Fine-grained Network Traffic Prediction from Coarse Data

Krzysztof Rusek
AGH University of Science and Technology
Institute of Telecommunications

Mathias Drton
Technical University of Munich,
Department of Mathematics

Abstract

ICT systems provide detailed information on computer network traffic. However, due to storage limitations, some of the information on past traffic is often only retained in an aggregated form. In this paper we show that Linear Gaussian State Space Models yield simple yet effective methods to make predictions based on time series at different aggregation levels. The models link coarse-grained and fine-grained time series to a single model that is able to provide fine-grained predictions. Our numerical experiments show up to 3.7 times improvement in expected mean absolute forecast error when forecasts are made using, instead of ignoring, additional coarse-grained observations. The forecasts are obtained in a Bayesian formulation of the model, which allows for provisioning of a traffic prediction service with highly informative priors obtained from coarse-grained historical data.

Keywords: network traffic prediction, state space model, Kalman filter, Bayesian structural time series.

1. Introduction

Modern ICT systems produce many different kinds of time series with information about computer network traffic. Network administrators often identify malfunction or Denial of Service Attacks upon visual inspection of the time series. Numerous open-source and commercial systems have been developed to support them in this task; e.g., Oetiker (2017a); Schnepp, Mathisen, Levsen, Pohl, and Stoeckel (2021). There has also been much interest in modeling network traffic data in order to make statistical predictions of future traffic. Indeed, the recent development of Software Defined Networking (SDN) allows one to use traffic predictions as one of multiple factors in optimization of network operation Jaglarz, Boryło, Szymański, and Chołda (2020).

A system optimizing network operation must be lightweight software with a small resource footprint. However, the traffic statistics it uses can quickly grow in size as the considered time horizon increases. Due to storage limitations, it is thus common practice to keep only the most recent data in full resolution (for at most a few days). Older observations are averaged over progressively larger time scales. Such an aggregation scheme is, for instance, used by the industry standard, high-performance data logging software *RRDtool* Oetiker (2017b).

For visual inspection by humans, the progressive decrease in resolution for older observations does not pose a significant problem. However, nonuniform resolutions present a challenge for automated quantitive predictions as used in the aforementioned SDNs. In this paper, we propose a general framework to address this challenge using a single Linear Gaussian State Space Model (LG-SSM). Specifically, this paper gives 1. a tractable LG-SSM that draws on fine- and coarse-grained data to obtain fine-grained predictions, 2. an implementation in a Bayesian framework that assesses uncertainty in prediction, and 3. a numerical evaluation of the resulting system based on real *mrtg* network traffic with a focus on longer time scales (hours to months). We note that our approach is not a classical fractal-based multi-scale model of the traffic Abry, Baraniuk, Flandrin, Riedi, and Veitch (2002).

To the authors' best knowledge, the related domain literature on analysis of network traffic does not propose direct competitors to the methods developed in this paper. In a broader context, work such as Folia and Rattray (2018) addresses aggregation issues but does so with a view towards arbitrary time scales, for which these methods adopt more complicated time discretization and resampling schemes for continuous-time stochastic processes. This is in contrast with the applications considered here, where there is a clear finest time scale that is discrete and where aggregation is done in a designed and regular fashion to address storage limitations. The aggregation schemes we consider also differ from down-sampling schemes in monitoring software as implemented in the MATLAB function *d2d*. Indeed, down-sampling creates thinned time series whereas we are here considering a system that retains averages.

The LG-SSM we propose is simple when compared to more involved machine learning methods for traffic prediction Andreoletti, Troia, Musumeci, Giordano, Maier, and Tornatore (2019). However, at least in their current design, these graph neural network methods are limited in three aspects: 1. the methods are limited to a single time scale and do not address our question about benefits of aggregated data, 2. they require order of magnitude larger dataset than offered by the *mrtg* tool, 3. uncertainty in neural networks is hard to estimate and it is a contemporary research topic Osband, Wen, Asghari, Ibrahimi, Lu, and Roy (2021). . Moreover, in practice, a simple Fourier decomposition Rzym, Boryło, and Chołda (2020) of the traffic data can help to obtain good forecast accuracy and performance from simple resource-aware models. A more advanced statistical forecast models are commonly based on ARIMA e.g. Zare Moayedi and Masnadi-Shirazi (2008); Papagiannaki, Taft, Zhang, and Diot (2003) or ARMA Sang and qi Li (2000). Since both the aforementioned models have state-space representation it is possible to extend our proposal to that kind of prediction model. Having said that, in this work, we focus on structured time series as the decomposition improves the explainability of the prediction. The main contribution of this paper is not a new forecasting method but rather a method for improved estimation of the classical models from limited data.

The paper is structured as follows. Section 2 outlines the data collection process in the considered network monitoring system, and it develops our joint model for fine and coarse data. Section 3 presents the setup and the results of our numerical evaluation on traffic in a university network. Section 4 summarizes the work, highlights some of its limitations, and suggests simple changes in monitoring systems that would help overcome these limitations.

2. Methods

The Multi Router Traffic Grapher (MRTG Oetiker (2017a)) is an open-source software for monitoring network link load. It gathers statistics by pooling SNMP counters from network devices. This process is continuous and while new measurements arrive the old ones are aggregated. The default sampling interval is 5 min. There are only 600 such measurements, which cover about 2 days. Older results are averaged in groups of six, which corresponds to 30 minute intervals. There are again 600 such measurements, which cover about 12 days. Measurements older than two weeks (2 days + 12 days) are aggregated in 2h windows. There

are again 600 of them covering 50 days. The remaining samples are averaged over 24 hours and give a coarse-grained traffic description over 2 years.

2.1. Linear Gaussian state space model

In order to model the described time series we adopt the framework of Linear Gaussian State Space Models (LG-SSM). In general, an LG-SSM may be described through a dynamical system Murphy (2012):

$$z_t = \boldsymbol{F} z_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{Q}), \tag{1}$$

$$x_t = \boldsymbol{H} z_t + \boldsymbol{\delta}_t, \quad \boldsymbol{\delta}_t \sim \mathcal{N}(\boldsymbol{c}, \boldsymbol{R}), \tag{2}$$

where $z_0 \sim \mathcal{N}(\boldsymbol{b}_0, \boldsymbol{Q}_0)$ is the n-dimensional initial hidden state and $\boldsymbol{x}_t \in \mathbb{R}^m$, $t \geq 0$, is the observed signal. Throughout, $\mathcal{N}(\cdot, \cdot)$ denotes a multivariate Gaussian distribution and the noise vectors $(\boldsymbol{\varepsilon}_t, \boldsymbol{\delta}_t)$ are independent across time t. The state dynamics are parameterized by the transition matrix $\boldsymbol{F} \in \mathbb{R}^{n \times n}$, the transition noise mean vector $\boldsymbol{b} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$. The observations x_t are noisy linear projections of the states z_t and further parameterized by the observation matrix $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ and the observation noise mean vector $\boldsymbol{c} \in \mathbb{R}^m$ and covariance matrix $\boldsymbol{R} \in \mathbb{R}^{m \times m}$. The system noise $\boldsymbol{\varepsilon}_t$, observation noise $\boldsymbol{\delta}_t$, and the initial state z_0 are independent. Under these assumptions, the Kalman filter can be used for efficient state estimation from noisy observations. This makes LG-SSM a popular choice for time series forecasting Harvey (1990).

For the particular application of interest, we follow Hu, Sim, Antoniades, and Dovrolis (2013) and specify an LG-SSM that incorporates three main components: trend, seasonal effect, and temporally dependent noise. The trend is modeled by introducing two real hidden states, namely, a current level l_t and a slope v_t. These are real-valued and evolve as

$$\begin{bmatrix} l_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_t^l \\ \varepsilon_t^v \end{bmatrix}, \tag{3}$$

where $\varepsilon_t^l \sim \mathcal{N}(0, \sigma_l^2)$ and $\varepsilon_t^v \sim \mathcal{N}(0, \sigma_v^2)$ are independent. This model allows for trend changes by a random normal perturbation. Next, we specify a harmonic signal $f_{t,\omega}$ of frequency ω, which can also be represented as an LG-SSM with known transition matrix with the help of an auxiliary variable $f_{t,\omega}^*$. Specifically,

$$\begin{bmatrix} f_{t,\omega} \\ f_{t,\omega}^* \end{bmatrix} = \begin{bmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{bmatrix} \begin{bmatrix} f_{t-1,\omega} \\ f_{t-1,\omega}^* \end{bmatrix} + \begin{bmatrix} \varepsilon_t^{f_\omega} \\ \varepsilon_t^{f_\omega^*} \end{bmatrix}. \tag{4}$$

Both components form a hidden state disturbed by independent noise representing phase fluctuations, $\varepsilon_t^{f_i^*}, \varepsilon_t^{f_i} \sim \mathcal{N}(0, \sigma_{f_\omega}^2)$. Typically the seasonal effect is not harmonic and it is approximated by a sum of multiple Fourier components for both weekly and daily seasonal effects (see Rzym *et al.* (2020) for details on how to choose the number of components). Finally, further temporal dependences are captured via a one-dimensional autoregressive (AR) process

$$d_t = \alpha d_{t-1} + \varepsilon_t^d, \tag{5}$$

where $\varepsilon_t^d \sim \mathcal{N}(0, \sigma_d^2)$. Summing the three components to one LG-SSM yields the *structural time series* model

$$x_t = l_t + \sum_\omega f_{t,\omega} + d_t + \varepsilon_t^x, \tag{6}$$

where $\varepsilon_t^x \sim \mathcal{N}(0, \sigma_x^2)$ is a total error Harvey (1990). In terms of the general representation from (1) we obtain a block-diagonal transition matrix $\boldsymbol{F}$, in which α is the only parameter. The variational inference method applied later ensures that $-1 < \alpha < 1$ as required for stationarity in (5). The observation matrix $\boldsymbol{H} = (1, 0, 1, 0 \dots, 1)$, with the zeros for states v_t

and $f_{t,\omega}^*$, performs the summation in (6). Note that in (1), $\boldsymbol{b} = 0$ and $\boldsymbol{Q}$ is diagonal. In (2), $\boldsymbol{c}, \boldsymbol{R}$ are one-dimensional with $\boldsymbol{c} = 0$.

2.2. Bayesian inference

In order to fit LG-SSM to the network traffic data of interest, we adopt a Bayesian approach Murphy (2012) that captures uncertainty via a posterior distribution. For our setting this approach has the advantage that today's posterior may play the role of tomorrow's prior distribution. The coarse-grained data may thus also serve for the purpose of refining prior distributions during the provisioning of the prediction service.

For Bayesian inference a joint prior distribution is to be specified for the initial states and the unknown model parameters. We assume all quantities to be independent a priori, with marginal normal distributions for the initial states, log-normal distributions ($\mathcal{LN}$) for the variance parameters, and a truncated normal distribution for the AR parameter α. The hyperparameters for these distributions are set using heuristics in TensorFlow Probability (TFP) Dillon, Langmore, Tran, Brevdo, Vasudevan, Moore, Patton, Alemi, Hoffman, and Saurous (2017) that aim to provide weakly informative prior distributions. In order to approximate the posterior distribution we apply Stochastic Variational Inference Murphy (2012) Given the parameters, the LG-SSM's predictions (forecasts) are normally distributed. However, in the Bayesian setup the normality is lost (in general the parameters are not not normally distributed) and the TFP library approximates forecast distributions by mixtures of Normal distributions obtained for a fixed number of posterior samples.

2.3. Aggregation to coarser time scales

Let $\boldsymbol{x}_t$ be a time series on a fine time scale. We are interested in leveraging information provided by aggregated versions of $\boldsymbol{x}_t$. Define the r-aggregated time series $\boldsymbol{x}_t'$ as the average of r consecutive values in non-overlapping windows of length r, that is,

$$\boldsymbol{x}_t' = \frac{1}{r} \sum_{i=0}^{r-1} \boldsymbol{x}_{rt+i}, \quad t = 0, 1, 2, \ldots. \tag{7}$$

Hereafter we use a $'$ to denote aggregated time series and their parameters and properties. Since r-aggregation is a linear operation, it retains Gaussianity of $\boldsymbol{x}_t$ for $\boldsymbol{x}_t'$. In fact, we also retain an LG-SSM as detailed in the following proposition, where we write $\bigoplus_{i=1}^{r} \boldsymbol{A}_i$ to denote a direct sum producing the block-diagonal matrix $\text{diag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_r)$.

Proposition 1. *Let $\boldsymbol{x}_t$ follow the LG-SSM from (1)-(2). Then the r-aggregated process $\boldsymbol{x}_t'$ follows an LG-SSM with rn states given by*

$$\boldsymbol{z}_t' = [\boldsymbol{z}_{rt}^T, \boldsymbol{\varepsilon}_{rt+1}^T, \ldots, \boldsymbol{\varepsilon}_{rt+r-1}^T]^T,$$

where $[\cdot]^T$ denotes matrix transposition. The transition and observation matrices are

$$\boldsymbol{F}' = \begin{bmatrix} \boldsymbol{F}^r & \boldsymbol{F}^{r-1} & \cdots & \boldsymbol{F} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{rn \times rn}, and \tag{8}$$

$$\boldsymbol{H}' = \frac{1}{r} \left[\sum_{i=0}^{r-1} \boldsymbol{H} \boldsymbol{F}^i, \sum_{i=0}^{r-2} \boldsymbol{H} \boldsymbol{F}^i, \ldots, \boldsymbol{H} \right] \in \mathbb{R}^{m \times rn}. \tag{9}$$

The initial state $\boldsymbol{z}_0'$ holds the original initial state and $r-1$ transition noises. The new system noise $\boldsymbol{\varepsilon}_t' = [\varepsilon_{rt}^T, \ldots, \varepsilon_{rt+r-1}^T]^T$ is constructed by stacking r system noise vectors, while the new observation noise $\boldsymbol{\delta}_t'$ is an average of the corresponding r observation noise vectors. They are distributed as

$$\boldsymbol{\varepsilon}_t' \sim \mathcal{N}\left([\boldsymbol{b}^T, \ldots, \boldsymbol{b}^T]^T, \bigoplus_{i=1}^{r} \boldsymbol{Q}\right), \quad \boldsymbol{\delta}_t' \sim \mathcal{N}\left(\boldsymbol{c}, \frac{1}{r}\boldsymbol{R}\right). \tag{10}$$

Proof. For clarity, the proof will be given for $r = 2$; cases with larger r are analogous. We have for $t \geq 1$,

$$\begin{bmatrix} \boldsymbol{x}'_{t-1} \\ \boldsymbol{x}'_t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{2(t-1)} \\ \boldsymbol{x}_{2(t-1)+1} \\ \boldsymbol{x}_{2t} \\ \boldsymbol{x}_{2t+1} \end{bmatrix}, \tag{11}$$

where $\boldsymbol{I}, \boldsymbol{0} \in \mathbb{R}^{m \times m}$ are the identity matrix and the matrix with all entries 0, respectively. Unrolling the dynamics from (1) and (2) we obtain that

$$\begin{bmatrix} \boldsymbol{x}_{2t-2} \\ \boldsymbol{x}_{2t-1} \\ \boldsymbol{x}_{2t} \\ \boldsymbol{x}_{2t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{HF} & \boldsymbol{H} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{HF}^2 & \boldsymbol{HF} & \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{HF}^3 & \boldsymbol{HF}^2 & \boldsymbol{HF} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{2t-2} \\ \boldsymbol{\varepsilon}_{2t-1} \\ \boldsymbol{\varepsilon}_{2t} \\ \boldsymbol{\varepsilon}_{2t+1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}_{2t-2} \\ \boldsymbol{\delta}_{2t-1} \\ \boldsymbol{\delta}_{2t} \\ \boldsymbol{\delta}_{2t+1} \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \boldsymbol{x}'_{t-1} \\ \boldsymbol{x}'_t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \boldsymbol{H} + \boldsymbol{HF} & \boldsymbol{H} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{HF}^2 + \boldsymbol{HF}^3 & \boldsymbol{HF} + \boldsymbol{HF}^2 & \boldsymbol{H} + \boldsymbol{HF} & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{2t-2} \\ \boldsymbol{\varepsilon}_{2t-1} \\ \boldsymbol{\varepsilon}_{2t} \\ \boldsymbol{\varepsilon}_{2t+1} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \boldsymbol{\delta}_{2t-2} + \boldsymbol{\delta}_{2t-1} \\ \boldsymbol{\delta}_{2t} + \boldsymbol{\delta}_{2t+1} \end{bmatrix},$$

which simplifies to

$$\begin{bmatrix} \boldsymbol{x}'_{t-1} \\ \boldsymbol{x}'_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}' & \boldsymbol{0} \\ \boldsymbol{H}'\boldsymbol{F}' & \boldsymbol{H}' \end{bmatrix} \begin{bmatrix} \boldsymbol{z}'_{t-1} \\ \boldsymbol{\varepsilon}'_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}'_{t-1} \\ \boldsymbol{\delta}'_t \end{bmatrix}. \tag{12}$$

Equation (12) defines a new state space model as

$$\boldsymbol{z}'_t = \boldsymbol{F}'\boldsymbol{z}'_{t-1} + \boldsymbol{\varepsilon}'_t,$$
$$\boldsymbol{x}'_t = \boldsymbol{H}'\boldsymbol{z}'_t + \boldsymbol{\delta}'_t. \qquad \qquad \square$$

3. Numerical experiment

Proposition 1 links observations at a different aggregation level ($\boldsymbol{x}'_t$) to the single underlying LG-SSM. This link allows the construction of a joint model for two or more time scales that, as we demonstrate now, yields improved network traffic predictions by leveraging aggregated historical information. In our Bayesian approach to statistical inference we work with simple vague priors that assume prior independence of parameters. Hyper parameters are obtained from data using heuristics from TensorFlow Probability (TFP) Dillon *et al.* (2017), as described in Section 2. For our experiments the following priors were obtained. At the finest time scale, the initial states l_0 and d_0 are normal with mean -219 and variance $5 \cdot 10^5$. The states v_0 and f_0 are centered normal with variances 10^5 and $3 \cdot 10^5$, respectively. The log-normal variances $\sigma_l, \sigma_v, \sigma_d, \sigma_{f_\omega}, \sigma_x$ are all derived from a normal distribution with variance 3. The means are 2.8 for the first three variances, and 1.2 for the rest. The AR parameter α is truncated standard normal, with truncation to $(-1, 1)$. These same priors are also used to induce priors for aggregated coarse-grained time series; recall Proposition 1, where the state z'_t combines a fine-scale initial state with system noise, the joint distribution of which is determined by the fine-scale priors and dynamics.

Our numerical experiment uses a real MRTG dataset from downlink traffic measurements at the first author's university. In the experiment, we use two non-overlapping sets of samples: 600 samples taken every 30 min (fine-grained) and 600 samples taken every 2h (coarse-grained), so in this case $r = 4$. While the model can use all four aggregation levels in the data, we focus here on only two of them because the practical solutions proposed in the literature are focused on a time scale of about 1h Jaglarz *et al.* (2020) and because we could substantially simplify our implementation for r being a power of two. The overall time span
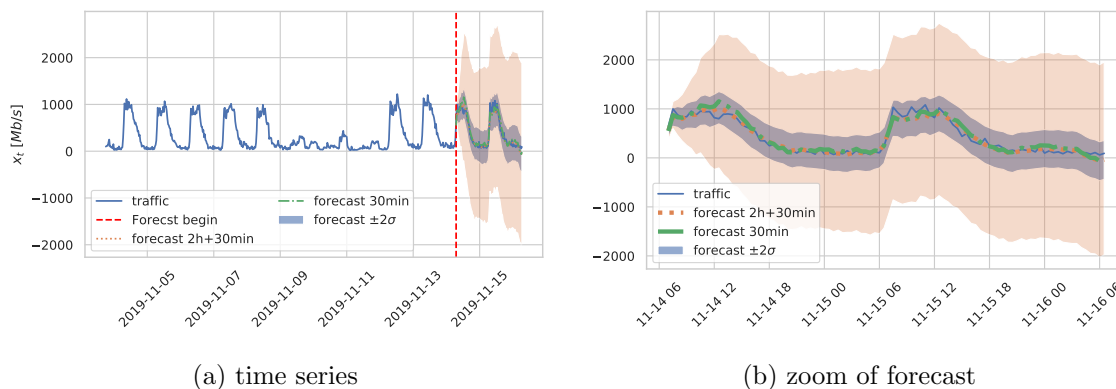
(a) time series

(b) zoom of forecast

Figure 1: Fine-grained traffic time series and prediction 1a. The orange ribbon represents forecast uncertainty for the fine-grained model. The blue ribbon shows improved uncertainty obtained from our joint model. Additional observations improve the likelihood of forecast (the likelihood of true future according to forecast distribution) by a factor of $e^{2.32} \approx 10$ (log-likelihood changed from 543.6 to 545.92) and expected mae over $\mathbf{3.7\times}$. Holiday: 11, weekend: 9 10 November.

then covers around 62 days, from which the last 2 days are used only for forecast validation. The data exhibits a strong seasonal component including both weekly and daily patterns (see Figure 1), which our model picks by inclusion of two periodic components (16 harmonics each). We note that since the fine-grained part of the series contains only one weekend and a holiday on Monday, it is difficult to estimate the weekly seasonal effect as well as long-range trend, as represented by (3), using only observations from one time scale. The forecast from the fine-grained model based on (6) is presented in Figure 1 and denoted as *forecast 30min*. The forecast distribution is a mixture of 50 equally probable Kalman prediction distributions obtained for 50 samples from the surrogate posterior in the variational approach.

Although the average is reasonably accurate, the forecast uncertainty is very large. This can be explained by the fact that the extra Monday breaks the periodic patterns and increases variability in the trend component. Addition of additional observations at the coarser 2h resolution substantially improves the prediction. The prediction interval decreases by more than a factor or three, and the likelihood of the forecast data increases about ten times.

Further intuition about the results can be provided by computing the expected mean absolute error, where the expectation is computed for estimated forecast distribution. In other words, this is the average error for all possible future values and thus it depends on the entire forecast distribution. The expectation computed with the Monte Carlo method using 100 samples (possible forecasts) yield the following values: $\mathrm{mae_{30min}} = \mathbf{602 \pm 36}$ Mb/s and $\mathrm{mae_{30min+2h}} = \mathbf{160 \pm 9}$ Mb/s. The use of historical data decreased mae over $\mathbf{3.7\times}$. We emphasize that this substantial improvement in model accuracy comes at a moderate computational cost.

As previously mentioned, the estimation and forecast were implemented using TensorFlow Probability Dillon *et al.* (2017). The implementation with additional numerical examples is openly available in Rusek (2021) as open-source software. The *LinearOperator* API provided by TensorFlow allows one to implement the model efficiently without instantiating the large block matrices from (8) and (9). Only small dense matrix multiplications are executed, which can be done efficiently via the associative scan procedure. Furthermore, our code was JIT-compiled using XLA – a TensorFlow linear algebra compiler and executed on GPU. We observed increased numerical precision of compiled (and optimized) computations compared to step-by-step execution. The computational cost of 300 variational steps of the estimation is about 2.5h on Tesla V100 SXM2 GPU. Most of the time is used for compilation (we observed a similar run time without JIT). However, as described earlier this cost is amortized by the fact that once we have determined the parameter posterior we can refine the model online

without lengthy computations.

4. Discussion and conclusions

In this paper we proposed a Bayesian structural time series model for network traffic measurements that feature different time scales. Our LG-SSM approach readily allows one to aggregate fine- versus coarse-grained observations. Our experiment shows that including older coarse-grained statistics on network traffic can drastically improve prediction accuracy and uncertainty. We are confident this observation would similarly hold in other modeling context, including for example single source of error models (aka exponential smoothing) Hyndman and Athanasopoulos (2018).

The data we consider were collected using the MRTG tool and come in the form of nonnegative measurements. The fact that we model these directly as Gaussian and with additive structure is a clear limitation of our work. Indeed, log-normal distributions have been shown to give more accurate descriptions of network traffic Alasmar, Parisis, Clegg, and Zakhleniu (2019). When working with a single time scale this issue can be addressed quite straightforwardly by applying our model to the log-transformed traffic data. With this transformation additive effects in the model would correspond to multiplicative effects on the original nonnegative scale, and thus more appropriately capture the fact that during the periods of highest traffic intensity (middle of the day) one observes higher fluctuations compared to nights when the traffic is smallest. Classical alternative to this transformation is GARCH model Zhou, He, and Sun (2006). However, with more than one time scale and aggregation of data the use of log-transformations or GARCH is more subtle because time aggregation by averaging is no longer on the log-scale. As a resolution of this problem we propose that software systems store geometric instead of arithmic means when aggregating older measurements of traffic size. This simple modification would make our model (and other possible LG-SSM) directly applicable to log-transformed data with multiple time scales.

We tested this proposal on the real traffic, see detail in the Appendix A. The experimental results support our claims for both log-transformed and non-scaled data. The reduction of prediction variance significantly improves predictions in the log-transformed domain. The application of log transform also reduces forecast error by eliminating unrealistic negative forecasts. We observed a 4% reduction of mae, solely due to forecast in the log domain. This is a clear indication that when it comes to network traffic, the geometric average is more informative compared to arithmetic aggregation.

5. Acknowledgments

References

Abry P, Baraniuk R, Flandrin P, Riedi R, Veitch D (2002). "Multiscale Nature of Network Traffic." *IEEE Signal Processing Magazine*, **19**(3), 28–46. doi:10.1109/79.998080.

Alasmar M, Parisis G, Clegg R, Zakhleniu N (2019). "On the Distribution of Traffic Volumes in the Internet and Its Implications." In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 955–963. doi:10.1109/INFOCOM.2019.8737483.

Andreoletti D, Troia S, Musumeci F, Giordano S, Maier G, Tornatore M (2019). "Network Traffic Prediction Based on Diffusion Convolutional Recurrent Neural Networks." In *IEEE*

INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFO-COM WKSHPS), pp. 246–251. `doi:10.1109/INFCOMW.2019.8845132`.

Dillon JV, Langmore I, Tran D, Brevdo E, Vasudevan S, Moore D, Patton B, Alemi A, Hoffman M, Saurous RA (2017). "TensorFlow Distributions." `doi:10.48550/arXiv.1711.10604`. `1711.10604`.

Folia MM, Rattray M (2018). "Trajectory Inference and Parameter Estimation in Stochastic Models with Temporally Aggregated Data." *Statistics and Computing*, **28**(5), 1053–1072. `doi:10.1007/s11222-017-9779-x`.

Harvey AC (1990). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press. ISBN 9781107717145. `doi:10.1017/CBO9781107049994`.

Hu K, Sim A, Antoniades D, Dovrolis C (2013). "Estimating and Forecasting Network Traffic Performance Based on Statistical Patterns Observed in SNMP Data." In P Perner (ed.), *Machine Learning and Data Mining in Pattern Recognition*, pp. 601–615. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-39712-7. `doi:10.1007/978-3-642-39712-7_46`.

Hyndman R, Athanasopoulos G (2018). *Forecasting: Principles and Practice*. 2nd edition. OTexts, Australia. Accessed: 2021-02-04.

Jaglarz P, Boryło P, Szymański A, Chołda P (2020). "Enhanced Lagrange Decomposition for Multi-objective Scalable TE in SDN." *Computer Networks*, **167**, 106992. ISSN 1389-1286. `doi:10.1016/j.comnet.2019.106992`.

Murphy KP (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series. MIT Press. ISBN 9780262018029.

Oetiker T (2017a). "MRTG — The Multi Router Traffic Grapher." `https://oss.oetiker.ch/mrtg/`. URL `https://oss.oetiker.ch/mrtg/`.

Oetiker T (2017b). "RRDtool — About RRDtool." `https://oss.oetiker.ch/rrdtool/`. URL `https://oss.oetiker.ch/rrdtool/`.

Osband I, Wen Z, Asghari M, Ibrahimi M, Lu X, Roy BV (2021). "Epistemic Neural Networks." `doi:10.48550/arXiv.2107.08924`. `2107.08924`.

Papagiannaki K, Taft N, Zhang ZL, Diot C (2003). "Long-term Forecasting of Internet Backbone Traffic: Observations and Initial Models." In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 2, pp. 1178–1188 vol.2. `doi:10.1109/INFCOM.2003.1208954`.

Rusek K (2021). "Algae: Aggregated Linear Gaussian State Space Models." URL `https://github.com/krzysztofrusek/Algae`.

Rzym G, Boryło P, Chołda P (2020). "A Time-efficient Shrinkage Algorithm for the Fourier-based Prediction Enabling Proactive Optimisation in Software-defined Networks." *International Journal of Communication Systems*, **33**(12), e4448. `doi:10.1002/dac.4448`. E4448 dac.4448, `https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.4448`.

Sang A, qi Li S (2000). "A Predictability Analysis of Network Traffic." In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 1, pp. 342–351 vol.1. `doi:10.1109/INFCOM.2000.832204`.

Schnepp S, Mathisen SS, Levsen H, Pohl G, Stoeckel Q (2021). "Munin Monitoring." `http://munin-monitoring.org/`. URL `http://munin-monitoring.org/`.

Zare Moayedi H, Masnadi-Shirazi MA (2008). "Arima Model for Network Traffic Prediction and Anomaly Detection." In *2008 International Symposium on Information Technology*, volume 4, pp. 1–6. doi:10.1109/ITSIM.2008.4631947.

Zhou B, He D, Sun Z (2006). "Traffic Modeling and Prediction Using ARIMA/GARCH Model." In A Nejat Ince, E Topuz (eds.), *Modeling and Simulation Tools for Emerging Telecommunication Networks*, pp. 101–121. Springer US, Boston, MA. ISBN 978-0-387-34167-5. doi:10.1007/0-387-34167-6_5.

A. Geometric averaging

In this section, we report the results of an experimental aggregator based on geometric averaging. Since MRTG offers only 600 raw samples at a resolution of 5 min it cannot be used to evaluate geometric aggregation over 2h and 30 min as this requires 18,000 samples.

For the purpose of evaluating the link was monitored for over two months to obtain a new time series. The traffic was aggregated into 600 samples at resolution 30 min and 600 at resolution 2h to match exactly our main experiment. Two aggregated datasets were produced: one from the raw observation (arithmetic), and one from the log-transformed observations (geometric). For both datasets, we repeated the main experiment reported in the paper obtaining two types of forecast: trained on 30 min data only and combined 2h and 30 min. The Mean absolute errors obtained in the four experiments are reported in table 1. The accuracy is

Table 1: Mean Absolute Error (Mb/s) of the forecast

Aggregation	30 min	30 min and 2 h
Arithmetic	395±1.8	389±1.8
Geometric	2749±407	**376±5**

calculated in the original scale. The forecasts in the geometric case are transformed forecasts from the log-transformed observations and visualized in figure 2.

The first observation from the experiment is that additional observations always improve prediction. In the arithmetic experiment, the benefit is not large as in the main experiment because the traffic is quite regular and the training dataset does not contain any holidays that may introduce a large variance of the trend component. Having said that in the geometric experiment we observe a huge improvement in the prediction. In fact, the additional observations at 2h resolution are necessary to make a long-term forecast. Without this, the large variance of the model gets magnified by the exp transform that results in the average (note that the mean of log-normal distribution depends on the variance of the underlying normal distribution) of over an order of magnitude too large in two days. Using a median as a point forecast would yield more stable and interpretable long-term predictions as with normal distribution, the mean equals the median. And the median of a transformed random variable is the transformed median. On the other hand, low variance forecasts from the model trained on the joint dataset follow the observed traffic. Furthermore, the forecast error is reduced by 5% compared to the baseline of a model trained on 30 min dataset with arithmetic aggregation.
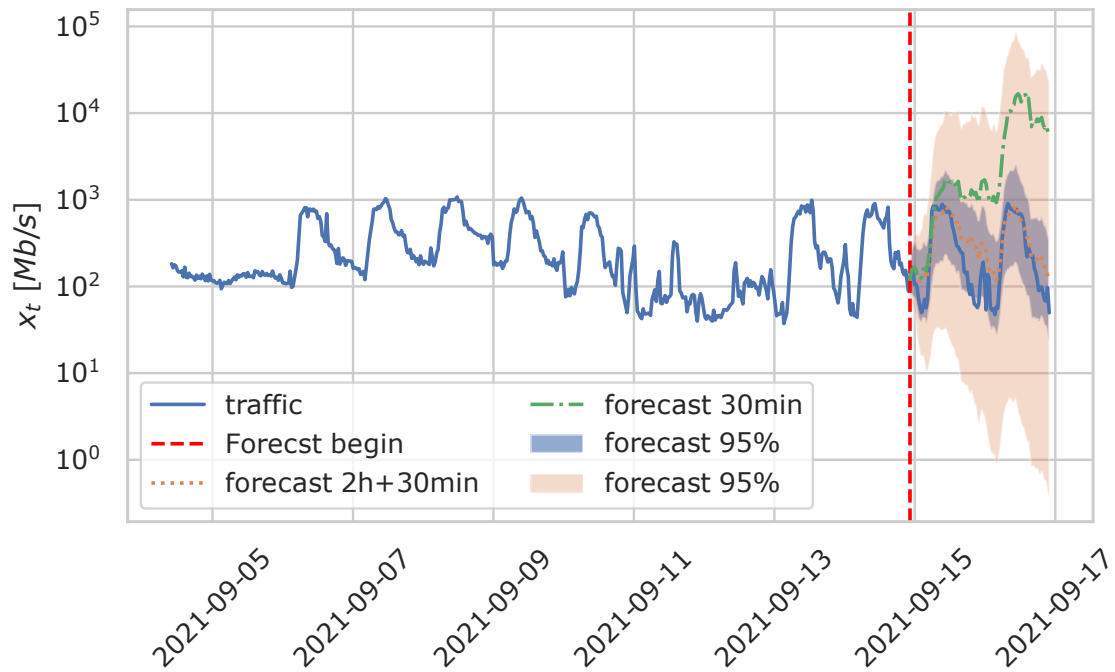
Figure 2: Fine-grained traffic time series and prediction (with prediction intervals) for geometric aggregation

Affiliation:

Krzysztof Rusek
AGH University of Science and Technology
Institute of Telecommunications, Al. Mickiewicza 30,
30-059 Kraków, Poland .
E-mail: krusek@agh.edu.pl
URL: https://www.agh.edu.pl

Mathias Drton
Technical University of Munich,
Department of Mathematics, Boltzmannstr. 3,
85748 Garching b. München, Germany
E-mail: mathias.drton@tum.de
URL: https://www.tum.de