appear in population processes, e.g., the famous Galton-Watson process [12, Chap. 1]. Numerical examples of the exact cardinal filter are given in Section V. Section VI gives the saddlepoint optimal cardinal filter, together with examples. The approximate filter is useful when the count data are large and/or the lag time is long. Filter parameter estimation is discussed in Section VII. This is important for adaptive filters that estimate the filter parameters (e.g., false alarm rates) from the available count data. Section VIII gives directions for future work. The Appendix discusses the saddlepoint approximation.

II. Overview

The length of the sequence is $K \geq 1$. The number of objects at time k is denoted by N_k, where $1 \leq k \leq K$. The current time is K. The *a priori* number of objects is denoted N_0. The number of objects may vary over time due to two random effects: existing objects may fail to transition (i.e., terminate) from one time to the next, and new objects may be born at each time. The number of objects born at time k is denoted by B_k. The number of objects N_k is equal to the sum of the number of objects that successfully transition from the previous time and the number of newborn objects at time k.

An object (including the newly born) may or may not be detected at time k. Object detection processes are independent of each other. If detected, an object generates a random number of measurements. In most problems each object generates at most one measurement, but this restriction is unnecessary. The number of measurements M_k is the sum of the number of measurements generated by detected objects and the random number of false alarms, denoted by F_k. False alarms are spurious measurements that are independent of the number of objects. We derive a Bayes optimal cardinal filter for object number N_k conditioned on the observed measurement counts, $M_k = m_k$.

The pmf of the sum of independent random numbers is the convolution of their pmfs; equivalently, the GF of the sum is the product of the GFs[1] We use this well-known property to show that

[1] In signal processing, the GF is called the z-transform and takes values in a complex-valued frequency domain.

GFs are powerful analytical tools for inference networks. We use them in the next section to derive a backward recursion for the (exact) joint GF of the random integers $\mathcal{V}$, where $\mathcal{V}$ is the set of the $4K+1$ integers N_k, M_k, B_k, and F_k. The joint GF is a low complexity calculation.

Let $\mathcal{I} \cup \mathcal{J} \cup \mathcal{K}$ be a pairwise disjoint partition of $\mathcal{V}$. It is remarkable that the joint GF enables us to write the (exact) Bayes optimal estimator for the variables in $\mathcal{I}$ conditioned on observations of the variables in $\mathcal{J}$ as the normalized mixed derivative of the joint GF marginalized over the indeterminates of the variables in $\mathcal{K}$ [5].

The Bayes optimal estimators of interest in this paper are the cardinality filters, for which $\mathcal{I} = \{N_k\}_1^K$, $\mathcal{J} = \{M_k\}_1^K$, and $\mathcal{K} = \{N_0\} \cup \{B_k, F_k\}_1^K$. Marginalization is easy since it involves setting the indeterminates corresponding to the random variables in $\mathcal{K}$ equal to one. The difficulty is computing the mixed derivatives.

The cost of evaluating the mixed derivatives can be significant. When exact derivative calculations are impractical, we replace them by their saddlepoint approximations [3], [10] Numerical experiments (not given here for lack of space) show that the relative approximation error is often small enough to be acceptable in many practical problems; however, a careful theoretical error analysis is difficult.

The practical value of saddlepoint approximations (cf. Eqn. (48) below) is that the computational complexity depends primarily on the number of conditioning variables (i.e., the number of variables in the set $\mathcal{J}$) and is largely independent of their numerical values.

The joint GF involves several numerical parameters (e.g., P_k^D in Eqn. (14) below). Some may be poorly known and others may drift slowly over time. MAP estimates of the parameters are derived from the likelihood function of the data. As shown in Section VII, the exact likelihood function is the (mixed) derivative of the appropriately marginalized joint GF. This derivative is a high-complexity calculation, so we replace it by its low-complexity saddle point approximation. The saddlepoint approximation to the likelihood is an explicit analytic function of the parameters, and the gradient with respect to the parameters is computed to machine accuracy by the complex step method [11]. Saddlepoint MAP estimates are found by gradient ascent.

III. Notation and Model Assumptions

The sequences of nonnegative integer random variables are

$$\begin{aligned} \mathbf{N} \equiv \mathbf{N}_{0:K} &= (N_0, \ldots, N_K) \\ \mathbf{M} \equiv \mathbf{M}_{1:K} &= (M_1, \ldots, M_K) \\ \mathbf{B} \equiv \mathbf{B}_{1:K} &= (B_1, \ldots, B_K) \\ \mathbf{F} \equiv \mathbf{F}_{1:K} &= (F_1, \ldots, F_K). \end{aligned} \tag{1}$$

The conditional independence relationships between them are depicted in Fig. 1. Realizations are denoted

$$\begin{aligned} \mathbf{n} &= (n_0, \ldots, n_K) \in \mathbb{Z}_+^{K+1} \\ \mathbf{m} &= (m_1, \ldots, m_K) \in \mathbb{Z}_+^K \\ \mathbf{b} &= (b_1, \ldots, b_K) \in \mathbb{Z}_+^K \\ \mathbf{f} &= (f_1, \ldots, f_K) \in \mathbb{Z}_+^K, \end{aligned} \tag{2}$$

where $\mathbb{Z}_+$ is the set of nonnegative integers. The joint probability mass function (pmf) is denoted

$$p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f}) \equiv \Pr\big\{\mathbf{N} = \mathbf{n}, \mathbf{M} = \mathbf{m}, \mathbf{B} = \mathbf{b}, \mathbf{F} = \mathbf{f}\big\}.$$

Let $p(n_0) = \Pr\{N_0 = n_0\}$ and, for $k = 1, \ldots, K$,

$$\begin{aligned} p(n_k) &= \Pr\{N_k = n_k\} \\ p(m_k) &= \Pr\{M_k = m_k\} \\ p(b_k) &= \Pr\{B_k = b_k\} \\ p(f_k) &= \Pr\{F_k = f_k\}. \end{aligned} \tag{3}$$

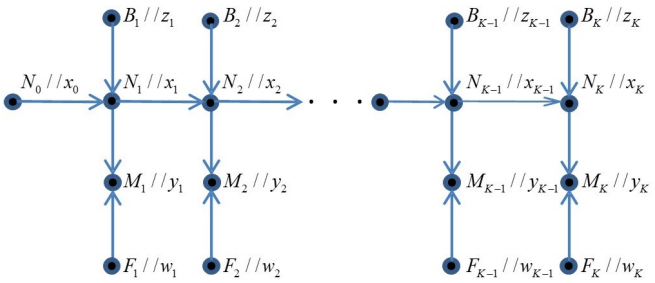

Fig. 1. Dependence network for the sequence of nonnegative random integers N_k, M_k, B_k, F_k and their associated indeterminate variables x_k, y_k, z_k, w_k; for example, the notation $M_2 // y_2$ indicates that y_2 corresponds to M_2.

Under the conditional independence assumptions depicted in Fig. 1, the joint pmf is the product

$$\begin{aligned} &p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f}) \\ &\quad = p(n_0) \prod_{k=1}^K p(b_k) p(f_k) p(n_k | n_{k-1}, b_k) p(m_k | n_k, f_k). \end{aligned} \tag{4}$$

The posterior pmf of the sequence of object counts $\mathbf{n}$ conditioned on the measurement sequence $\mathbf{m}$ is, by definition,

$$p(\mathbf{n}|\mathbf{m}) = \kappa \sum_{\mathbf{b},\mathbf{f}} p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f}), \tag{5}$$

where $\kappa^{-1} = \sum_{\mathbf{n},\mathbf{b},\mathbf{f}} p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f})$ is the normalization constant. Computing the numerator of this sum directly is strongly reminiscent of calculations in multi-hypothesis tracking (MHT) algorithms in that it requires enumerating – and evaluating the probabilities of – all feasible combinations of measurements to objects and false alarms. Direct computation is prohibitive except for small problems.

The GF approach used in this paper completely avoids direct enumeration. The steps in our method are outlined as follows:

1) Define the multivariate GF of the joint pmf $p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f})$.
2) Evaluate the GF by backward recursion.
3) Write the pmf of the cardinal filter as a derivative of the GF.
4) Approximate the pmf by applying the saddlepoint method to approximate the derivative.

The steps are exact up to the approximation in step 4). The saddlepoint approximation is an established technique in physics.

In the remainder of this section we develop the models that are used in the next section to complete step 2). Assign the indeterminate variables

$$\begin{aligned} \mathbf{x} &= (x_0, \ldots, x_K) \in \mathbb{C}^{K+1} \\ \mathbf{y} &= (y_1, \ldots, y_K) \in \mathbb{C}^K \\ \mathbf{z} &= (z_1, \ldots, z_K) \in \mathbb{C}^K \\ \mathbf{w} &= (w_1, \ldots, w_K) \in \mathbb{C}^K \end{aligned} \tag{6}$$

to the random variables $\mathbf{n}$, $\mathbf{m}$, $\mathbf{b}$, and $\mathbf{f}$, respectively. Define the shorthand notation

$$\begin{aligned} \mathbf{x}^{\mathbf{n}} &= x_0^{n_0} x_1^{n_1} \cdots x_K^{n_K} \\ \mathbf{y}^{\mathbf{m}} &= y_1^{m_1} \cdots y_K^{m_K} \\ \mathbf{z}^{\mathbf{b}} &= z_1^{b_1} \cdots z_K^{b_K} \\ \mathbf{w}^{\mathbf{f}} &= w_1^{f_1} \cdots w_K^{f_K}. \end{aligned} \tag{7}$$

By definition, the joint GF of $p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f})$ is, the formidable looking expression

$$\Psi_K(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \sum_{\mathbf{n},\mathbf{m},\mathbf{b},\mathbf{f}} p(\mathbf{n}, \mathbf{m}, \mathbf{b}, \mathbf{f})\, \mathbf{x}^{\mathbf{n}} \mathbf{y}^{\mathbf{m}} \mathbf{z}^{\mathbf{b}} \mathbf{w}^{\mathbf{f}}. \tag{8}$$

The sums on all $4K+1$ indices range from 0 to ∞. This multivariate power series converges when the magnitude of every indeterminate variable is less than or equal to 1, a result that follows from the fact that the GF is that of a probability distribution. (The GF is treated as a formal power series outside this "polydisc.") It is computed via the backward recursion given below in Section IV. The recursion is written in terms of the (univariate) GFs discussed in this section.

The *a priori* number of objects (i.e., the number at time 0) is an arbitrarily specified random variable. For example, if it is Poisson distributed, its GF is

$$G_{N_0}(x_0) = \sum_{n_0=0}^{\infty} p(n_0) x_0^{n_0} = e^{-\lambda_0^N + \lambda_0^N x_0}, \tag{9}$$

where $\lambda_0^N \geq 0$ is the expected number of objects. The *a priori* numbers of false alarms and new objects are arbitrarily specified random variables. If they are Poisson distributed, their GFs are

$$G_{B_k}(z_k) = \sum_{b_k=0}^{\infty} p(b_k) z_k^{b_k} = e^{-\lambda_k^B + \lambda_k^B z_k}, \quad 1 \leq k \leq K, \tag{10}$$

$$G_{F_k}(w_k) = \sum_{f_k=0}^{\infty} p(f_k) w_k^{f_k} = e^{-\lambda_k^F + \lambda_k^F w_k}, \quad 1 \leq k \leq K, \tag{11}$$

where $\lambda_k^B \geq 0$ and $\lambda_k^F \geq 0$. If at most one object is born at any given time, then newly born objects are Bernoulli with GF

$$G_{B_k}(z_k) = 1 - P_k^B + P_k^B z_k, \tag{12}$$

where $P_k^B \geq 0$ is the probability that an object is born at time k.

The number of objects N_k at time k is the sum of the number of objects N_{k-1} at time $k-1$ and the number of newly born objects B_k at time k. For $N_{k-1} = n_{k-1}$ and $B_k = b_k$, the GF is defined by

$$G_{N_k | n_{k-1} b_k}(x_k) = \sum_{n_k=0}^{\infty} p(n_k \,|\, n_{k-1}, b_k)\, x_k^{n_k}, \; 1 \leq k \leq K. \tag{13}$$

We derive an explicit expression for this GF from first principles. Objects may or may not successfully transition from time $k-1$ to time k. We assume that successful transitions occur with probability $P_k^S \geq 0$, that this probability is the same for all objects, and that objects transition independently of each other. We also assume that a successful transition results in exactly one object (i.e., objects do not split). With these assumptions, the GF of transition success is

$$G_{S_k}(x_k) = 1 - P_k^S + P_k^S x_k, \qquad 1 \leq k \leq K. \tag{14}$$

The number of objects that transition from time $k-1$ to time k is conditioned on N_{k-1}. Objects transition independently, so if $N_{k-1} = n_{k-1}$, the GF of the number of successful transitions is $\big(G_{S_k}(x_k)\big)^{n_{k-1}}$. The number of objects N_k is the sum of the number of objects that transition from time $k-1$ and the number that are newly born at time k. They are conditionally independent, so

$$G_{N_k | n_{k-1} b_k}(x_k) = \big(G_{S_k}(x_k)\big)^{n_{k-1}} x_k^{b_k}, \qquad 1 \leq k \leq K, \tag{15}$$

$$= \Big(1 - P_k^S + P_k^S x_k\Big)^{n_{k-1}} x_k^{b_k} \tag{16}$$

is the GF of N_k conditioned on $N_{k-1} = n_{k-1}$ and $B_k = b_k$.

Similarly, the number of measurements M_k is a random variable conditional on $N_k = n_k$ and $F_k = f_k$. Its GF is

$$G_{M_k | n_k f_k}(y_k) = \sum_{m_k=0}^{\infty} p(m_k \,|\, n_k, f_k)\, y_k^{m_k}, \quad 1 \leq k \leq K. \tag{17}$$

We assume that an object is detected with probability $P_k^D \geq 0$, that this probability is the same for all objects, and that object detections are independent of each other. For concreteness, we assume that detected objects generate exactly one measurement. Thus, the number of measurements generated by one object at time k is Bernoulli distributed, and its GF is

$$G_{D_k}(y_k) = 1 - P_k^D + P_k^D y_k, \qquad 1 \leq k \leq K. \tag{18}$$

The number of measurements M_k is the sum of the number of measurements generated by objects and the number of false alarms. Assuming that object detections and false alarms are independent, so

$$G_{M_k | n_k f_k}(y_k) = \big(G_{D_k}(y_k)\big)^{n_k} y_k^{f_k}, \qquad 1 \leq k \leq K, \tag{19}$$

$$= \Big(1 - P_k^D + P_k^D y_k\Big)^{n_k} y_k^{f_k} \tag{20}$$

is the GF of M_k conditioned on $N_k = n_k$ and $F_k = f_k$.

The expressions (15) and (19) hold in the general setting in which objects may, when transitioning, split into a random number of "parts," each of which counts as an object and each of which may generate a random number of measurements. In this general setting, $G_{S_k}(x_k)$ is the GF of the number of objects at time k, and $G_{D_k}(y_k)$ is the GF of the number of measurements produced by an object. The models (14) and (18) are special cases.

IV. General Recursion for the Joint GF

Nesting the sums in the definition (8) gives

$$\Psi_K(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \sum_{n_0} \left\{ \sum_{b_1, n_1, f_1, m_1} \cdots \left\{ \sum_{b_K, n_K, f_K, m_K} (\,\cdot\,) \right\} \cdots \right\}.$$

A low computational complexity backward recursion for this nested sum is derived in [4] in a different notation. For clarity, we derive the GF for the "no lag" special case $K = 1$ in the current notation in Subsection IV-A. This special case is also sufficient to capture the spirit of the general derivation. We give the GFs for lagged cardinal filters with lags 1 and 2 in Subsections IV-B and IV-C. Further details are omitted for lack of space.

BACKWARD RECURSION FOR $\Psi_K(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})$.

Let $\zeta_{K+1} = \eta_{K+1} = \xi_{K+1} = 1$

FOR $k = K, 1, -1$,

GF of false alarms at time k:

$\zeta_k = \zeta_{k+1} G_{F_k}\big(y_k w_k\big)$

GF of the trajectories of all objects born at time k:

$\eta_k = \eta_{k+1} G_{B_k}\big(z_k x_k G_{D_k}(y_k)\, \xi_{k+1}\big)$

GF of the trajectory of a preexisting object from k to K:

$\xi_k = G_{S_k}\big(x_k G_{D_k}(y_k)\, \xi_{k+1}\big)$

END FOR

$\Psi_K^{\mathbb{F}}(\mathbf{y}, \mathbf{z}) = \zeta_1$

$\Psi_K^{\mathbb{B}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \eta_1$

$\Psi_K^{\mathbb{O}}(\mathbf{x}, \mathbf{y}) = G_{N_0}(x_0\, \xi_1)$

$\Psi_K(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \Psi_K^{\mathbb{F}}(\mathbf{y}, \mathbf{w}) \Psi_K^{\mathbb{B}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \Psi_K^{\mathbb{O}}(\mathbf{x}, \mathbf{y})$.

The joint GF is a product of three independent "trajectory" processes:

- *Preexisting Objects Trajectories,* $\Psi_K^{\mathbb{O}}(\mathbf{x}, \mathbf{y})$. The ξ terms correspond to objects that are present before the first measurement is made at time $k = 1$. They may continue to exist up to time K, but they may fail to transition at any intermediate time k. They are not reborn once they fail to transition. Thus, preexisting objects generate a cohort of "trajectories" of varying lengths that all begin at $k = 0$.
- *Newborn Object Trajectories,* $\Psi_K^{\mathbb{B}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The η terms corresponds to objects that are born in the time interval $[1, K]$. Once born, they are indistinguishable from preexisting objects.

They generate trajectories that begin at their birth and terminate when they fail to transition. Their trajectories are superposed with those of preexisting objects. The GF of objects born at time k is $G_{B_k}\big(z_k x_k G_{D_k}(y_k)\,\xi_{k+1}\big)$. Objects born at different times are independent, so the product of their GFs is the joint GF of all objects born in the interval $[1, K]$.

- *False Alarms,* $\Psi_K^{\mathbb{F}}(\mathbf{y}, \mathbf{w})$. The ζ terms correspond to false alarms. The numbers of false alarms are different times are independent, so the joint GF for all false alarms the product of the GFs at each time.

A. GF for cardinal filters with no time lag

Recursive filters with no lag are derived from the GF for $K = 1$. We evaluate (8) by nesting the sums. From (8), the GF for $K = 1$ is

$$\Psi_1(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \sum_{n_0, n_1, m_1, b_1, f_1} p(n_0)p(b_1)p(f_1)p(n_1|n_0, b_1)p(m_1|n_1, f_1) \times x_0^{n_0} x_1^{n_1} y_1^{m_1} z_1^{b_1} w_1^{f_1}.$$

The product in the summand is reordered as

$$\underbrace{p(n_0)x_0^{n_0}}\,\underbrace{p(b_1)z_1^{b_1}}\,\underbrace{p(n_1|n_0, b_1)x_1^{n_1}}\,\underbrace{p(f_1)w_1^{f_1}}\,\underbrace{p(m_1|n_1, f_1)y_1^{m_1}}.$$

Arranging the sums in the same order gives

$$\sum_{n_0, n_1, m_1, b_1, f_1} = \sum_{n_0} \left\{ \sum_{b_1} \left\{ \sum_{n_1} \left\{ \sum_{f_1} \left\{ \sum_{m_1} (\,\cdot\,) \right\} \right\} \right\} \right\}.$$

The innermost sum,

$$\sum_{m_1} p(m_1|n_1, f_1)y_1^{m_1}, \tag{21}$$

is identical to (17) with $k = 1$. From (19), it is equal to

$$\big(G_{D_1}(y_1)\big)^{n_1} y_1^{f_1}. \tag{22}$$

Multiplying by $p(f_1)w_1^{f_1}$ and summing over f_1 gives

$$\sum_{f_1} p(f_1)w_1^{f_1} \left\{ \big(G_{D_1}(y_1)\big)^{n_1} y_1^{f_1} \right\} = \big(G_{D_1}(y_1)\big)^{n_1} \sum_{f_1} p(f_1)\big(y_1 w_1\big)^{f_1}. \tag{23}$$

The sum is the GF (11) of the number of false alarms at time $k = 1$, but with a different argument. Thus, (23) equals

$$\big(G_{D_1}(y_1)\big)^{n_1} G_{F_1}(y_1 w_1). \tag{24}$$

Multiplying by $p(n_1|n_0, b_1)x_1^{n_1}$ and summing over n_1 gives

$$\sum_{n_1} p(n_1|n_0, b_1)x_1^{n_1} \left\{ \big(G_{D_1}(y_1)\big)^{n_1} G_{F_1}(y_1 w_1) \right\} = G_{F_1}(y_1 w_1) \sum_{n_1} p(n_1|n_0, b_1)\big(x_1 G_{D_1}(y_1)\big)^{n_1}. \tag{25}$$

The sum is the conditional GF (13) at time $k = 1$ evaluated for the argument $x_1 G_{D_1}(y_1)$. It follows from (15) that (25) equals

$$G_{F_1}(y_1 w_1)\Big(G_{S_1}\big(x_1 G_{D_1}(y_1)\big)\Big)^{n_0}\Big(x_1 G_{D_1}(y_1)\Big)^{b_1}.$$

Multiplying by $p(b_1)z_1^{b_1}$ and summing over b_1 gives

$$G_{F_1}(y_1 w_1)\Big(G_{S_1}\big(x_1 G_{D_1}(y_1)\big)\Big)^{n_0} \sum_{b_1} p(b_1)\Big(z_1 x_1 G_{D_1}(y_1)\Big)^{b_1}.$$

Using the GF (10), this expression is equal to

$$G_{F_1}(y_1 w_1)\, G_{B_1}\big(z_1 x_1 G_{D_1}(y_1)\big)\, \big(G_{S_1}\big(x_1 G_{D_1}(y_1)\big)\big)^{n_0}. \tag{26}$$

Finally, multiplying by $p(n_0)x_0^{n_0}$ and summing over n_0 gives

$$\begin{aligned}\Psi_1(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) &= G_{F_1}(y_1 w_1) \\ &\quad \times G_{B_1}\big(z_1 x_1 G_{D_1}(y_1)\big) \\ &\quad \times G_{N_0}\big(x_0 G_{S_1}(x_1 G_{D_1}(y_1))\big).\end{aligned} \tag{27}$$

Note that $\Psi_1(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = 1$ for $x_0 = x_1 = y_1 = z_1 = w_1 = 1$.

The terms in the exponent of (27) are the logarithms of the GFs of the three *a priori* variables evaluated with complicated arguments. The arguments in these priors correspond to the principle that "water flows downhill from the source," where downhill is defined by the directed edges in Fig. 1.

The goal is to find the distribution of N_1 conditioned on the number of measurements $m_1 \geq 0$. The number of objects at the initial time is unknown, as are the numbers of new objects and false alarms. To marginalize over them, we set $x_0 = z_1 = w_1 = 1$, and define the GF

$$\boxed{\begin{aligned}\Psi_1(x_1, y_1) &\equiv G_{F_1}(y_1) \\ &\quad \times G_{B_1}\big(x_1 G_{D_1}(y_1)\big) \\ &\quad \times G_{N_0}\big(G_{S_1}(x_1 G_{D_1}(y_1))\big).\end{aligned}} \tag{28}$$

The GF of the posterior pmf of the number of objects conditioned on $M_1 = m_1$ measurements is the normalized derivative,

$$G_{N_1|m_1}(x_1) = \frac{D^{\,0,m_1}\Psi_1(x_1, 0)}{D^{\,0,m_1}\Psi_1(1, 0)}, \tag{29}$$

where, for all $m \geq 0$,

$$D^{\,0,m}\Psi_1(x_1, 0) = \frac{\partial^m}{\partial y_1^m}\Psi_1(x_1, y_1)\bigg|_{y_1=0}. \tag{30}$$

The mean of the posterior pmf at time $k = 1$ is the derivative

$$\widehat{N}_{1|1} = \frac{\partial}{\partial x_1} G_{N_1|m_1}(x_1)\Big|_{x_1=1} \equiv \frac{D^{\,1,m_1}\Psi_1(1, 0)}{D^{\,0,m_1}\Psi_1(1, 0)}. \tag{31}$$

These expressions hold for arbitrary GFs.

B. GF for cardinal filters with a one step time lag

Filters with a one step lag correspond to $K = 2$. The backward recursion for the joint GF gives

$$\begin{aligned}\Psi_2(\mathbf{x},\mathbf{y}, \mathbf{z}, \mathbf{w}) &= G_{F_1}(y_1 w_1)\, G_{F_2}(y_2 w_2) \\ &\quad \times G_{B_1}\big(z_1 x_1 G_{D_1}(y_1) G_{S_2}(x_2 G_{D_2}(y_2))\big) \\ &\quad \times G_{B_2}\big(z_2 x_2 G_{D_2}(y_2)\big) \\ &\quad \times G_{N_0}\big(x_0 G_{S_1}\big(x_1 G_{D_1}(y_1) G_{S_2}(x_2 G_{D_2}(y_2))\big)\big).\end{aligned}$$

An intuitive interpretation of this expression can be found by parsing the factors. This pleasant task is left to the reader. Marginalizing over all variables at time $k = 2$ by setting $x_2 = y_2 = z_2 = w_2 = 1$ reduces it to the GF (27) for $k = 1$, as theory requires.

For a filter of lag 1, we marginalize over the unobserved numbers of new objects and false alarms at times $k = 1, 2$ and the number of objects at times $k = 0, 2$ by setting $x_0 = x_2 = z_1 = z_2 = w_1 = w_2 = 1$. This gives an explicit expression for the desired GF,

$$\boxed{\begin{aligned}\Psi_2(x_1, y_1, y_2) &= G_{F_1}(y_1)\, G_{F_2}(y_2) \\ &\quad \times G_{B_1}\big(x_1 G_{D_1}(y_1) G_{S_2}(G_{D_2}(y_2))\big) \\ &\quad \times G_{B_2}\big(G_{D_2}(y_2)\big) \\ &\quad \times G_{N_0}\big(G_{S_1}\big(x_1 G_{D_1}(y_1) G_{S_2}(G_{D_2}(y_2))\big)\big).\end{aligned}} \tag{32}$$

The GF of the number of objects at time $k = 1$ conditioned on $M_1 = m_1 \geq 0$ and $M_2 = m_2 \geq 0$ is the normalized derivative,

$$G_{N_1|m_1,m_2}(x_1) = \frac{D^{\,0,m_1,m_2}\Psi_2(x_1,0,0)}{D^{\,0,m_1,m_2}\Psi_2(1,0,0)}, \tag{33}$$

where

$$D^{\,0,m_1,m_2}\Psi_2(x_1,0,0) = \frac{\partial^{m_1}}{\partial y_1^{m_1}}\frac{\partial^{m_2}}{\partial y_2^{m_2}}\Psi_2(x_1,y_1,y_2)\Big|_{\substack{y_1=0\\y_2=0}}. \tag{34}$$

The mean of the posterior pmf at time $k = 1$ is the derivative

$$\widehat{N}_{1|2} = \frac{\partial}{\partial x_1}G_{N_1|m_1,m_2}(x_1)\Big|_{x_1=1} \equiv \frac{D^{\,1,m_1,m_2}\Psi_2(1,0,0)}{D^{\,0,m_1,m_2}\Psi_2(1,0,0)}. \tag{35}$$

These expressions hold for arbitrary GFs.

C. GF for cardinal filters with a two step time lag

A two time step lag corresponds to $K = 3$. The recursion for $K = 3$ gives the joint GF, and marginalizing it appropriately gives the joint GF for the cardinal filter with a lag of 2; explicitly,

$$\begin{aligned}\Psi_3(x_1,y_1,y_2,y_3) &= G_{F_1}(y_1)\,G_{F_2}(y_2)\,G_{F_3}(y_3)\\ &\times G_{B_1}(x_1 G_{D_1}(y_1)\,G_{S_2}(G_{D_2}(y_2)\,G_{S_3}(G_{D_3}(y_3))))\\ &\times G_{B_2}(G_{D_2}(y_2)\,G_{S_3}(G_{D_3}(y_3)))\\ &\times G_{B_3}(G_{D_3}(y_3))\\ &\times G_{N_0}(G_{S_1}(x_1 G_{D_1}(y_1)\,G_{S_2}(G_{D_2}(y_2)\,G_{S_3}(G_{D_3}(y_3))))).\end{aligned}$$

The pattern of the nested sequence of GF evaluations is seen by contrasting this expression with (28) and (32). Similar nesting patterns also appear in other population processes (e.g., the Galton-Watson process [12, Chap. 1]).

V. Bayes Optimal Cardinal Filters

The performance of the exact cardinal filter with no lag is shown on simulated data in Subsection V-A. It is derived from the joint GF (28). Its performance is compared to a cardinal filter with lag of one in Subsection V-B. Cardinal filters with larger lags are also discussed.

A. Cardinal filter with no lag

The number of objects is deterministic, not random, and ranges from 1 to 3 in the time interval $[0, 500]$ with increments/decrements of ± 1 as depicted by the jumps in the black curve in Fig. 2. Measurement counts m_k are available for $k = 1, \ldots, 500$. Object detection probability is stationary with $P_k^D = 0.95$ for all k. Object transition probability is stationary with $P_k^S = 0.97$ for all k. The object birth process is stationary and Poisson distributed with expected number $\lambda_k^B = 0.1$ for all k. The false alarm process is stationary and Poisson distributed with expected number $\lambda_k^F = 4$ for all k. The measurement count m_k is simulated by randomly detecting the number of objects at time k and adding a random number of false alarms. In the simulated data, object births and deaths are not random. Somewhat arbitrarily, we define the signal-to-nose ratio (SNR) as $\mathrm{SNR} = 10\log_{10}(\#objects/\lambda_k^F)$. The SNR with 1 object present is $-6\,\mathrm{dB}$, with 2 objects is $-3\,\mathrm{dB}$, and with 3 is $-1.2\,\mathrm{dB}$.

The pmf of the prior distribution on object count at time $k = 0$ is that no objects are present, so that $\Pr\{N_0 = 0\} = 1$. The GF of this prior is $G_{N_0}(\cdot) = 1$. Thus, from (28), the joint GF of the first time step $k = 1$ of the exact cardinal filter simplifies to

$$\Psi_1(x_1,y_1) \equiv G_{F_1}(y_1)G_{B_1}\big(x_1 G_{D_1}(y_1)\big). \tag{36}$$

The GF of the posterior count is the normalized derivative (29). The coefficients of the series expansion

$$G_{N_1|m_1}(x_1) = \sum_{n=0}^{\infty} A_n^{(0)} x_1^n \tag{37}$$

are the probabilities, $A_n^{(0)} = \Pr\{N_0 = n \,|\, M_1 = m_1\}$.

We write the posterior pmf in an equivalent notation borrowed from analytic combinatorics; for example, if $f(z) = \sum_{n\geq 0} a_n z^n$, then $a_n = [z^n]f(z)$. In this notation, $A_n^{(0)}$ is written

$$\Pr\{N_1 = n \,|\, M_1 = m_1\} = \big[x_1^n\big]\, G_{N_1|m_1}(x_1). \tag{38}$$

The MAP (maximum *a posteriori*) estimate of the number of objects in this notation is

$$\widehat{N}_{1|1}^{\mathrm{MAP}} = \arg\max_n \Big\{ \big[x_1^n\big]\, G_{N_1|m_1}(x_1) \Big\}. \tag{39}$$

The second step of the Bayesian recursion uses the joint GF (28) but the initial GF $G_{N_0}(\cdot)$ is replaced by $G_{N_1|m_1}(\cdot)$. The GF of the posterior pmf for the second step is, using (37),

$$\begin{aligned}G_{N_1|m_1}(x_1) &= G_{F_1}(y_1)G_{B_1}\big(x_1 G_{D_1}(y_1)\big)\\ &\times \sum_{n=0}^{\infty} A_n^{(0)}\big(G_{S_1}(x_1 G_{D_1}(y_1))\big)^n.\end{aligned} \tag{40}$$

Later steps of the filter take the same form.

In every step of the recursion, we compute the required derivatives and series expansion up to degree 15 using Mathematica. Proceeding in this way gives the (exact) cardinal filter at time k for a given count sequence $\{m_k\}_{k=1}^{500}$.

Fig. 2 shows the performance of the Bayes optimal cardinal filter for one measurement sequence. The simulated number of objects starts at 1, moves one step at a time up to 3 and back to 1 at the end. The simulated counts range from 0 to 14; they are depicted by the ragged black line. The blue line connects the means of the conditional pmfs, which are computed by evaluating (31) at each time step k. The pmf at each time k is computed by evaluating the power series (37). The MAP estimate of the number of objects is the location of the brightest orange box at each time k.

A close look at Fig. 2 shows that the MAP estimate is often correct or within ± 1 of the correct number of objects. Moreover, the support of the posterior pmf is typically slightly bigger than 1, which is significantly smaller than the support of the Poisson distribution of the number of false alarms. Nonetheless, it is apparent from the figure that estimating the number of objects using only count data and the statistical model depicted graphically in Fig. 1 is a nontrivial exercise.

Fig. 3 shows the the posterior pmf of the cardinal filter averaged over 100 independent Monte Carlo trials. It shows that the MAP estimator of the cardinal filter performs well, at least on average.

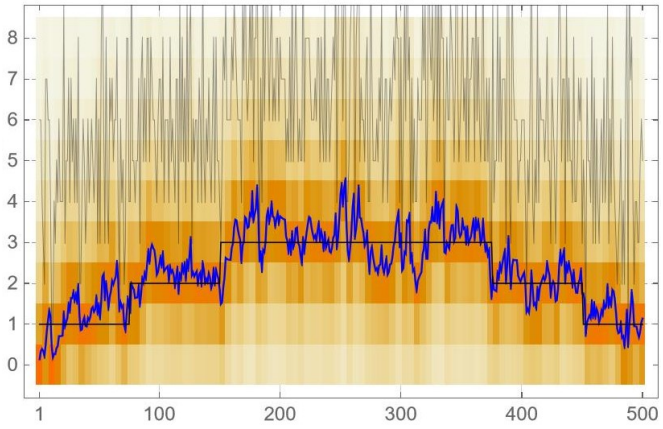

Fig. 2. Heat map of the pmf of the cardinal filter with parameters $P_k^S = 0.97$ and $P_k^D = 0.95$. Mean number of false alarms is $\lambda_k^F = 4$. The stepped black line is the simulated number of objects. The ragged black line is the input sequence of counts. The blue line is the mean of the pmf of the filter.

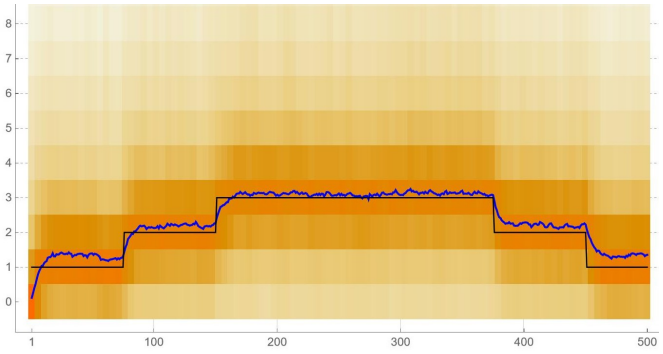

Fig. 3. Heat map of the pmf of the cardinal filter, averaged over 100 Monte Carlo trials. Filter parameters are those of Fig. 2. The blue line is the mean of the posterior pmf, averaged over the 100 trials.

However, it also shows that the cardinal filter is biased. The bias is also seen in other examples (not given here). The size of the bias depends on the filter parameters and the SNR. The bias in Fig. 3 decreases with increasing SNR.

The bias may be due to the fact that the time-varying number of objects in the simulated data sequences is an unlikely realization of the statistical model. The parameters chosen in these examples were selected on the basis of intuition. One such estimation algorithm is discussed in Section VII.

B. Cardinal filters with lag

Exact expressions for cardinal filters with lags are determined by differentiating the joint GF. In general, for a filter of lag $K-1 \geq 0$, we compute the joint GF $\Psi_K(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{w})$ using the backward recursion, and then set $z_{1:K} = w_{1:K} = 1$, and $x_0 = x_{k:k\neq 1} = 1$. The resulting marginal GF is denoted $\Psi_K(x_1, y_{1:K})$. Using the derivative notation of (34), the GF of the number of objects conditioned on the counts $m_{1:K} = (m_1, \ldots, m_K)$ is the normalized derivative evaluated at the origin,

$$G_{N_1|m_{1:K}}(x_1) = \frac{D^{\,0,m_{1:K}}\Psi_K(x_1, 0_{1:K})}{D^{\,0,m_{1:K}}\Psi_K(1, 0_{1:K})}. \tag{41}$$

Just to be clear, the numerator requires computing a mixed derivative of order m_1 with respect to y_1, order m_2 with respect to y_2, etc. The pmf of the posterior is, in the notation of (38),

$$\Pr\{N_1 = n \,|\, m_{1:K}\} = \left[x_1^n\right] G_{N_1|m_{1:K}}(x_1), \quad n \geq 0. \tag{42}$$

The posterior pmf of the cardinal filter with lag $K-1$ is computed by expanding $G_{N_1|m_{1:K}}(x_1)$ in a power series about $x_1 = 0$. The mean of the posterior pmf is the derivative (cf. (35))

$$\widehat{N}_{1|K} = \frac{D^{\,1,m_{1:K}}\Psi_K(1, 0_{1:K})}{D^{\,0,m_{1:K}}\Psi_K(1, 0_{1:K})}. \tag{43}$$

Computing this derivative is tantamount to computing the mean from the power series.

We ran the cardinal filter with a lag of 1 ($K = 2$) on exactly the same data as the previous example. The results differed little from those of the zero-lag filter (Fig. 3) and are not shown here. Neither the observed average biases (with 100 trials) nor the delays in adapting to changes in the number of objects were statistically significant.

If lagged cardinal filters are to improve object count estimates, it is clear that the lag must be larger than 1. Unfortunately, exact filters are too time consuming to compute for lags of 2 or more using our implementation, so they are not presented.

VI. Saddlepoint Optimal Cardinal Filters

The computational complexity of cardinal filters grows rapidly with the size of the count and the length of the lag. Consequently, approximations are needed to study their performance. In this section saddlepoint methods are applied to approximate the posterior pmf of optimal cardinal filters.

The count sequence $m_{1:K}$ is given. To compute the posterior pmf of the cardinal filter for this sequence, we expand the GF of the filter ratio (41) into a power series about zero. The coefficients of the series are mixed derivatives, so we approximate them by the saddlepoint method. Except for the constant term, each coefficient requires a separate saddlepoint approximation.

The denominator in (41) is a normalizing constant, so

$$\begin{aligned}\Pr\{N_1 = n \,|\, m_{1:K}\} &= \left[x_1^n\right] G_{N_1|m_{1:K}}(x_1) \\ &\propto \left[x_1^n\right] D^{\,0,m_{1:K}}\Psi_K(x_1, 0_{1:K}) \\ &= D^{\,n,m_{1:K}}\Psi_K(0, 0_{1:K}) \\ &\equiv I_n.\end{aligned}$$

Since the proportionality constant is the same for all n, we truncate the power series at a sufficiently high degree, compute the numbers I_n, and then normalize them so that they sum to 1.

We use the method in the Appendix to approximate I_n. From (49),

$$I_n = \frac{1}{(2\pi i)^{K+1}} \oint_{C(r_0)} \cdots \oint_{C(r_n)} \frac{\Psi_K(x_1, y_{1:K})\, \mathrm{d}x_1 \mathrm{d}y_1 \cdots \mathrm{d}y_K}{x_1^{n+1}\, y_1^{m_1+1} \cdots y_K^{m_K+1}}.$$

Let $\mathbf{s} = (s_0, s_1, \ldots, s_K)$. From (50), the action for this integral is

$$\mathcal{L}_n(\mathbf{s}) = \log \Psi_K(s_{0:K}) - (n+1)\log s_0 - \sum_{k=1}^{K}(m_k+1)\log s_k,$$

and the saddlepoint is, from (51),

$$\hat{\mathbf{s}}(n) = \underset{\mathbf{s}>0}{\operatorname{argmin}}\ \mathcal{L}_n(\mathbf{s}).$$

Using (52), the saddlepoint approximation of I_n is

$$I_n \approx \frac{e^{\mathcal{L}(\hat{\mathbf{s}}(n))}}{(2\pi)^{(K+1)/2}\sqrt{\left|H_n(\hat{\mathbf{s}}(n))\right|}},$$

where $\left|H_n\left(\hat{\mathbf{s}}(n)\right)\right|$ is the determinant of the $(K+1)\times(K+1)$ Hessian matrix of the action $\mathcal{L}_n(\mathbf{s})$ evaluated at $\mathbf{s} = \hat{\mathbf{s}}(n)$. Absorbing the constant into the proportionality constant, we have the elegant expression

$$\log I_n \approx \mathcal{L}\left(\hat{\mathbf{s}}(n)\right) - \tfrac{1}{2}\log\left|H_n(\hat{\mathbf{s}}(n))\right|. \tag{44}$$

The saddlepoint approximation to the pmf is easy to implement if we use a numerical algorithm that returns both the minimum and the Hessian matrix at the minimum.[2]

We reexamine the example of Section V-A, but with a lag of 3 time steps, which corresponds $K = 4$. The filter parameters are unchanged. At each time step, we approximate the Bayes optimal pmf from $n = 0$ to $n = 10$ using the expression (44).

The performance of the lagged cardinal filter on the same data as in Fig. 2 is shown in Fig. 4. It is apparent by comparing these two figures that the cardinal filter with a lag of 3 is, in some sense, smoother in that it has smaller changes in the estimated object count from one time step to the next. It is also apparent that the variance in the estimated count is comparable to that of the no lag filter.

Fig. 5 shows the average of the lagged filter output over 100 Monte Carlo trials. The data in these 100 trials are identical to the data used to generate Fig. 3. It is clear that the lagged filter is significantly

[2]If the determinant $\left|H_n(\hat{\mathbf{s}}(n))\right|$ is a "weak" function of n, it may not need to be computed. See the discussion of saddlepoint bounds in [10, p. 549].

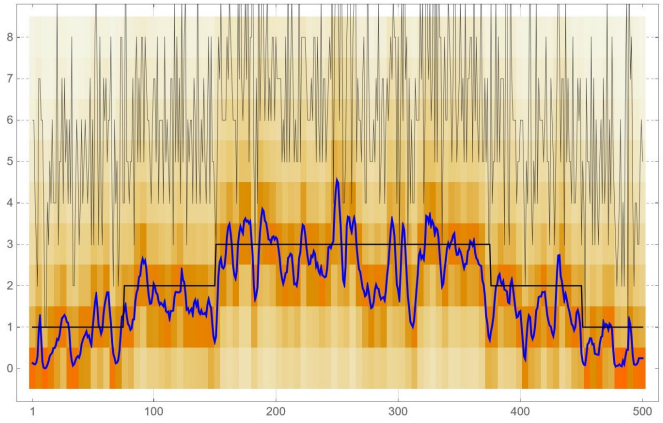

Fig. 4. Heat map of the pmf of the cardinal filter with lag 3 ($K = 4$) for the same count sequence and filter parameters as Fig. 2. The blue line is the mean of the pmf. The stepped black line is the simulated number of objects. The ragged black line is the input count sequence.

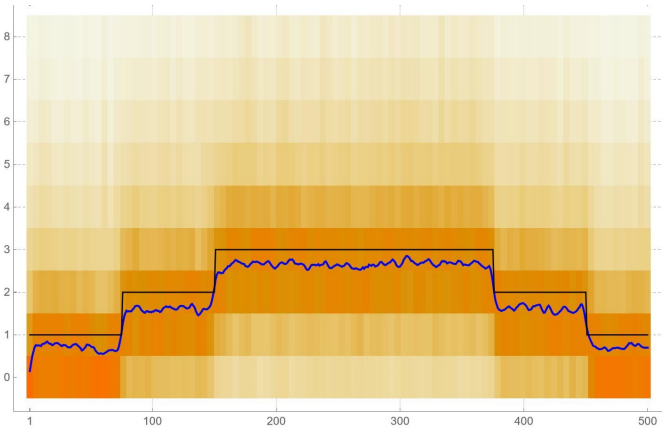

Fig. 5. Heat map of the pmf of the cardinal filter with a lag of 3 time steps ($K = 4$), averaged over 100 Monte Carlo trials. The blue line is the mean of the averaged pmf. Filter parameters are those of Fig. 4.

more biased than the filter with no lag. The variance in the pmf is also apparently larger than that of the no lag filter. This result is rather surprising, but it could be a result of the bias. On average the lagged filter does appear to detect the changes in the number of objects somewhat more rapidly than the no lag filter, but on any one run, as shown in Fig. 4, it is difficult to see this slight improvement.

VII. Parameter Estimation

The examples in Sections V and VI are limited to one sensor data stream. By experimenting with the different filter parameters, it is clear that performance can be improved by adaptively "tuning" them to reduce the bias and, potentially, the variance in the estimate as well. Filter parameter estimation is the topic of this section.

The normalizing constant is not used in Section VI to compute the pmf, but it is fundamental to the problem of filter parameter estimation. Let θ denote a parameter in the statistical model to be estimated from the count sequence $m_{1:K}$. An initial feasible value of θ is given. The basic idea is simple. Use the backward recursion to find the joint GF of the likelihood of the count sequence. To show the dependence of the GF on θ, we write the GF as $\Psi_K(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}; \theta)$. Setting $x_{0,K} = z_{1:K} = w_{1:K} = 1$ gives the marginal GF

$$\Psi_K(y_{1:K}; \theta) \equiv \Psi_K(1, \mathbf{y}, 1, 1; \theta). \tag{45}$$

The mixed derivative of the marginal GF

$$L(\theta) \equiv D^{\,m_{1:K}} \Psi_K(y_{1:K}; \theta)\Big|_{y_{1:K}=0} \tag{46}$$

is the likelihood function of θ conditioned on the count data $m_{1:K}$. From (50), the action of the likelihood function is

$$\mathcal{L}_\theta(t_{1:K}) = \log \Psi_K(t_{1:K}, \theta) - \sum_{k=1}^{K} (m_k + 1) \log t_k, \tag{47}$$

where $t_{1:K} > 0$. The saddlepoint $\hat{\mathbf{t}} \equiv \hat{t}_{1:K}$ minimizes the action. It is a function of θ, and we denote the dependence explicitly by writing $\hat{\mathbf{t}}_k(\theta)$. Taking the logarithm of the saddlepoint approximation of the likelihood function $L(\theta)$ and dropping irrelevant constants gives the approximate loglikelihood function for θ as

$$\ell(\theta) \approx \mathcal{L}_\theta\big(\hat{\mathbf{t}}(\theta)\big) - \tfrac{1}{2} \log\big|H_{\mathcal{L}}\big(\hat{\mathbf{t}}(\theta)\big)\big|, \tag{48}$$

where $H_{\mathcal{L}}(\hat{\mathbf{t}}(\theta))$ is the Hessian matrix of the action $\mathcal{L}_\theta(t_{1:K})$ evaluated at the saddlepoint. The gradient of this approximation with respect to θ is easily computed by the complex step method [11]. The initial value θ can then by updated by suitable gradient methods. Note that changing the parameter changes the action, so it is necessary to recompute the saddlepoint at every iteration.

VIII. Concluding Remarks

It is apparent from the examples that accurately estimating the number of objects from count data that is "corrupted by insertions and deletions" (indels) is a difficult task. When the application is tracking real objects, or targets, it is clear that augmenting the count data with, say, a set of kinematic point measurements, will improve the count estimate. This would make the data discrete-continuous, which complicates the formulation of the approach but does not fundamentally alter it (e.g., the backward recursion becomes a sequence of nested integrals [6]). Augmenting the count data in this way detracts from one of the salient advantages cardinality filters — they count all targets regardless of their motion.

Multiple sensors may be counting detections in the same prescribed field of view. If the sensor counts are independent conditioned on the number of objects, and the numbers of false alarms in the sensors are independent, then the joint GF of all the sensors is computed by modifying the backward recursion — it needs to use a sensor-specific indeterminant $\mathbf{y}$ and the GF $G_{F_k}(y_k w_k)$ must be replaced by the product of the GFs of the sensor false alarm processes. This increases the order of the derivatives in the cardinality filter, but the saddlepoint method still applies. The potential advantage of using multiple sensors is that the resulting MAP estimate would be less biased and have smaller variance.

The MAP estimate gives, at each time k, the individually most likely number of objects. It also maximizes the expected number of correct individual counts. The sequence of individual MAP estimates is different from the most likely sequence of the number of objects. Given the Markovian nature of the statistical model, the most likely sequence of object number is the Viterbi sequence. The Viterbi sequence may provide an estimate that is better suited to some applications. How to find the Viterbi sequence using the saddlepoint approximation remains to be explored.

The algorithm outlined in Section VII adapts the filter parameters to "fit" the available data and thereby improve performance. It is unique in one special regard — filter parameters are estimated by stochastic gradient descent applied to the saddlepoint approximation of the data likelihood function. The gradient is easily computed to machine precision by the "complex step" method [11]. This technique is applicable to any problem whose likelihood function is approximated by saddlepoint method. It is the subject future work.

Exact cardinal filters are symbolic mixed derivatives, which is why all the examples in this paper were implemented using *Mathematica*. The implementation gave us a first look at filter performance. It was not designed to be fast, so it was perforce inefficient and slow.

Saddlepoint approximations do *not* compute symbolic derivatives. The method enables the numerical study of much larger problems without the burden of symbolic processing.

Appendix – Saddlepoint Approximation

Let $\mathbf{k} = (k_1, \ldots, k_n)$, where integers $k_j \geq 0$, $j = 1, \ldots, n$, and $n \geq 1$. Let $f : \mathbb{C}^n \to \mathbb{C}$ is a multivariate analytic function of $z = (z_1, \ldots, z_n)$ in an open neighborhood of the origin in $\mathbb{C}^n$. Let $f^{\mathbf{k}}(z)$ denote the mixed derivative of f. The multivariate Cauchy integral to be approximated is

$$\begin{aligned} I_{\mathbf{k}} &= \frac{1}{k_1! \cdots k_n!} f^{\mathbf{k}}(z)\Big|_{z=0 \in \mathbb{C}^n} \\ &= \frac{1}{(2\pi i)^n} \oint_{C(r_1)} \cdots \oint_{C(r_n)} f(z) \, \frac{\mathrm{d}z_1 \cdots \mathrm{d}z_n}{z_1^{k_1+1} \cdots z_n^{k_n+1}}. \end{aligned} \tag{49}$$

The function f can be any of the GFs used in this paper — they are analytic in each variable separately, which (by Hartogs' Theorem) makes them jointly analytic. The domain of integration is the polydisc $C(r_1) \times \cdots \times C(r_n)$, where $C(r_j)$ is a circle of radius $r_j > 0$ centered at the origin of the j^{th} copy of the complex plane in the Cartesian product $\mathbb{C}^n$. Let $r = (r_1, \ldots, r_n) \in \mathbb{R}_+^n$.

The computational complexity of the saddlepoint approximation depends on the number of variables n but *not* on the order of the derivative. The action[3] $\mathcal{L}_{\mathbf{k}} : \mathbb{R}_+^n \to \mathbb{R}$ is defined by

$$\begin{aligned} \mathcal{L}_{\mathbf{k}}(r) &= \log \frac{f(r)}{r_1^{k_1+1} \cdots r_n^{k_n+1}} \\ &= \log f(r) - \sum_{j=1}^{n} (k_j + 1) \log r_j. \end{aligned} \tag{50}$$

The saddlepoint $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_n) \in \mathbb{R}_+^n$ is defined by

$$\hat{r} = \underset{\mathbf{r} > 0}{\operatorname{argmin}} \; \mathcal{L}_{\mathbf{k}}(r). \tag{51}$$

The assumption of nonnegative coefficients guarantees that the saddlepoint exists and is unique in each variable separately, provided that $f(z)$ is not a polynomial in the variable [10].

The integral $I_{\mathbf{k}}$ is approximated by a series expansion about the saddlepoint. The first term in the series is

$$I_{\mathbf{k}} \approx \frac{e^{\mathcal{L}_{\mathbf{k}}(\hat{r})}}{(2\pi)^{n/2} \sqrt{|H(\hat{r})|}} \tag{52}$$

$$= \frac{1}{(2\pi)^{n/2} \sqrt{|H(\hat{r})|}} \, \frac{f(\hat{r})}{\hat{r}_1^{k_1+1} \cdots \hat{r}_n^{k_n+1}}, \tag{53}$$

where $|H(\hat{r})|$ is the determinant of the $n \times n$ Hessian matrix of the action evaluated at $\hat{r}$,

$$\begin{aligned} H(\hat{r}) &= \nabla_{rr} \, \mathcal{L}_{\mathbf{k}}(r)\Big|_{r=\hat{r}} \\ &= \operatorname{Diag}\left(\frac{k_1 + 1}{\hat{r}_1^2}, \ldots, \frac{k_n + 1}{\hat{r}_n^2} \right) + \nabla_{zz} \log f(z)\Big|_{z=\hat{r}}. \end{aligned} \tag{54}$$

This term is often used to derive asymptotic expansions for large values of n. See [3] for additional terms in the series and an analysis of the approximation error.

[3]The name is borrowed from the Lagrangian approach to mechanics in physics. The function is not given a name in the saddlepoint literature.

The saddlepoint approximation is sensitive to changes of variables. For example, substituting $z_j = r_j e^{i\theta_j}$ into (49) gives

$$I_{\mathbf{k}} = \frac{1}{(2\pi)^n} \int_0^{2\pi} \cdots \int_0^{2\pi} \frac{f\left(r_1 e^{i\theta_1}, \ldots, r_n e^{i\theta_n}\right)}{r_1^{k_1} \cdots r_n^{k_n}} \, \mathrm{d}\theta_1 \cdots \mathrm{d}\theta_n. \tag{55}$$

Letting $\theta_j = 0$ gives the action (compare to (50))

$$\mathcal{L}_c(r) = \log f(r) - \sum_{j=1}^{n} k_j \log r_j. \tag{56}$$

If $\hat{s}$ denotes the saddlepoint of (56), then

$$I_{\mathbf{k}} \approx \frac{1}{(2\pi)^{n/2} \sqrt{|H(\hat{s})|}} \, \frac{e^{\mathcal{L}_c(\hat{s})}}{\hat{s}_1 \cdots \hat{s}_n} \tag{57}$$

$$= \frac{1}{(2\pi)^{n/2} \sqrt{|H(\hat{s})|}} \, \frac{f(\hat{s})}{\hat{s}_1^{k_1+1} \cdots \hat{s}_n^{k_n+1}}. \tag{58}$$

The two approximations differ only in the choice of saddlepoint.

The gradient and the Hessian of the action are computed to machine accuracy by the complex step method. See [11] for details.

Fixed point iteration. Setting the gradient $\nabla_r \, \mathcal{L}_{\mathbf{k}}(r) = 0$ gives necessary conditions for the minimum (51). Thus, $\hat{r}$ satisfies

$$r_i = (k_i + 1) \left(\frac{\partial}{\partial r_i} \log f(r) \right)^{-1}, \quad 1 \leq i \leq n. \tag{59}$$

The mathematical form of the system suggests the following multivariate fixed point iteration: For $\ell = 0, 1, 2, \ldots$, compute

$$r_i^{(\ell+1)} = (k_i + 1) \left(\frac{\partial}{\partial r_i} \log f(r)\Big|_{r=r^{(\ell)}} \right)^{-1}, \quad 1 \leq i \leq n, \tag{60}$$

where $r^{(\ell)} = \left(r_1^{(\ell)}, \ldots, r_n^{(\ell)}\right)$ and $r^{(0)} > 0$ is a given initialization. A sufficient condition for convergence is that the L_1 norms of the gradients of the iterated functions be bounded by 1, where the iterated functions here are $g_i(r) = (k_i + 1)(\frac{\partial}{\partial r_i} \log f(r))^{-1}$.

References

[1] T. Liu, Y. Lin, and E. Zhou, "Bayesian stochastic gradient descent for stochastic optimization with streaming input data," SIAM J. on Optimization, 34(1), 389-418, 2024.

[2] A. Balakrishnan, M.S. Pangburn, and E. Stavrulaki, "'Stack them high, let'em fly': lot-sizing policies when inventories stimulate demand," Management Science, 50, 630–644, 2004.

[3] J.L. Jensen, *Saddlepoint Approximations*, Clarendon Press, Oxford, 1995.

[4] R.L. Streit, "How to count targets given only the number of measurements," 16th International Conf. on Information Fusion, Istanbul, Turkey, July 2013.

[5] R. Streit, R.B. Angle, and M. Efe, *Analytic Combinatorics for Multiple Object Tracking*, Springer, 2021.

[6] R.L. Streit, "Interval/smoothing filters for multiple object tracking via analytic combinatorics," 20th International Conf. on Information Fusion, Xi'an, China, July 2017.

[7] A. Bakut and N.A. Ivanchuk, "Calculation of a-posteriori characteristics of flow of resolved objects," Engineering Cybernetics, vol. 14.6, pp. 148–156, 1976. (Available at https://www.stochasticflows.com)

[8] D. Clark, R. Streit, and A. Narikov, "Stochastic flows – a primer on early multi-object filtering work with point processes," 26th International Conf. on Information Fusion, Charleston, USA, 2023.

[9] D.E. Clark, A. Narykov, and R.L. Streit, "Stochastic flows: the foundation of multi-object filtering with point processes." [Online Project] https://www.stochasticflows.com

[10] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, United Kingdom, 2009.

[11] G. Lantoine, R.P. Russell, T. Dargent, "Using Multicomplex Variables for Automatic Computation of High-Order Derivatives," ACM Trans. on Math. Software, Vol. 38, No. 3, p. 16:1-16:21, April 2012.

[12] K.B. Athreya and P.E. Ney, *Branching Processes*, Springer-Verlag, Berlin, 1972. (Dover edition, 2004.)