LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-483318

Network Traffic Monitoring Using Poisson Dynamic Linear Models

D. M. Merl

May 9, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-TR-483318

Network Traffic Monitoring Using Poisson Dynamic Linear Models

D. M. Merl*
Applied Statistics Group
Lawrence Livermore National Laboratory

May 9, 2011

Abstract

In this article, we discuss an approach for network forensics using a class of nonstationary Poisson processes with embedded dynamic linear models. As a modeling strategy, the Poisson DLM (PoDLM) provides a very flexible framework for specifying structured effects that may influence the evolution of the underlying Poisson rate parameter, including diurnal and weekly usage patterns. We develop a novel particle learning algorithm for online smoothing and prediction for the PoDLM, and demonstrate the suitability of the approach to real-time deployment settings via a new application to computer network traffic monitoring.

[*] To whom correspondence should be addressed: `dmerl@llnl.gov`.

1 Introduction

The development of suitable analytics for real-time analysis and monitoring of computer network traffic remains a challenging and open problem in applied statistics. Nonstationary Poisson process models have been applied in this problem area with some success [4, 6, 5, 7], however most previous work in this area has been focused on retrospective analysis techniques, and is therefore poorly suited for real-time deployment.

In this paper, we develop a flexible statistical framework for the analysis of count-valued time series. This framework is based on characterizing the evolution of the rate parameter of a nonstationary Poisson process with a Gaussian dynamic linear model (DLM) [9], thereby embedding the DLM within the Poisson process. There are several significant advantages to this approach. One advantage is that the underlying DLM can be specified so as to account for known sources of variation influencing the evolution of the rate parameter. In the context of computer network traffic, such sources of variation include multiscale cyclical effects (e.g. daytime and nighttime effects, weekday and weekend effects), effects due to the subset subset of applications currently in use on the machine, and others. Another key advantage is that the Poisson DLM (PoDLM) will inherit many of the same efficient updating recursions of the underlying DLM. As we will show, this enables extremely efficient sequential inference for the PoDLM, making it ideally suited for real-time deployment.

The remainder of this article is organized as follows. In Section 2 we provide the full hierarchical model specification for the PoDLM. In Section 3, we derive a new particle learning approach for performing online inference for the PoDLM. Section 4 describes an application of our modeling approach to a new computer network traffic data set. Finally, Section 5 summarizes our results and indicates directions for future research.

2 Model specification

In this section we describe a class of nonstationary Poisson processes derived by modeling the evolution of the log Poisson rate as a Gaussian dynamic linear model (DLM), thus effectively embedding the DLM within the Poisson process. We will refer to this approach as the PoDLM. This approach is an extension of methods described previously by Cargnoni et al [2] and Taddy [8], and builds upon the dynamic linear model framework described by West and Harrison [9].

The full model specification is as follows:

$$
\begin{aligned}
y_t|\lambda_t &\sim \text{Poisson}(y_t|\lambda_t) && (1)\\
\log(\lambda_t) &= \eta_t && (2)\\
\eta_t|F_t, \theta_t &\sim \text{N}(\eta_t|F_t'\theta_t, V) && (3)\\
\theta_t|G_t, \theta_{t-1} &\sim \text{N}(\theta_t|G_t\theta_{t-1}, W_t) && (4)
\end{aligned}
$$

In this formulation, Equations 3 and 4 represent the standard Gaussian DLM observation and evolution equations. Thus in the PoDLM, the log rate of the Poisson process evolves according to the DLM structure specified through $\{F_t, G_t, V, W_t\}$. For readers unfamiliar with the DLM framework, a brief overview is as follows. The DLM is a discrete time state space model, where the unobserved state vector θ_t evolves from time $t-1$ to t according to a possibly time-dependent evolution matrix G_t, plus a Gaussian noise term. This noise term is controlled by the "system" covariance matrix W_t. The log Poisson rate is then modeled as a linear combination of elements of this state vector, with possible time-dependent design vectors F_t, plus a Gaussian noise term. This second noise term is controlled by the "observational" variance parameters V.

Following standard results from West and Harrison [9], we can choose a DLM structure that decomposes the evolution of η into a first-order polynomial effect, and cyclical, or seasonal effects, according to the desired periodicity. A key novelty of this approach is the representation of the seasonal effects using a Fourier representation of the cyclical trend. This is accomplished for a cyclical trend of length p by choosing F_t and G_t as follows. If p is an odd-length cycle, then in block notation we have:

$$\{F, G\} = \left\{ \begin{pmatrix} 1 \\ E_2 \\ E_2 \\ \vdots \\ E_2 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & J_2(1,\omega) & 0 & \dots & 0 \\ 0 & 0 & J_2(1,2\omega) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & J_2(1,h\omega) \end{pmatrix} \right\} \quad (5)$$

where $E_2 = [1 \quad 0]'$, $\omega = 2\pi/p$, $h = (p-1)/2$ (the number of harmonics), and

$$J_2(1,\omega) = \begin{pmatrix} \cos(\omega) & \sin(\omega) \\ -sin(\omega) & \cos(\omega) \end{pmatrix}$$

Similarly, for an even-length cycle, F_t and G_t are defined as

$$\{F, G\} = \left\{ \begin{pmatrix} 1 \\ E_2 \\ E_2 \\ \vdots \\ E_2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & J_2(1,\omega) & 0 & \dots & 0 & 0 \\ 0 & 0 & J_2(1,2\omega) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & J_2(1,h\omega-\omega) & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 \end{pmatrix} \right\} \quad (6)$$

where now $h = p/2$. One benefit of the Fourier representation lies in the ease with which parameter identifiability constraints can be enforced. An additional benefit lies in the ability to perform significance testing on individual harmonics in order to pare down the full Fourier representation to just those components with statistically meaningful predictive power. This pruning process can lead to increased computational efficiency by producing a model with far fewer parameters than the full representation.

Note that it is possible to further extend this basic model structure to represent different configurations of cyclical or design effects by augmenting the F and G parameters appropriately. This may necessitate time-specific F_t in order to represent different design effects associated with different time points; this poses no problem for the derivation of the particle learning approach below.

Finally, we assume that the "observational" variance V is unknown, with prior distribution $V^{-1} \sim \text{Gamma}(V^{-1}|n_0/2, d_0/2)$. We define the "system" variance using a discount factor approach where $W_t = (1-\delta)^2/\delta V$ for $\delta \in (0,1)$. The model is completed by specifying a prior mean for θ_0.

3 Sequential inference via particle learning

The goal of inference for the PoDLM is to produce estimates of the "hidden" state parameters, namely $\{\theta_t\}$ and the log Poisson rates $\{\eta_t\}$. As is common for Bayesian models such as this, inference is accomplished by producing samples from the posterior distributions of these parameters at each time step for the entire observation period. These distributions represent the uncertainty about the parameter value, conditional on all data observed up to the given time point. This sort of sequential or recursive inference process is ideal for this study because it allows one-step-ahead prediction of the yet-unobserved y_t. The next section will showcase this capability as a useful mechanism for anomaly detection.

An attractive feature of Gaussian DLMs is the availability of closed form, recursive updating equations for conducting sequential, on-line inference. Thus, conditional on the η parameters, the PoDLM will benefit from the same efficient updating strategy as the standard Gaussian DLM. Following the particle learning approach developed by Carvalho et al [3], and adapted for a simpler version of the Poisson DLM described by Taddy [8], we define a *resample-propagate* sequential Monte Carlo algorithm for producing draws from the joint posterior distribution of model parameters at time t as follows.

Fundamental to the particle learning approach is the notion of "sufficient statistics" for the posterior distribution at time t, denoted z_t. The term is in quotations because we will use a loose interpretation in which "sufficient statistics" refers to both the standard sample moments as well as any auxiliary variables required for posterior sampling. In this sense z_t can be thought of as concise way of representing the target distribution from which we would like to produce samples. Of course this vector z_t is itself a random variable, and therefore the aim of the particle learning algorithm is to maintain a set of samples from the current posterior distribution of $p(z_t|y^t)$ (the superscript denotes the collection of all subscripted values up to the superscripts, i.e. $y^t = \{y_1, y_2, \ldots, y_{t-1}, y_t\}$). The distribution $p(z_t|y^t)$ will be approximated in the usual way as $p(z_t|y^t) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{z_t^{(i)}}(z_t)$. In other words, the distribution is represented using an N sample Monte Carlo approximation where each particle is characterized by a distinct $z_t^{(i)}$.

For the PoDLM, we have $z_t = \{\eta_t, s_t\}$, where η_t is defined as above and treated as an auxiliary variable, and s_t denotes the sufficient statistics pertaining to the embedded Gaussian DLM (i.e. those sufficient statistics for the current distribution of θ_t and V). Thus, assuming we have a set of samples/particles representing $p(z_t|y^t)$, our goal is to produce samples/particles representing $p(z_{t+1}|y^{t+1})$. Following the strategy described in [3], we have:

$$\begin{aligned} p(z_{t+1}|y^{t+1}) &= p(\eta_{t+1}, s_{t+1}|y^{t+1}) && (7) \\ &= \int p(\eta_{t+1}, s_{t+1}|z_t, y^{t+1}) p(z_t|y^{t+1}) \mathrm{d}z_t && (8) \\ &= \int p(s_{t+1}|\eta_{t+1}, s_t) p(\eta_{t+1}|s_t, y_{t+1}) p(z_t|y^{t+1}) \mathrm{d}z_t && (9) \end{aligned}$$

In Equation 9, we make use of the conditional independences between s_{t+1} and y_{t+1} given η_{t+1}, and between η_{t+1} and y^t given s_t. Thus given a particle representation of $p(z_t|y^{t+1})$, we would be able to produce samples from the target distribution by

1. Sampling a value of η_{t+1} from $p(\eta_{t+1}|s_t, y_{t+1})$.

2. Updating the [recursively defined] sufficient statistics s_{t+1} given the newly sampled η_{t+1} and the previous s_t.

In order to produce a particle representation of $p(z_t|y^{t+1})$, we follow Carvalho et al [3] and observe that

$$p(z_t|y^{t+1}) \quad \propto \quad p(y_{t+1}|z_t)p(z_t|y^t) \tag{10}$$

where

$$p(y_{t+1}|z_t) \quad = \quad \int p(y_{t+1}|\eta_{t+1})p(\eta_{t+1}|z_t)\mathrm{d}\eta_t \tag{11}$$

Equation 10 implies that given a uniformly weighted particle approximation to $p(z_t|y^t)$, we can produce a uniformly weighted sample from $p(z_t|y^{t+1})$ by resampling with replacement from the original particles, with weights proportional to Equation 11.

The primary difficulty in both the propagate and resample steps lie in evaluating the integral specified in Equation 11. This integral appears in the propagate step in that

$$p(\eta_{t+1}|s_t, y_{t+1}) \quad = \quad p(y_{t+1}|\eta_{t+1})p(\eta_{t+1}|s_t)/p(y_{t+1}|s_t). \tag{12}$$

In the case of the Poisson DLM, Equation 11 is the convolution of a Poisson density $(p(y|\eta)$ with a log-normal density $(p(\eta|z))$. This integral was studied previously by Aitchison and Ho [1] in the context of deriving a Poisson-log normal distribution. Although analytic solution of the integral is not available, since generation of samples from the Gaussian density $p(\eta_{t+1}|z_t)$ is so trivial, a Monte Carlo approximation of the integral is easily obtained.

Furthermore, suppose $\{\eta_{t+1}^{(i)}\}$ are the samples produced from $p(\eta_{t+1}|z_t)$ and used to evaluate the resampling weight $c \approx \frac{1}{N}\sum_i p(y_{t+1}|\eta_{t+1}^{(i)})$. These samples can be reused in order to accomplish the propagate step by resampling a new η_{t+1} from this population with weights proportional to $p(y_{t+1}|\eta_{t+1}^{(i)})$.

Propagation rules for the remaining DLM sufficient statistics (s_t) follow from standard results in West and Harrison [9].

In summary, the particle learning algorithm for the PoDLM is as follows:

1. For each particle $z_{t-1}^{(i)}$, evaluate the unnormalized resampling weights as $w_t^{(i)} = \frac{1}{N}\sum_j^N Pois(y_t|e^{\eta_j})$ for some N samples from $N(\eta_t^{(i)}|f_t^{(i)}, Q_t^{(i)})$ (the DLM predictive distribution of η_t for particle i).

2. Resample particle indices by sampling with replacement using weights proportional to $\{w_t^{(i)}\}$.

3. For each resampled particle, sample $\eta_t^{(i)}$ by sampling from the N predictive samples used in 1. with weights proportional to $Pois(y_t|e^{\eta_j})$.

4. Propagate the remaining sufficient statistics according to the DLM recursions.

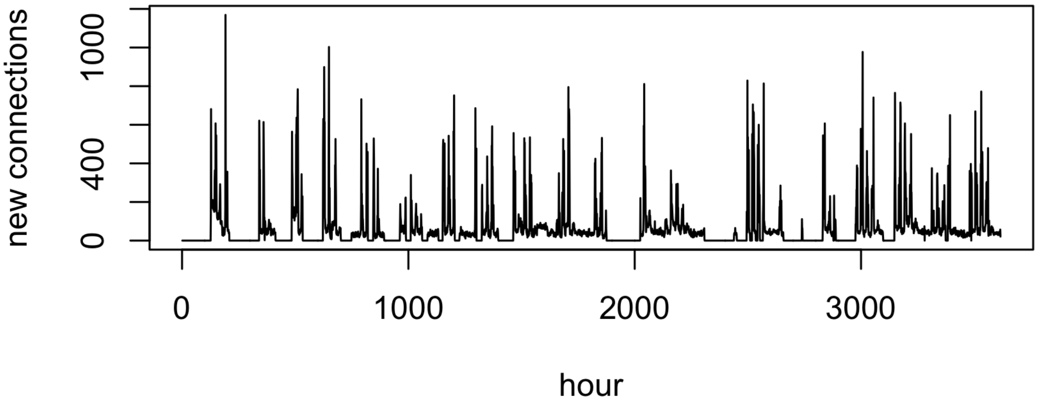

Figure 1: Total number of network connections generated per hour. Note the obvious periodicities in count incidence.

4 Application to network traffic modeling

In this section we demonstrate the Poisson DLM methodology in the context of modeling the rate of network traffic creation originating from a single host in a computer network monitored by host-based sensors. The purpose of this sort of analysis is to characterize the host-specific network usage patterns in order to identify situations where the amount of traffic originating from an individual machine differs significantly from historical patterns. Identification of such incidents plays a role in both malware discovery and network quality assurance.

Figure 1 shows the sequence of hourly network connection counts generated by one machine in a corporate network over the course of a $3,616$ hour ($\approx$ 144 day) monitoring period. This host was selected for demonstration purposes from among approximately 70 monitored machines due to the consistency with which its sensor reported count data. Although every effort was made to ensure reliable data collection, data collected via host-based sensors is intrinsically intermittent due to effects ranging from sensor software issues to the machine's owner powering down the machine during nights and weekends. As will be discussed below, a key feature of the PoDLM approach is its robustness to this type of intermittently missing/unobserved data.

In designing the structure of the embedded DLM, we must consider the possible sources of variation in the observable process. Firstly, it is reasonable to assume the presence of diurnal effects, owing to the differential usage patterns between workday and evening. Figure 2 demonstrates this by portraying separate time series for each time-of-day for the target host. The key observation here is the increased variability in connection count during the 8am-6pm window, as compared to the interval 6pm-8am. This suggests the need for at least a workday effect, if not a 24-hour cyclical effect.

Along the same lines, it is reasonable to assume a decrease in variability during weekends in

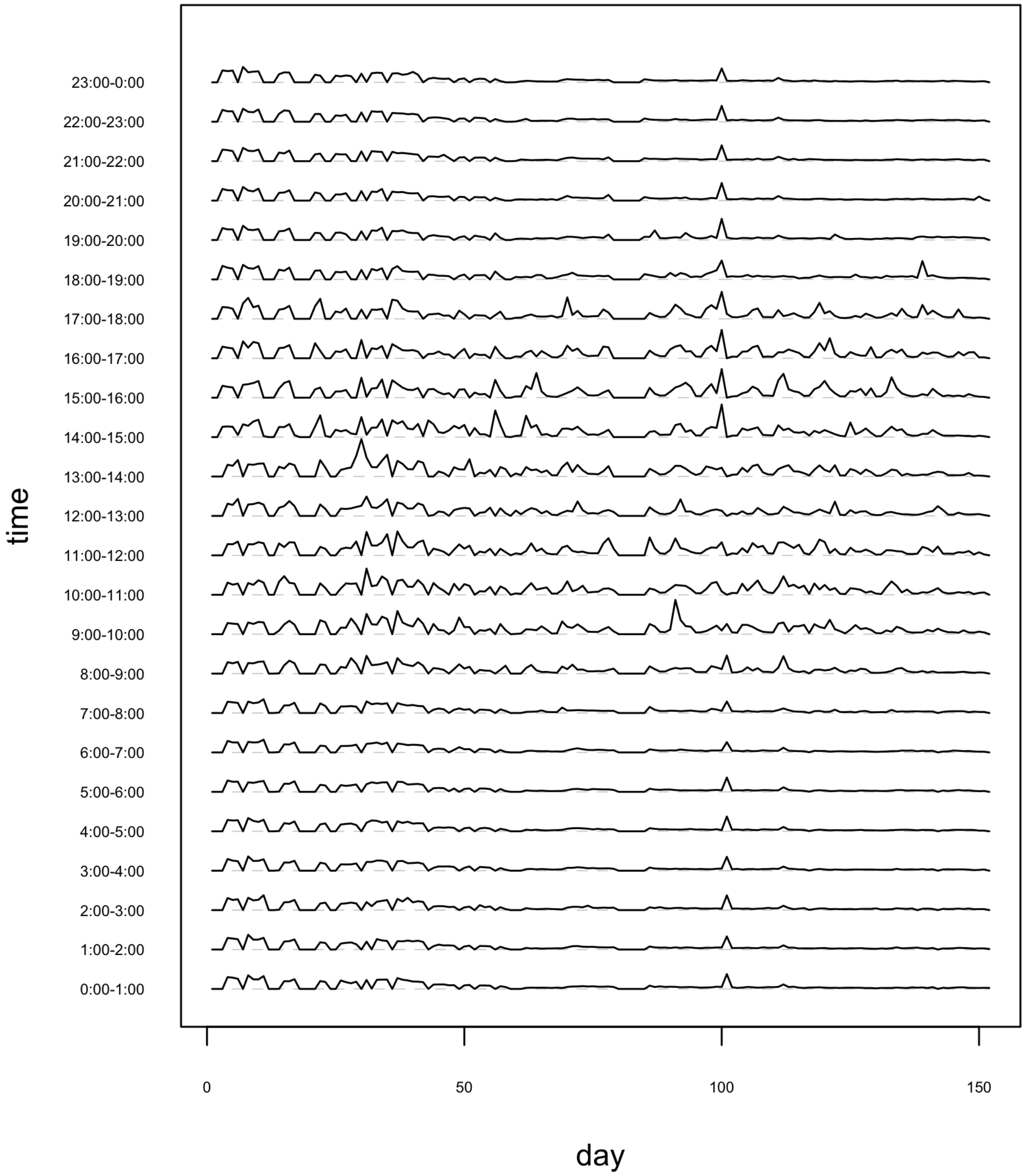

Figure 2: Hourly totals for separated by time-of-day. Note the consistency in the hourly time series for non-work hours (18:00-08:00) as well as for work hours (08:00-18:00). This suggests the need for seasonal effects on at least a 24-hour cycle.

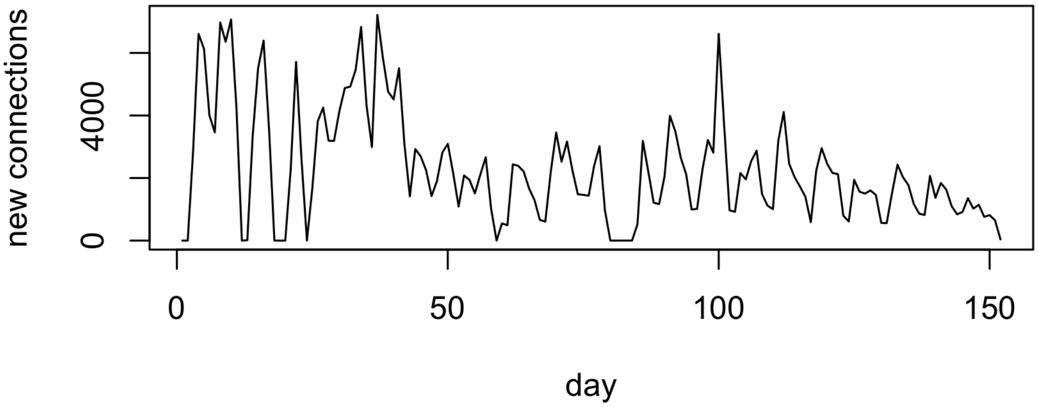

Figure 3: Total number of network connections generated per day.

contrast to weekdays. Figure 3 and 4 separate the day-specific connection counts for the target host. Although Figure 3 contains some evidence of the periodic drops in daily connection counts, Figure 4 does not provide much evidence that these periodic decreases are due to decreased weekend activity.

This leads us to define a simple embedded DLM characterized by a first order polynomial effect, and a 24-hour cyclical effect. The cyclical effect is represented in terms of its Fourier harmonics, and so the model conforms to the basic structure defined in Equation 6. The model is completed by specifying $\delta = 0.95$, $m_0 = \mathbf{0}$ (the prior mean for θ_0, and $n_0 = 2$, $d_0 = 2$. The particle learning algorithm defined above is then applied to the host connection count time series, using $N = 200$ particles.

Figure 5 shows the separation of the underlying Poisson process log rate time series into distinct effects. Each of the panels in Figure 5 represents the mean of the particular effect at each time point, averaged over each particle in the particle filter. The top panel shows the first order polynomial effect. There is evidently some amount of periodic signal that has not been captured by the cyclical effect manifesting itself in this first order effect. The middle and bottom panels represent the first 2 (of 12) harmonic effects. It is clear that the first harmonic dominates the others in terms of magnitude, and therefore the diurnal pattern in the underlying Poisson process would likely be well modeled by a single harmonic.

Figure 6 summarizes the predictive ability of the model. Before processing each observation, we evaluated the central 90% region of the one-step-ahead forecast for the observation. This provides one basis for anomaly detection, via identification of arrivals that fall above or below this predicted region. Figure 6 suggests that this metric may be overly stringent, as indicated by the large number of observations that fall outside this interval. This is likely due to properties of the Poisson distribution, in particular the association (equality really) between the mean of the distribution

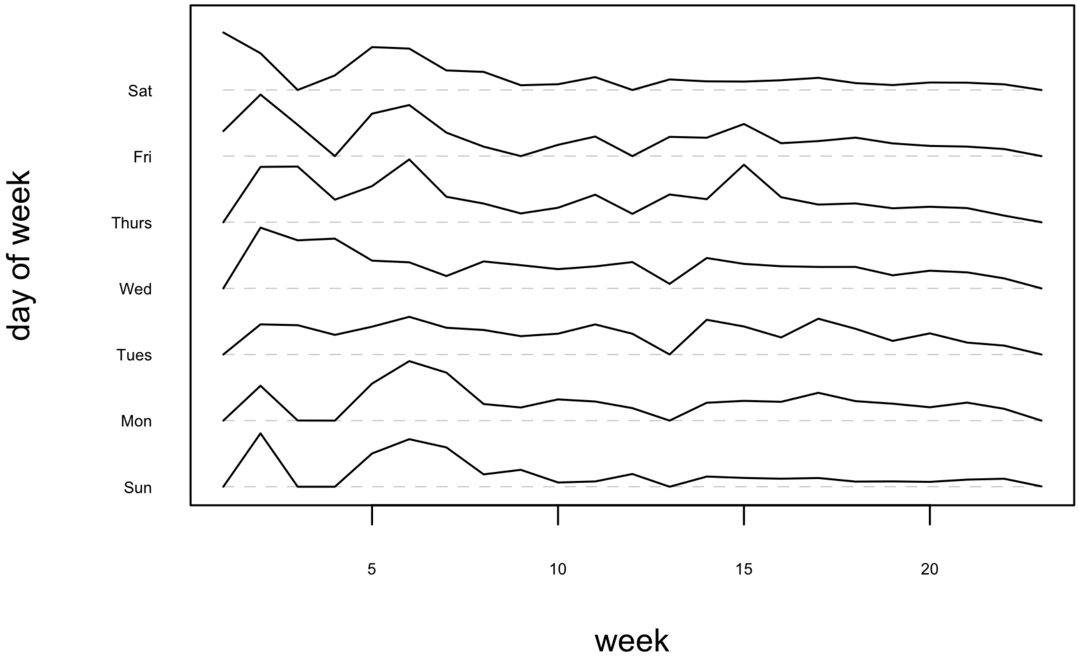

Figure 4: Daily totals separated by day-of-week. The data does not appear to provide support for cyclical trends on the weekly scale.

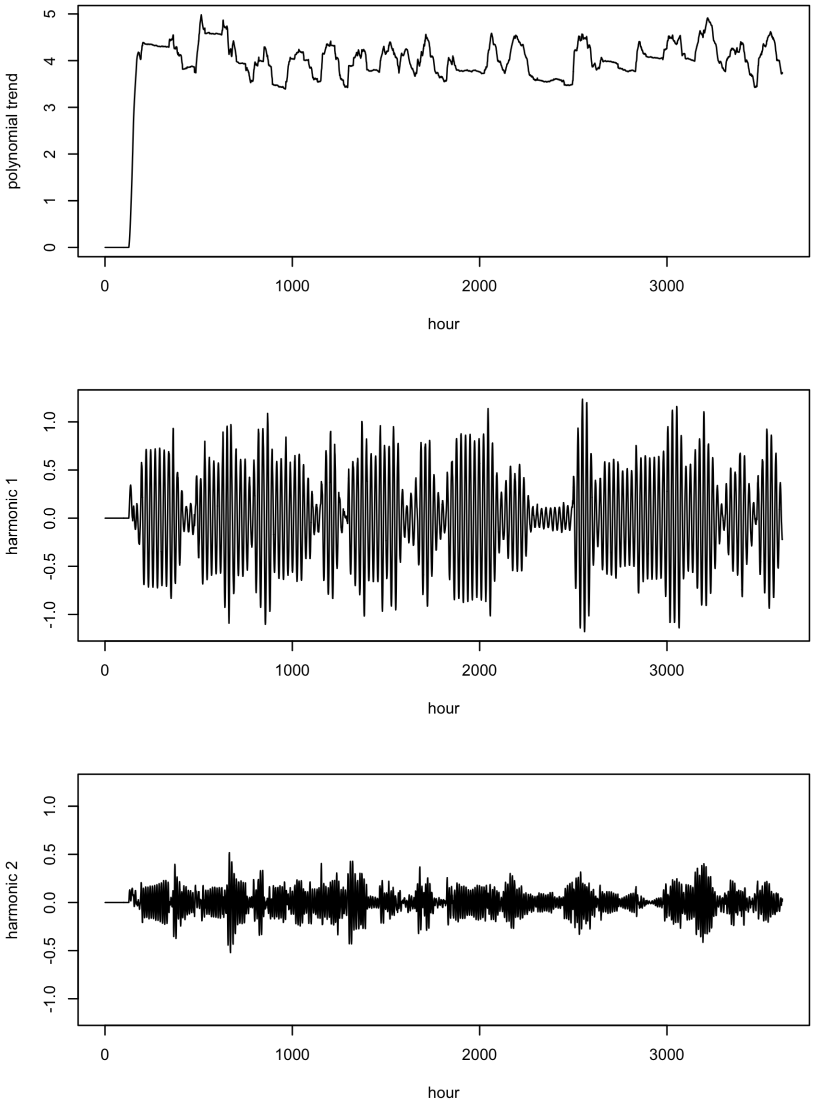

Figure 5: Trend decomposition of the Poisson process log rate parameter. The first panel shows the first order polynomial effect; the second and third panels show the first two Fourier harmonic effects. The large magnitude of the first harmonic effect indicates that the primary diurnal pattern is captured primarily by the effects appear to be captured primarily by the this effect.

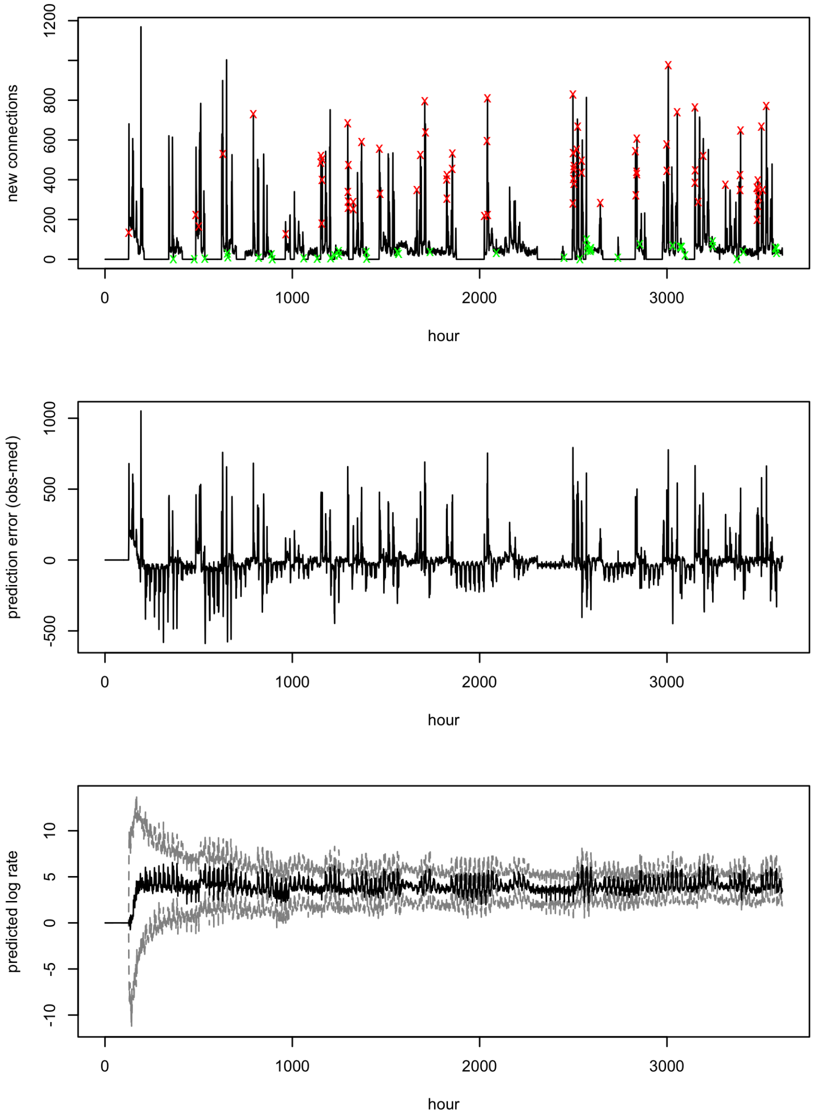

Figure 6: Predictive ability of the model. The first panel shows the observed connection count time series overlaid with indicators signifying that the observation exceeded the 95%-ile of the predictive distribution (red) or fell below the 5%-ile. The second panel shows the total prediction error, and the third panel shows the predicted 90% interval for the underlying Poisson process [log] rate parameter.

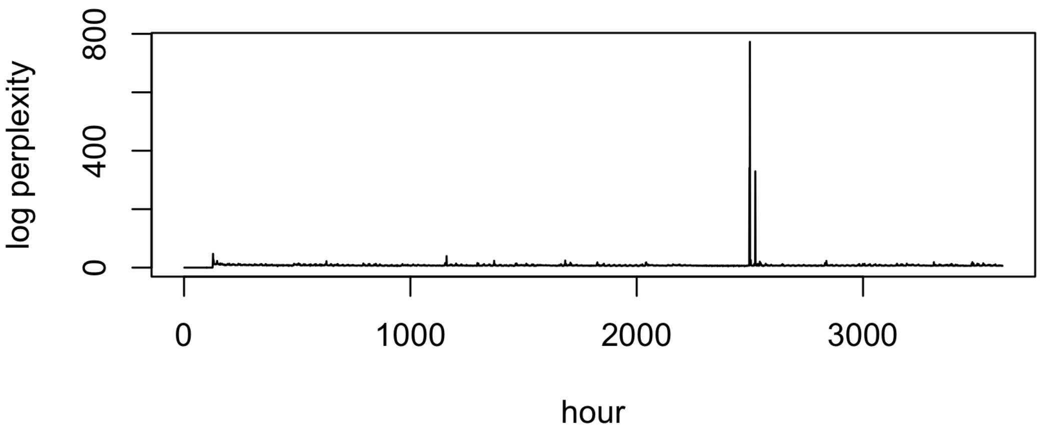

Figure 7: Log cloud " perplexity" for the target host.

and the variance.

For this reason, a better measure of anomaly in this setting can be derived by considering, informally, the level of surprise expressed by the particle cloud upon consideration of a new observation. This measure bears similarities to the usual "perplexity" score considered in text analysis. There, perplexity usually represents average log predictive density, averaged over multiple observations in a held out test sample. In our setting, a measure of perplexity is the average log predictive density, averaged over *particles*:

$$\log \rho_t = \frac{1}{N} \sum_{i=1}^{N} \log p(y_t | z_{t-1}^{(i)})$$

This measure has better properties with regard to the degree of dispersion of the underlying Poisson distributions.

Figure 7 shows the log perplexity score for each arrival over the course of the observation period. Viewed this way, there is a single significant incidents, and the incident occurring around hour $2,500$. Note that this spike in perplexity is coincident with a cluster of observations exceeding the predictive interval (Figure 6). Inspection of this second perplexity event revealed a significant and unprecedented spike in network activity in several Windows system processes, primarily attributable to a process called `RCGUI.EXE`, which presumably provides some type of remote desktop functionality. Figures 8 and 9 provide insight to the properties of the particle filter perplexity metric for a larger number of hosts. To produce Figures 8 and 9, we fitted the PoDLM specified above to network connection count time series for each of the top 25 most consistently reporting hosts in the sensor network. Figure 8 shows the one-step-ahead forecasting mismatches (analogous to Figure 6), and Figure 9 shows the particle filter perplexity time series (analogous to Figure 7). Evidently a perplexity-based anomaly detection would identify vastly fewer target events than a forecasting-based approach.

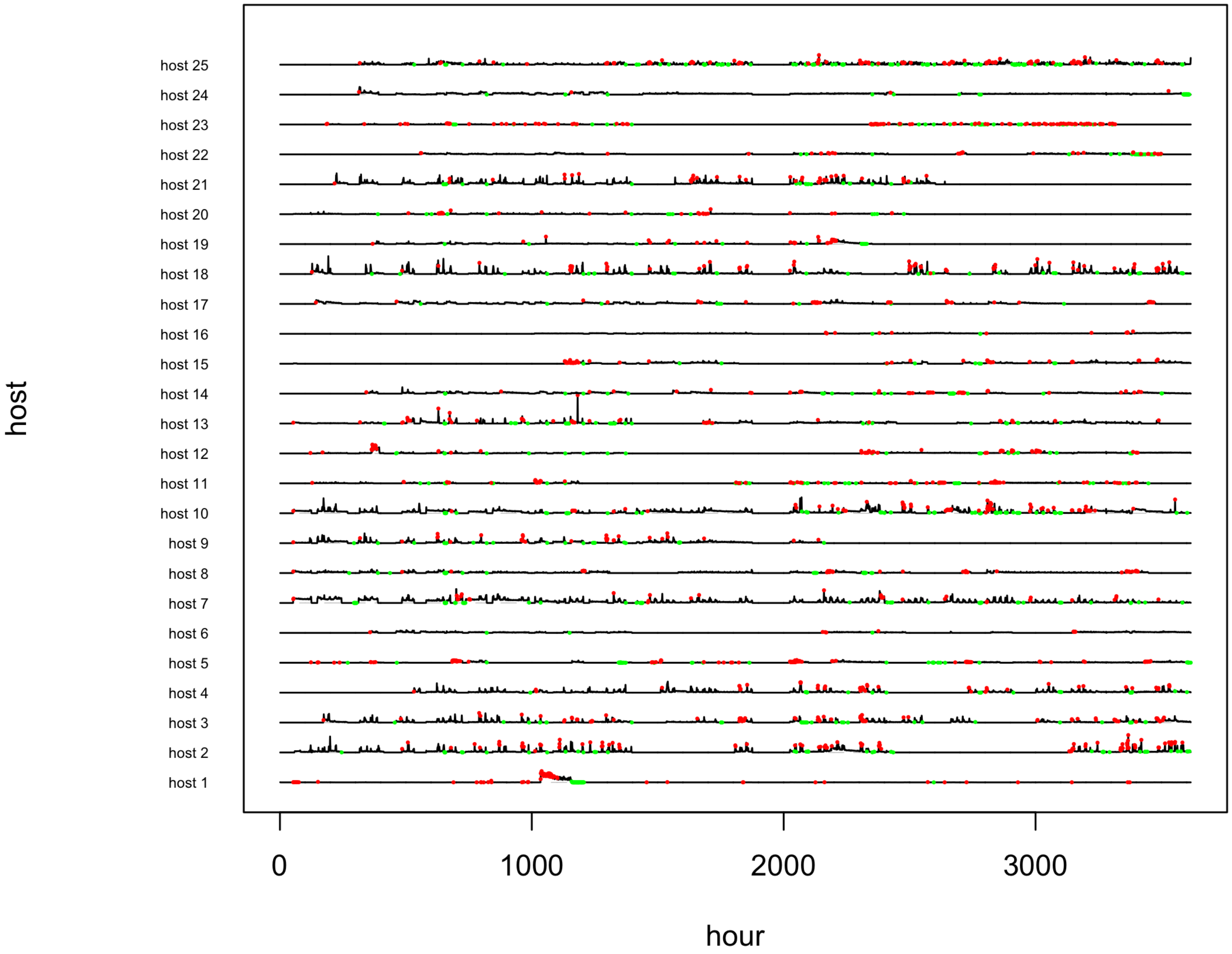

Figure 8: Anomaly detection on the basis of the one-step-ahead forecasting interval for the top 25 most active hosts in the sensor network.

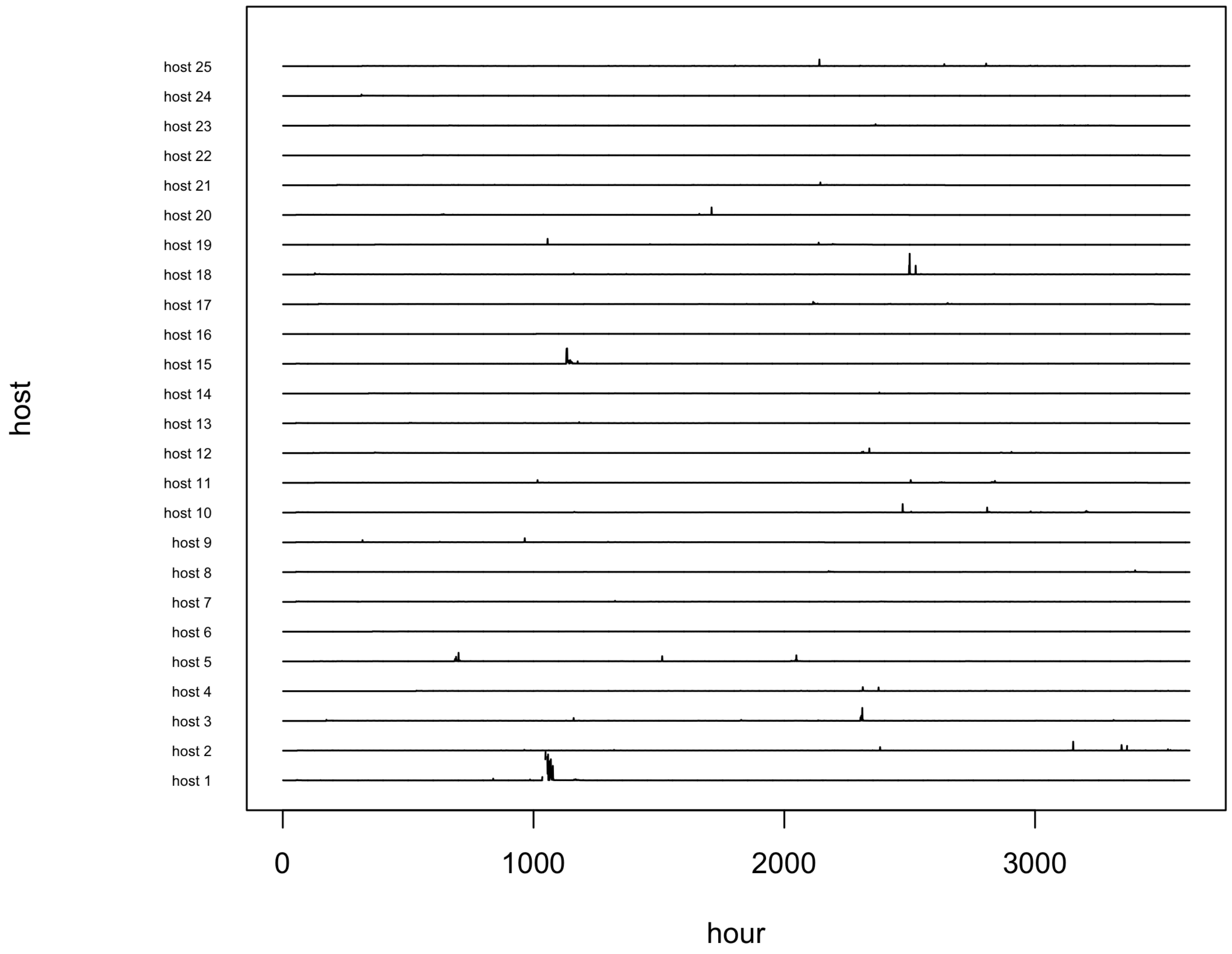

Figure 9: Perplexity per hour for the top 25 most active hosts in the sensor network.

Moreover, inspection of the top three highest-perplexity events in this data were all associated with reasonable, though ultimately benign, incidents. These top three events occurred in Host 1 (around hour 1,000), Host 15 (around hour 1,000), and Host 18 (around hour 2,500). Host 18 is the target host considered above. The event involving Host 1 was due to a single firefox process increasing its number of outbound network connections by an order of magnitude relative to the previous usage on that machine. The event involving Host 15 was associated with the first interactive use of that machine by a human user (the previous data collected from the host was entirely system-level/background processes). Although these particular incidents were themselves innocuous, they are characteristic of the sort of events that are commonly associated with malware and other illicit network usage.

5 Discussion

In this paper we have developed and demonstrated a novel statistical methodology for the analysis of count-valued time series. Our approach is based on modeling the evolutionary dynamics of a nonstationary Poisson process using an embedded Gaussian dynamic linear model. This approach facilitates the development of structured evolutionary models containing a wide range of possible effects, including polynomial, cyclical, and autoregressive effects. We enable full online inference on the underlying DLM model parameters through an efficient particle learning algorithm. The particle learning approach also provides a natural measure of anomaly via the particle filter perplexity score.

In anomaly detection situations, the primary difficulty of analysis lies in identifying a modest number of candidate anomalies. An analysis technique that produces an excessive number of anomalies is unlikely to find practical use due to the fact that the human analyst charged with verifying/investigating the events would be quickly overwhelmed. In our network traffic monitoring application, we were able to substantially reduce the number of identified events through the use of the particle filter perplexity score. In the approximately 4 months of data collection represented here, a perplexity-based anomaly detection scheme would identify only several significant events, whereas a forecasting-based approach would identify hundreds.

The modeling structure presented and used here represents only an initial strategy for realistic modeling of the forces driving the patterns of network usage. Possible extensions to our approach would include the addition of a weekday/weekend effect, process-based effects, and autoregressive effects (e.g. based on the value of the observable in the previous hour). These models will be considered and compared to the results presented here as this work continues.

References

[1] J. Aitchison and C. H. Ho. The multivariate Poisson-log normal distribution. *Biometrika*, 76(4):643–653, 1989.

[2] C. Cargnoni, P. Muller, and M. West. Bayesian forecasting of multinomial time series through conditionally Gaussian dynamic models. *Journal of the American Statistical Association*, 92(438):640–647, 1997.

[3] C. Carvalho, M. Johannes, H. Lopes, and N. Polson. Particle learning and smoothing. *Statistical Science*, 25(1):88–106, 2010.

[4] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. Adaptive event detection with time-varying Poisson processes. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 207–216, New York, NY, USA, 2006. ACM.

[5] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. Learning to detect events with Markov-modulated Poisson processes. *ACM Trans. Knowl. Discov. Data*, 1, December 2007.

[6] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary Poisson view of internet traffic. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1558–1569, march 2004.

[7] S. Scott. A Bayesian paradigm for designing intrusion detection systems. *Computational Statistics and Data Analysis*, 45:69–83, 2004.

[8] M. A. Taddy. Autoregressive mixture models for dynamic spatial Poisson processes: Application to tracking intensity of violent crime. *Journal of the American Statistical Association*, 105(492):1403–1417, 2010.

[9] M. West and J. Harrison. *Bayesian forecasting and dynamic models.* Springer, 1997.