Efficient Bayesian Methods for Counting Processes in Partially Observable Environments

Ferdian Jovan
University of Birmingham

Jeremy Wyatt
University of Birmingham

Nick Hawes
University of Oxford

Abstract

When sensors that count events are unreliable, the data sets that result cannot be trusted. We address this common problem by developing practical Bayesian estimators for a partially observable Poisson process (POPP). Unlike Bayesian estimation for a fully observable Poisson process (FOPP) this is non-trivial, since there is no conjugate density for a POPP and the posterior has a number of elements that grow exponentially in the number of observed intervals. We present two tractable approximations, which we combine in a switching filter. This switching filter enables efficient and accurate estimation of the posterior. We perform a detailed empirical analysis, using both simulated and real-world data.

1 Introduction

There is an increasing trend toward the creation of large data sets of counts of events, often gathered using automatic event detection techniques. Application domains include detection and counting of people, vehicles, cells, pixel level events in cameras, minor earthquakes and geological events, astronomical bodies, etc. Where these counts are made using sensor data, both humans and all the currently available event counting algorithms have some level of unreliability. This means that large data-sets typically contain systematic errors that lead to bias in the statistical estimates produced by the event detection and counting processes. In this paper, we address this problem by formulating a *partially observable Poisson process* (POPP)[1]. We contrast this with standard fully observable Poisson process (FOPP).

We make several technical contributions. First, we address the problem of how to efficiently correct counts made by either single or multiple unreliable counting devices. The main difficulty in inference arises from the fact that a POPP has no conjugate density. We address this by utilising two approximations. One (the *Gamma filter*) is fast, but prone to drift from the true posterior in certain circumstances. The second (the *histogram filter*) is slower but avoids drift. We also combine them in a *switching filter* that avoids drift while being reasonably quick to compute. We demonstrate the properties of the filters by numerical simulations. Finally, we show the benefit of the POPP model and the switching filter on a person counting task performed by a mobile robot.

2 Preliminaries - the FOPP process

A fully observable Poisson process is a counting process $N(t_1, t_2)$ where a counter tells, without error, the number of events that occurred during a specified interval (t_1, t_2). $N(t_1, t_2) = x_i$ states that in the i-th observation of interval (t_1, t_2), there are x_i events. The number of events $N(t_1, t_2)$ in a finite interval of length $t = t_2 - t_1$ obeys the Poisson distribution,

$$Poi(N(t_1, t_2) = x \mid \lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$$

where λ represents the *arrival rate, mean count, intensity*, or *expected number of events* in a fixed interval $(t_1,\ t_2)$. Bayesian estimation for fully observable Poisson processes is straightforward. The conjugate density to the Poisson is a Gamma density

$$Gam(\lambda \mid \alpha, \beta) \quad = \quad \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$$

where α, β are the shape and the rate parameters. An interpretation of these is that we have made β samples

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

[1] We note that similar terms have been used to describe other stochastic processes.

$x'_1, \ldots, x'_\beta$ from a Poisson with cumulative count $\alpha = x'_1 + \ldots + x'_\beta$. The posterior is thus also Gamma:

$$\begin{aligned} P(\lambda \mid x_1, \ldots, x_n) \quad &\propto Poi(x_1, \ldots, x_n \mid \lambda)\ Gam(\lambda \mid \alpha, \beta) \\ &= Gam\left(\lambda \mid \sum_{i=1}^{n} x_i + \alpha, n + \beta\right) \end{aligned}$$

3 Related Work

There are many variations of the basic Poisson process. The Markov modulated Poisson process (MMPP), for example, is a Poisson process in which fully observable counts are conditioned on a latent state which evolves according to a Markov chain (Ludkovski & Sezer, 2012; Ihler, Hutchins, & Smyth, 2006; Scott, 1998; Meier-Hellstern, 1987; Rydén, 1996; Prabhu & Zhu, 1989). Ludkovski and Sezer (2012) employed an MMPP in a reward-maximisation decision-making problem. An extended version of the MMPP includes a non-homogeneous Poisson process which models the posterior probabilities of Poisson arrival rates as a function of time (Ihler et al., 2006; Scott & Smyth, 2003; Ihler, Hutchins, & Smyth, 2007). Hutchins, Ihler, and Smyth (2007) extended the work of (Ihler et al., 2006) from single to multiple counters, and applied it to estimating the occupancy level of a building. This extension also modelled unreliable counters for the first time. A similar example to MMPP is a non-homogeneous Poisson process where the λ parameter is modulated by Weibull type distribution which governs how the λ should increase (or decrease) over time by small amount. This shows its usefulness in predicting noise exceedances, ozone exceedances, and software reliability issues (Guarnaccia, Quartieri, Barrios, & Rodrigues, 2014; Achcar, Barrios, & Rodrigues, 2012; Achcar, 2001). Other work uses non-homogeneous Poisson process (NHPP) to predict re-occurring patterns of human activity level at particular times and places (Jovan, Wyatt, Hawes, & Krajník, 2016; Hawes et al., 2017).

Our work is hence most related to (Hutchins et al., 2007). They used multiple unreliable counters, each at a different exit or entrance. Thus, each sensor is associated with a different Poisson process. For each entrance or exit they used a MMNHPP to estimate the arrival rate and a noise model to capture under- and over-counting. Their work differs from ours in that they are interested in estimation of a single latent variable (occupancy) that influences arrivals at multiple exits and entrances. They employed an MCMC estimator. Here, we consider multiple unreliable sensors applied to estimating the parameter of a single Poisson process. We present three precise and tractable Bayesian estimators for this problem.

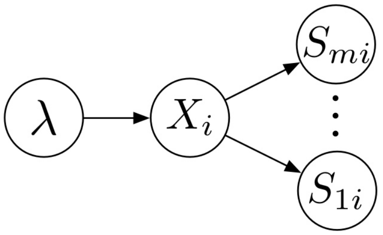


Figure 1: Structure of the POPP.

4 The POPP Process

In the FOPP, the counter is fully reliable. We now remove this assumption. Counters may now either under-count or over-count. We distinguish the *true count* (or simply *count*) and the *sensed count*. The true count x_i is the number of events that actually occurred in the i-th sample from the interval (t_1, t_2). We suppose that we have several sensors, and thus the sensed count s_{ji} is the count given by sensor j in the i-th sample from the interval (t_1, t_2).

We obtain a graphical model with the structure shown in Figure 1. There are m sensors polled every sample i, $\overrightarrow{s_i} = (s_{1i}, \ldots, s_{mi})$. The true count x_i is a latent variable with posterior inferred from $\overrightarrow{s_i}$, and the posterior of λ is inferred from the posterior of x_i after multiple samples $i = 1 \ldots n$.

One way to estimate the parameter λ is by Bayesian averaging the posterior $P(\lambda \mid x_i)$ over all possible count values x_i with mixing proportions equal to the posterior over x_i. The posterior of λ, given n samples $\overrightarrow{s} = (\overrightarrow{s_1} \ldots \overrightarrow{s_n})$, each consisting of m sensors, is:

$$P(\lambda \mid \overrightarrow{s}) = \sum_{x_1=0}^{\infty} \ldots \sum_{x_n=0}^{\infty} P(\lambda \mid \overrightarrow{x})\ P(\overrightarrow{x} \mid \overrightarrow{s}) \qquad (1)$$

where

$$P(\lambda \mid \overrightarrow{x}) = Gam\left(\lambda \mid \sum_{i=1}^{n} x_i + \alpha, n + \beta\right)$$

with $\overrightarrow{x} = (x_1, \ldots, x_n)$ for $1 \leq i \leq n$. This shows the difficulty of belief state estimation in a POPP. There is no conjugate density and the resulting posterior is a sum of countably infinite sums. Even if we place an upper bound x_{max} on the value of each x_i the number of elements in the curtailed approximate posterior grows by a factor x_{max} with every sample. Having made this observation, we will now show how to obtain tractable approximations under conditional independence assumptions.

We now factor $P(\overrightarrow{x} \mid \overrightarrow{s})$. First, we make the assumption that the vector of sensed counts for sample

i, $\overrightarrow{s_i}$, is conditionally independent of all the other sensor vectors given x_i. Then we assume that each sensor is conditionally independent of the other sensors given x_i. Consequently, the probability that the vector of true counts is $\overrightarrow{x}$, given n samples of the vector of m sensed counts $\overrightarrow{s_1}, \ldots, \overrightarrow{s_n}$, is

$$\begin{aligned} P(\overrightarrow{x} \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n}) &\propto P(\overrightarrow{s_1}, \ldots, \overrightarrow{s_n} \mid \overrightarrow{x})\ P(\overrightarrow{x}) \\ &\propto \prod_{i=1}^{n} P(\overrightarrow{s_i} \mid x_i)\ P(x_i) \\ &\propto \prod_{i=1}^{n} \prod_{j=1}^{m} P(s_{ji} \mid x_i)\ P(x_i \mid \overrightarrow{x_{-1}}) \end{aligned} \tag{2}$$

where $\overrightarrow{x_{-1}} = x_{i-1}, \ldots, x_1$.

To complete Eqn. 2, we define $P(s_{ji} \mid x_i)$ and $P(x_i \mid \overrightarrow{x_{-1}})$. $P(x_i \mid \overrightarrow{x_{-1}})$ can be considered as the unconditional distribution of x_i informed by the previous counts $x_{i-1}, \ldots, x_1$. It is obtained by imposing a Gamma prior $Gam(\lambda \mid \alpha, \beta)$ on the count parameter λ of the Poisson distribution $Poi(x_i \mid \lambda)$ and integrating out λ. It is, in fact, the posterior predictive distribution of a Poisson-Gamma mixture, and takes the form of a negative binomial distribution.

$$\begin{aligned} P(x_i \mid \overrightarrow{x_{-1}}) &= \int_{\lambda=0}^{\infty} P(x_i \mid \lambda)\ P(\lambda \mid \overrightarrow{x_{-1}})\ d\lambda \\ &= \int_{\lambda=0}^{\infty} Poi(x_i \mid \lambda)\ Gam(\lambda \mid \alpha, \beta)\ d\lambda \\ &= \int_{\lambda=0}^{\infty} \frac{e^{-\lambda} \lambda^{x_i}}{x_i!} \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} \\ &= \frac{\beta^{\alpha}}{x_i!\ \Gamma(\alpha)} \frac{\Gamma(x_i + \alpha)}{(\beta+1)^{x_i+\alpha}} \\ &= \binom{x_i + \alpha - 1}{x_i} \left(\frac{\beta}{\beta+1}\right)^{\alpha} \left(\frac{1}{\beta+1}\right)^{x_i} \\ &= NB\left(x_i \mid \alpha, \frac{\beta}{\beta+1}\right) \end{aligned} \tag{3}$$

To define an arbitrarily close approximation to the probability $P(s_{ji} \mid x_i)$, we first assume there exists a small enough finite subinterval of length δ for which the probability of more than one event occurring is less than some small value ϵ. With this assumption, the whole interval (t_1, t_2) can be divided into l subintervals $I_1, \ldots, I_l$ of equal size, such that $l > \lambda$. Consequently, the whole interval $(t_1, t_2) = I_1, \ldots, I_l$ is a series of Bernoulli trials, where the k^{th} trial corresponds to whether (1) an event e_k happens with probability λ / l and (2) a sensor j captures the event e_k as d_k at the subinterval I_k.

Given the argument above, the probability of s_{ji} detections given x_i arrivals is the aggregate of the true positives tp_{ji} in x_i sub-intervals, and the false positives fp_{ji} in $l - x_i$ sub-intervals. The probability of a TP for sensor j in a sub-interval is $tpr_j = P_j(d = 1 \mid e = 1)$, and the probability of an FP is $fpr_j = P_j(d = 1 \mid e = 0)$. Thus $P(s_{ji} \mid x_i)$ is defined as a sum of two binomial distributions $B(r \mid n, \pi)$, where the aggregate is constrained to be s_{ji}:

$$P(s_{ji} \mid x_i) = \sum_{tp_{ji}=0}^{x_i} B\Big(tp_{ji} \mid x_i, tpr_j\Big) B\Big(fp_{ji} \mid \Delta x_i, fpr_j\Big) \tag{4}$$

where $s_{ji} = tp_{ji} + fp_{ji}$, $tpr_j = P_j(d = 1 \mid e = 1)$, $fpr_j = P_j(d = 1 \mid e = 0)$, and $\Delta x_i = (l - x_i)$.

Eqn. 1 now makes the posterior of λ depend not only on the nature of the problem, and the settings of hyper-parameters α and β but also on the reliability of each individual sensor. However, as we noted above, it also requires an exponentially rising computation time in the number of samples n.

To provide an efficient estimator we propose three filters, each of which offers an approximation to the posterior. First, we can often approximate the posterior Eqn. 1 with a single Gamma distribution. This gradually worsens as sensor reliability degrades, so we also define a histogram filter. Finally, we can combine these two approximations in a switching filter.

4.1 Strategy I: Gamma filter

Eqn. 1 is an infinite mixture of gamma distributions. Given a matrix of sensor observations $\overrightarrow{s}$, if the sensors were reliable, the $\sum x_i$ would be determined, and Eqn. 1 would simplify by definition to the single gamma posterior for a FOPP. We could thus use a single gamma as an approximation to the posterior under the assumption that the sensors have minor unreliabilities, and expect this approximation to deteriorate with sensor reliability. Thus, in this approximate filter, as observations arrive, we calculate the true posterior as a sum of gamma distributions and then fit a single gamma to it by gradient descent on the KL-divergence. Specifically, to approximate $P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n})$, we use an optimisation algorithm which minimises KL-divergence $D_{KL}(P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n}) \mid\mid Gam(\lambda \mid \alpha, \beta))$.[2] We employ a numerical hill climbing algorithm which changes the β parameter of the Gamma prior to minimise $D_{KL}(P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n}) \mid\mid Gam(\lambda \mid \alpha, \beta))$ given $P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n})$.

[2] Note that this problem of approximating a mixture of gammas is entirely different from the well-known problem of estimating the sum of a several i.i.d. variables, each of which follows a gamma distribution.

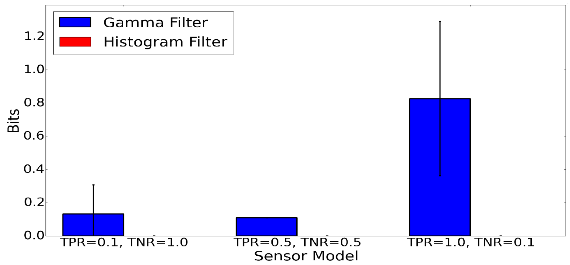


Figure 2: Average KL-divergence (in bits) from the gamma and histogram filters to the true posterior $P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n})$ with variations on the sensor model. Standard error is shown

4.2 Strategy II: Histogram filter

As sensor reliability deteriorates we would expect the Gamma filter to worsen as an approximation to the posterior belief for λ. We may instead use a histogram filter for $P(\lambda \mid \overrightarrow{x})$ by quantising λ:

$$P_{HF}(\lambda \mid \overrightarrow{x}) = \frac{P(x_j \mid \lambda)\; P(\lambda \mid \overrightarrow{x_{-1}})}{\sum_{\lambda_i \in \mathbb{R}} P(x_j \mid \lambda_i)\; P(\lambda_i \mid \overrightarrow{x_{-1}})} \tag{5}$$

and redefine Eqn. 3 as:

$$P_{HF}(x_j \mid \overrightarrow{x_{-1}}) = \sum_{\lambda_i \in \mathbb{R}} P(x_j \mid \lambda)\; P(\lambda \mid \overrightarrow{x_{-1}}) \tag{6}$$

with $\overrightarrow{x_{-1}} = x_{j-1}, \ldots, x_1$.

The advantage of this filter over the gamma filter is that it can track the posterior to an arbitrary fidelity via finer quantisation. The disadvantage is increased computational time, as typically its run-time is 10-40 times that of the gamma filter.

4.3 Strategy III: Switching filter

As we shall see from experimental data, the gamma filter is fast and can provide a good approximation of the posterior when sensors are relatively reliable. The histogram filter is slow, but can track the posterior near perfectly given enough computation. This suggests the possibility of a switching filter. This runs either one of the filters, using a switching mechanism in each posterior update to determine which one to run. Given the latest sampled sensor vector $\overrightarrow{s_i}$, the posterior $P(\lambda \mid \overrightarrow{s_i})$ is calculated by Eqn. 1. The gamma filter then returns an approximation to this. If the KL-divergence D_{KL} exceeds a threshold θ then the histogram filter takes over. Each step, the gamma filter still runs, using the histogram filter as a prior, and the histogram filter is switched off if $D_{KL} < \theta$. This will happen as observations accumulate and the posterior tightens. This ensures that the gamma filter only replaces $P(\lambda \mid \overrightarrow{s_i})$ with a single gamma density whenever $P(\lambda \mid \overrightarrow{s_i})$ resembles a gamma distribution. Hence, the general dissimilarity between the estimated and the true posterior $P(\lambda \mid \overrightarrow{s_i})$ is minimised, while unnecessary computation is minimised.

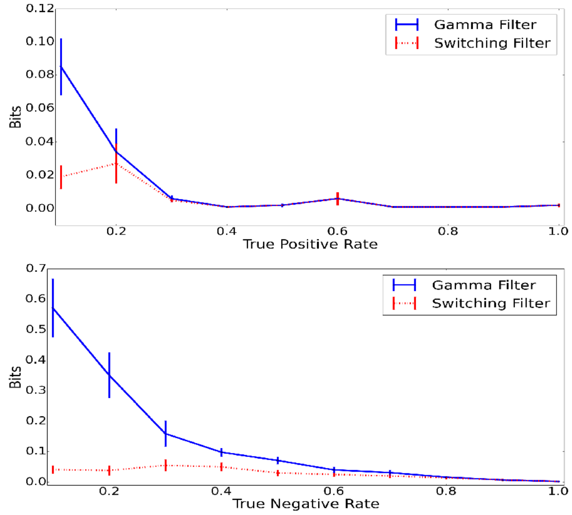


Figure 3: Average KL-divergence (over 60 trial of 144 observations) from the gamma and switching filters to the proxy ground truth posterior $P(\lambda \mid \overrightarrow{s_1}, \ldots, \overrightarrow{s_n})$. The horizontal axis shows the true positive rate (top) and true negative rate (bottom) of the simulated sensor. Standard error is shown.

5 Numerical evaluation of filter performance

To evaluate filter behaviour we first performed evaluations on synthetic data, such that we can precisely control sensor unreliability, and know the true λ and the true counts x_i of each sample.

5.1 Comparing filters to the true posterior

First, we measured the performance of a baseline case with a single unreliable sensor. We measured the KL-divergence from the tested filters to the true posterior. Since the true posterior takes an exponential amount of time to compute this can only be achieved for very short sequences of samples from the Poisson, in our case seven samples. We tested three different levels of sensor unreliability: the case where the true positive rate (TPR) is 0.1 and the true negative rate (TNR) is 1.0; the reverse of this; and a case in the middle where TPR=TNR=0.5. The prior for all filters was $Gamma(\lambda \mid \alpha = 1.01, \beta = 0.01)$.

Figure 2 shows the divergence in bits. The histogram filter perfectly tracks the true posterior. The gamma

Table 1: Comparison of the update time for three filters at each observation. 100, 1000, and 10000 represents the number of bins (histogram filter) or epochs (gamma filter) used on each observation.

Method	Avg. updating time (std. dev)		
	100	1000	10000
Gamma	0.2s (0.03)	1.0s (0.33)	9.3s (3.62)
Switching	0.4 (0.46)	4.7 (3.24)	85.9s (71.39)
Histogram	2.0s (0.22)	16.4s (2.50)	338.3 (1.96)

filter has low divergence except in the case that the TNR is low. So, the gamma filter is a poor fit to the posterior when the sensor produces many false positives, but copes well with sensors with low sensitivity.

5.2 Filter behaviour on long sequences

Given that the histogram filter in practice tracks the true posterior with zero error, we can use it as a proxy ground truth posterior. This enables us to compare the behaviour of the gamma and switching filters on much longer sample sequences (up to 144 samples). Using numerical simulation we first measured the average KL-divergence from the gamma and switching filters to this proxy posterior. For each trial, we generated simulated counts $x_1, \ldots, x_{144}$ by random sampling according to a Poisson with $\lambda = 3$. These were fed to a simulated sensor that counted unreliably. We then recursively updated $P(\lambda \mid \overrightarrow{s_i})$ using the gamma filter and the histogram filter. We ran 60 trials, and show the average for each level of sensor unreliability in Figure 3.

The results clearly show that the switching filter produces a lower divergence, and that its advantage over the simple gamma filter improves as the sensor unreliability increases. Again, note that the gamma filter is adversely affected by a low TNR.

5.3 Computational efficiency

We examined the computation time per sample for each of the three filters. Table 1 displays how the runtime for the three filters grows, as a function of the control parameters of each filter. The measurements were tested on a laptop with Intel i7 core processor and 16 gigabytes of RAM. In general, the gamma filter is ten to thirty-seven times faster than the histogram filter. The switching filter sits in the middle. The standard deviation of the gamma-histogram is quite high due to the switching mechanism on each update. Note that, for all remaining evaluations in the paper, we set the epoch to 100 for the gamma filter, and used 1000 bins for the histogram filter. This combination brings down the average update time for the switching filter to slightly less than 4.655 seconds per sample.

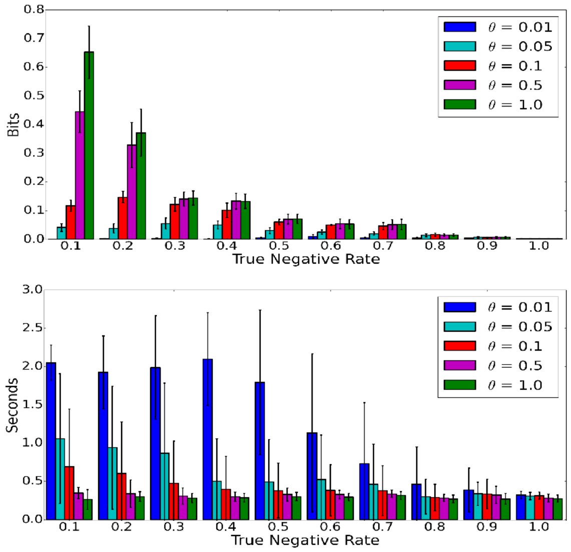


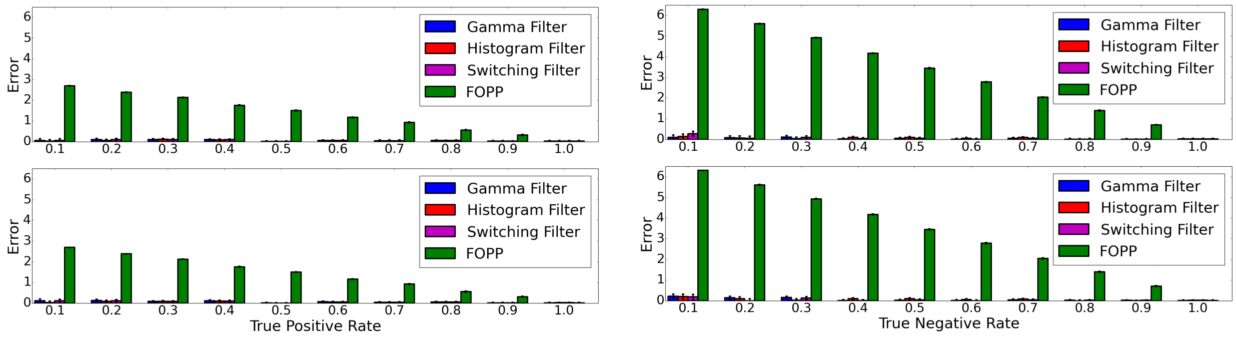
Figure 4: The divergence (top) and calculation time (bottom) of the switching filter for different θ values. In each graph the TNR of a single sensor is varied. Each trial consisted of a stream of $\overrightarrow{s_1} \ldots \overrightarrow{s_{144}}$ samples. Each data point is an average of 30 trials. Standard errors are shown.

5.4 θ selection of switching filter

θ is the acceptable KL-divergence in a single filtering step in switching filter from the true posterior (or the histogram filter as a proxy) to the approximation. The selected θ influences how low the KL-divergence is and how fast the computation time per sample is. In previous sections, we chose θ for the switching filter to be 0.05. In this subsection, we present a systematic study of the effect of θ on computation time and KL-divergence over long runs, and show why we chose 0.05 for θ. As shown in Figure 3, it is low TNR values that affect the gamma approximation most. Hence, we varied the TNR of a single sensor for this experiment.

In each trial, simulated counts $x_1, \ldots, x_{144}$ were generated by sampling a Poisson with $\lambda = 3$ and fed to the simulated sensor. $P(\lambda \mid \overrightarrow{s_i})$ was updated using only the switching filter with variation on θ. We ran 30 trials, and show the average divergence and calculation time for each level of TNR reliability in Figure 4.

Figure 4 shows how five different values for θ (0.01, 0.05, 0.1, 0.5 and 1.0) change the divergence and computation as the TNR changes. For, $\theta = 0.05$ the total bit divergence after a simulated run of 144 steps with a TNR of 0.1 and a single sensor is around 0.04 bits. We



(a) Estimation accuracy for λ while varying the TPR.

(b) Estimation accuracy for λ while varying the TNR.

Figure 5: The RMSE of posterior estimates of λ for different filters. In each graph the unreliability of a single sensor is varied. Each trial consisted of a stream of $\overrightarrow{s_1} \dots \overrightarrow{s_{144}}$ samples. Top panels show accuracy of the MAP estimate, and the bottom panels the accuracy of the expectation of the posterior. Each data point is an average of 30 trials. Standard errors are shown.

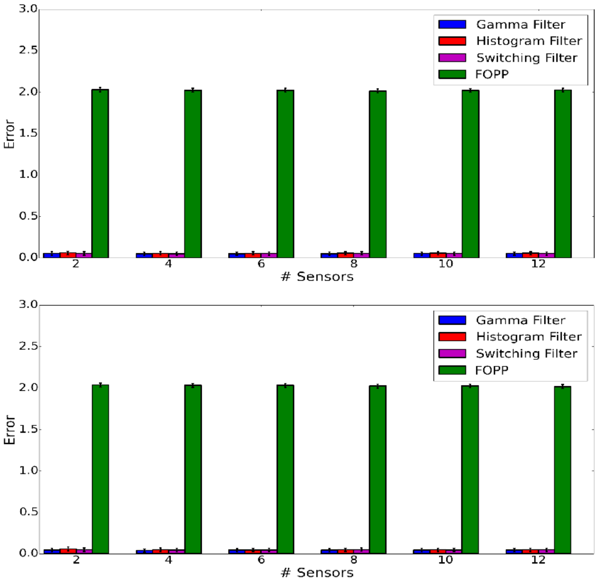


Figure 6: This shows the RMSE of different posterior estimates of the λ parameter as more sensors are added. Each trial consisted of a stream of $\overrightarrow{s_1} \dots \overrightarrow{s_{144}}$ samples. The top panel is the accuracy of MAP estimates, the bottom panel is the accuracies of expectation estimates. Each data point represents the average of 30 trials. Standard errors are shown.

can further reduce this by setting $\theta = 0.01$ to 0 bits, but in that case the switching filter only ever runs the histogram filter.

5.5 Accuracy of posterior estimates of λ

Finally, having characterised the quality of the posterior for each filter, we examined the difference that filtering according to a POPP model makes relative to filtering according a FOPP model.

To make this comparison, we evaluated the corrected estimate of the arrival rate λ of a Poisson process by applying Eqn. 1, using all three filters for the POPP model. We compared these against the uncorrected estimate using the standard Bayesian inference according to the FOPP model. We generated another series of simulated counts $x_1, \dots, x_n$ sampled from a Poisson process $P(x \mid \lambda')$.

In each experiment, we again sampled 144 counts from a Poisson process together with their corresponding sensor readings. We varied sensor specificity (true negative rate) while fixing sensor sensitivity (true positive rate) and vice versa. The performance of both POPP filters and a standard FOPP filter were assessed by comparing the RMSE of two estimators: the MAP hypothesis (mode) and the expectation (mean) of each posterior over λ to the true λ'. The results for a filter with a single sensor input are shown in Figure 5. This shows that all POPP filters generate much better estimates of λ than the FOPP filter. The FOPP filter's performance deteriorates linearly with declining sensor reliability.

Figure 6 shows the results for increasing numbers of unreliable sensors. We varied the sensor models in such a way that each pair of sensors have their sum of TPR and TNR equal to one (e.g. sensor A TPR=0.1, TNR=0.2, sensor B TPR=0.9, TNR=0.8). The figure shows that–given this balancing–the number of sensors has no effect on the accuracy of the estimates of λ for POPP model filters. Whereas the FOPP model filter is far from the true λ'.

6 Performance on a real world dataset

We also investigated the performance and practicality of the POPP model and its associated filters–versus

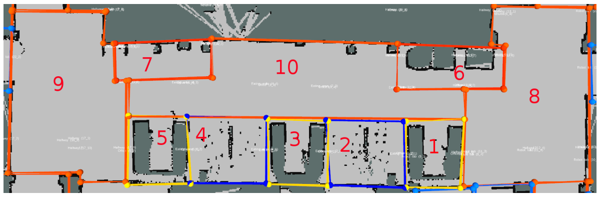


Figure 7: The environment in which the robot gathered data, with the regions marked.

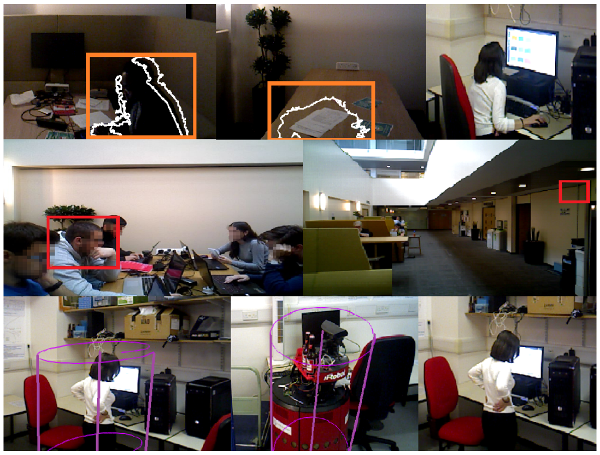


Figure 8: Correct and incorrect detections (and non-detections) from different regions in the environment for each sensor. Top row: change detector. Middle row: upper body detector. Bottom row: leg detector. Detections are marked with 2D or 3D bounding boxes.

Table 2: Averaged sensor model across all areas trained from 15 days of data.

Sensor	True Positive	True Negative
Leg	0.315	0.894
Upper body	0.266	0.853
Scenery change	0.611	0.820

the FOPP model–on a large, real world data set. This dataset was gathered from an office building in which a mobile robot counts the numbers of people passing by, as it patrols. The data set contains a time series of counts from three different automated person detectors (Dondrup, Bellotto, Jovan, & Hanheide, 2015). These use laser, depth camera and RGB information. We refer to them respectively as the leg detector (LD), upper body detector (UBD), and change detector (CD). Each returns a sensor count of the number of people it detected in each 10 minute interval during the day. These detectors are unreliable, as can be seen from Figure 8, which shows examples of correct and incorrect detections.

The data set was acquired during a 21-day deployment of the mobile robot. The detections are organised according to time/date and the spatial region where each detection was made. Since these observations were made while the robot patrolled the environment, the detections are temporally and spatially scattered. In addition, due to the robot's patrol policy the samples are not uniformly distributed across space.

By comparing the ground truth with the detections made by sensors, we calculated a sensor unreliability model for each region. An example of such a sensor model can be seen in Table 2. Although the robot operated 24/7, the sensor models were built using the data collected from 10am-8pm each week day, there being negligible detections outside these times. By limiting the robot to weekdays, we obtained 15 days of observations. We specified a time interval for the Poisson of 10 minutes, and recorded both the true counts and the detections made by each sensor in each interval. We estimated the parameter λ of the Poisson distribution by running a FOPP filter on the true counts.

The different POPP filters rely on sensor models that must be calculated from a confusion matrix relating true counts and the different sensor counts. To separate the training and testing data we performed five fold cross-validation with the unit being whole days, i.e., we used 3-days of data as a training set for a sensor model and then used the remaining 12-days of data as a test set on which to test the inferences made by each filter from the sensor counts.

For the 12 days of test data the different filters each made predictions of the λ parameter of the Poisson. Given this, we recorded the root mean squared error (RMSE) for each estimator compared to the true counts. Using this metric, we compared the performance of the POPP model, using the switching filter, to the standard Bayes' filter arising from the FOPP model. Each filter had to produce an estimated arrivals rate λ for each region within the patrol space.

The results are shown in Figure 9. As can be seen, the POPP model with the switching filter consistently produces better estimates than FOPP. The POPP filter's RMSE also varies less than that of the FOPP filter with respect to region.

In some areas (e.g. Region 4) the underlying sensor models for the POPP model were constructed from limited data. This resulted in a smaller gain in RMSE over the FOPP estimator for Region 4. We also show how the POPP and FOPP estimators evolved with time, again in terms of RMSE. The bottom panel of figure 9 shows that as time passes the performance of the POPP estimator steadily becomes better. This is

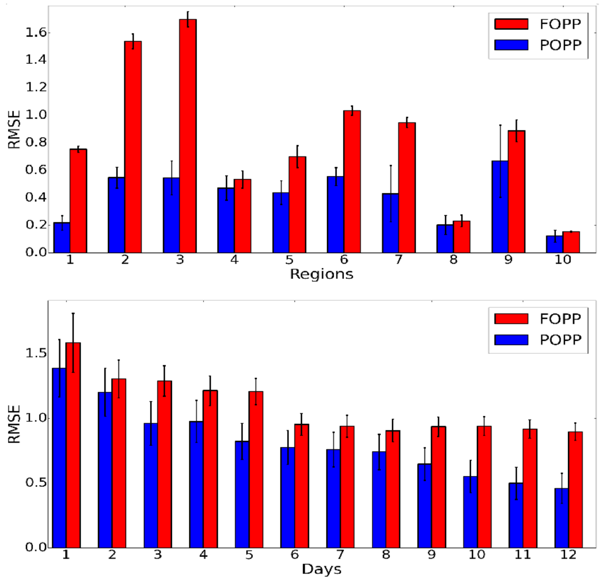


Figure 9: The top panel is the accuracy of estimators of λ derived from the POPP and FOPP filters. The bottom panel is the RMSE evolution from Day 1 to Day 12, averaged across all regions. Standard errors are shown.

shown by the growing gap between the RMSE of the two estimators.

7 Discussion and Conclusion

Estimation of counts from streaming data, generated by automated detectors, is an important problem. Currently, many studies in machine learning fail to address issues of any unreliability in their underlying data. When detectors are unreliable, this inevitably leads to significant under or over-counts. In this paper we showed how to compensate for unreliable count data from multiple sensors observing a single Poisson process. We have proposed a POPP model, presented three filters for it, and empirically investigated their properties. We have been able to show a significant improvement over the baseline FOPP model.

We constructed the model for multiple sensors. However, the model here assumes that sensor failures are conditionally independent from one other given the true count. Removing this assumption is our future work.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 600623, STRANDS.

References

Achcar, J. (2001). Bayesian analysis for software reliability data. *Advances in Reliability*, *20*(C), 733 - 748.

Achcar, J., Barrios, J., & Rodrigues, E. (2012). Comparing the adequacy of some non-homogeneous Poisson models to estimate ozone exceedances in mexico city. *JEP*, *3*(9A), 1213-1227.

Dondrup, C., Bellotto, N., Jovan, F., & Hanheide, M. (2015). Real-time multisensor people tracking for human-robot spatial interaction. In *ICRA WS on machine learning for social robotics.*

Guarnaccia, C., Quartieri, J., Barrios, J. M., & Rodrigues, E. R. (2014). Modeling environmental noise exceedances using non-homogeneous Poisson processes. *JAS*, *136*(4), 1631-1639.

Hawes, N., Burbridge, C., Jovan, F., Kunze, L., Lacerda, B., Mudrová, L., ... Hanheide, M. (2017). The STRANDS project: Long-term autonomy in everyday environments. *IEEE RAM*.

Hutchins, J., Ihler, A., & Smyth, P. (2007). Modeling count data from multiple sensors: A building occupancy model. In *Int. WS on comp. advances in multi-sensor adaptive processing.*

Ihler, A., Hutchins, J., & Smyth, P. (2006). Adaptive event detection with time-varying Poisson processes. In *IEEE KDD* (p. 207-216).

Ihler, A., Hutchins, J., & Smyth, P. (2007, Dec). Learning to detect events with Markov-modulated Poisson processes. *ACM TKDD*, *1*(3), 13:1-13:23.

Jovan, F., Wyatt, J., Hawes, N., & Krajník, T. (2016, Oct). A Poisson-spectral model for modelling temporal patterns in human data observed by a robot. In *IROS* (p. 4013-4018).

Ludkovski, M., & Sezer, S. O. (2012). Finite horizon decision timing with partially observable Poisson processes. *Stochastic Models*, *28*(2), 207–247.

Meier-Hellstern, K. (1987). A fitting algorithm for Markov-modulated poisson processes having two arrival rates. *EJOR*, *29*(3), 370 - 377.

Prabhu, N. U., & Zhu, Y. (1989). Markov-modulated queueing systems. *Queue. Sys.*, *5*(1), 215–245.

Rydén, T. (1996). An EM algorithm for estimation in Markov-modulated Poisson processes. *CSDA*, *21*(4), 431 - 447.

Scott, S. L. (1998). *Bayesian methods and extensions for the two state Markov modulated Poisson process* (Unpublished doctoral dissertation). Harvard.

Scott, S. L., & Smyth, P. (2003). The Markov modulated Poisson process and Markov Poisson cascade with applications to Web traffic data. *Bayesian statistics*, *7*, 671–680.

Supplementary

Here we provide detailed sensor models for each region. The sensor model was built from 15 days of data to give a general idea how each sensor performs across days. For the experiment itself, as we performed five fold cross validation, we only used 3 days of data to build the sensor model.

Table 3: Averaged sensor model for each region trained from 15 days of data.

Region	Sensor	True Negative	True Positive
1	Leg	0.820	0.102
	Upper body	0.749	0.244
	Scenery change	0.760	0.612
2	Leg	0.991	0.655
	Upper body	0.862	0.691
	Scenery change	0.826	0.778
3	Leg	0.854	0.116
	Upper body	0.833	0.130
	Scenery change	0.780	0.687
4	Leg	0.896	0.180
	Upper body	0.967	0.227
	Scenery change	0.897	0.592
5	Leg	0.918	0.086
	Upper body	0.881	0.200
	Scenery change	0.877	0.957
6	Leg	0.964	0.351
	Upper body	0.929	0.143
	Scenery change	0.803	0.541
7	Leg	0.949	0.264
	Upper body	0.829	0.071
	Scenery change	0.939	0.090
8	Leg	0.889	0.473
	Upper body	0.791	0.360
	Scenery change	0.900	0.591
9	Leg	0.702	0.383
	Upper body	0.711	0.172
	Scenery change	0.591	0.673
10	Leg	0.956	0.537
	Upper body	0.973	0.423
	Scenery change	0.823	0.584