International Journal of Statistics and Probability; Vol. 4, No. 2; 2015
ISSN 1927-7032 E-ISSN 1927-7040
Published by Canadian Center of Science and Education

Best Predictive Generalized Linear Mixed Model with Predictive Lasso for High-Speed Network Data Analysis

Kejia Hu[1], Jaesik Choi[2], Alex Sim[3] & Jiming Jiang[4]

[1] Kellogg School of Management, Northwestern University, United States

[2] Ulsan National Institute of Science and Technology, Korea

[3] Lawrence Berkeley National Laboratory, United States

[4] University of California, Davis, United States

Correspondence: Jiming Jiang, University of California, Davis, United States. Tel: 1-530-564-0639. E-mail: jiang@wald.ucdavis.edu

Received: February 22, 2015 Accepted: March 16, 2015 Online Published: April 27, 2015

doi:10.5539/ijsp.v4n2p132 URL: http://dx.doi.org/10.5539/ijsp.v4n2p132

The research is financed by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Jiming Jiang's research is partially supported by the NSF grants SES-1121794 and NIH grant R01-GM085205A1. Jaesik Choi's research is partially supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2014R1A1A1002662), the NRF grant funded by the MSIP (NRF-2014M2A8A2074096).

Abstract

Optimizing network usage is important to maximize the network performance. When the network usage grows rapidly, it is important to build an accurate predictive model. We present a new predictive algorithm which can analyze the network performance in various network conditions and traffic patterns. Our approach is based on the best predictive generalized linear mixed model (GLMM). The parameters of the best predictive GLMM are estimated by minimizing the mean squared prediction error (MSPE). To expedite the parameter learning with the big data collected through the network, our algorithm introduced regularization, LASSO, and an innovative bootstrap. The merits of our new approach validated through data and simulation are that (1) the highest prediction accuracy even under a model misspecification; and (2) the least computation time compared to the Estimation-oriented GLMM with Lasso and Stepwise Selection GLMM. A major computational advantage of our method is that, unlike some of the current approaches, our method does not require the EM (Expectation-Maximization algorithm) procedure.

Keywords: Generalized Linear Mixed Models (GLMM), Mean squared prediction error (MSPE), Model misspecification, Lasso regularization, Tuning Parameters Selection

1. Introduction

Efficient data access is essential for sharing massive amounts of data among many geographically distributed collaborators. The analysis of network traffic is getting more and more important today to efficiently utilize the limited resources offered by the network infrastructures and plan large data transfers wisely. Data transfer performance for large dataset can be improved by learning the current condition and accurately predicting the future network performance condition. Short-term prediction of network traffic performance guides the immediate scientific data placements for network users. Long-term forecast of network traffic enables the capacity planning of the network infrastructure up to the future needs for network designers. Such prediction becomes non-trivial when the amount of network measurement data grows in unprecedented speed and volumes, and prediction models are unlikely to be correctly specified. The available data sources are network transfer data collected under Simple Network Management Protocol (SNMP) (Stallings, W., 1999) and flow-level data such as Cisco's NetFlow (Cisco Systems Inc., 1999). SNMP provides low-volume data which is a single time series regarding the time and the corresponding

aggregated traffic volume in the network. Recent statistical research on SNMP data are Hu, Sim, Antoniades & Dovrolis (2013) and Antoniades, Hu, Sim & Dovrolis (2013). On the other hand, NetFlow measurements provide high volume with abundant specific information of each data flow such as time, path and delivery condition. A technical report about statistical analysis on Netflow data with preliminary results are done by Hu, Choi, Jiang and Sim (2013).

In this paper, statistical models are built to predict network usage based on the high volume NetFlow data. An accurate prediction of network performance is crucial for both the network users and network designers. For network users, an accurate prediction of the duration of a data transfer can help them choose the start time and the path in order to transfer data in a fast and stable delivery condition. A concrete practical example is to predict the required time to finish a transfer given the data size, the start time and the transfer path. For network designers, accurate prediction can identify the future needs in the network usage and allocation of the bandwidth resources as well as choosing locations of the hub in the long run. For example, if a selected path is predicted to have frequent congestions, then the designer can accordingly expand the network bandwidth to meet the size of the data flow in the path or adding alternative paths to the current network and rerouting partial data flow.

The motivation for modeling the NetFlow measurements using Generalized Linear Mixed Model (GLMM) comes from two perspectives: (1) the features of NetFlow data; and (2) the capability of GLMM. NetFlow records are composed of multiple time series with uneven collecting time stamps, and thus the traditional single time series model is infeasible to model the data. Also, NetFlow records show mixed effects in the data, and simple consideration of fixed effects will cause information loss in the modeling (Hu, Choi, Jiang & Sim 2013). Moreover, the large volume and high dimension of NetFlow data require a fast algorithm so that the future transfer planning can respond quickly to the predicted network conditions. On the other hand, GLMM has a very flexible structure in the model, and can incorporate various variance sources and mixed effects, and thus fits our needs of analyzing the NetFlow data.

Previous research (Jiang et al. 2011 and Bondell et al. 2010) discussed two issues in Linear Mixed Model (LMM), which is the GLMM with identity link and Gaussian assumption. Jiang et al. (2011) shows how to obtain the best prediction in the LMM (Linear Mixed Model), and Bondell et al.(2010) uses the Lasso to select random effect in LMM for the estimation purpose. However, there are no existing methods for selecting both random effects and fixed effects for the purpose of best prediction via the Lasso (Tibshirani 2011) in LMM.

To match GLMM with the prediction interest and NetFlow data, we improve the GLMM accordingly in Section 2. We first discuss the approach to obtain the estimates of fixed and random effects by Lasso with minimum mean squared prediction error (MSPE) in LMM. Then, we extend the methodologies to GLMM with the log link and Poisson assumption.

The major computational advantage of our method is the dramatic cut in computing time. Unlike Bondell et al. (2010) and Ibrahim et al. (2011), which requires the EM algorithm (Dempster et al. 1977) to handle the unobserved random effects, our procedure does not require the EM and thus saves a large amount of the computing time.

Moreover, we propose a new approach based on bootstrapping to select the optimal penalty parameter λ in Lasso. In the theoretical derivation, two advantages of this approach are analyzed: immunity to model misspecification and fast computational algorithm. After discussing the methodology in Section 2, we show the model application with NetFlow data in Section 3 and the simulation in Section 4, followed by the summary and discussion.

2. Methodology

In this section, we first discuss the NetFlow measurements with its format and how it matches with GLMM. Then, we show the derivation of the best predictive GLMM with Lasso.

2.1 NetFlow Dataset

NetFlow measurements provide high volume with abundant specific information as shown in Table 1 (with IP address is masked for privacy issues). For each record, it shows a particular data transfer with the following *variables*.

Start, End The start and end time of the recorded data transfer.

Sif, Dif The source and destination interface assigned automatically for the transfer.

SrcIPaddress, DstIPaddress The source and destination IP addresses of the transfer.

SrcP, DstP The source and destination port chosen based on the transfer type such as email, FTP, SSH, etc.

P The protocol chosen based on the general transfer type such as TCP, UDP, etc..

Fl The flags measured the transfer error caused by the congestion in the network.

Pkts The number of packets of the recorded data transfer.

Octets The Octets measures the size of the transfer in bytes.

Table 1. NetFlow Records

Start	End	Sif	SrcIPaddress(masked)	SrcP	Dif
DstIPaddress(masked)	DstP	P	Fl	Pkts	Octets
0930.23:59:37.920	0930.23:59:37.925	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	22364	6	0	1	52
0930.23:59:38.345	0930.23:59:39.051	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	28335	6	0	4	208
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	20492	6	0	2	104
0930.23:59:59.443	0930.23:59:59.443	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	26649	6	0	1	52
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	26915	6	0	1	52
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	20886	6	0	2	104

Considering features of NetFlow data and its prediction interests, Generalized Linear Mixed Model (GLMM) is advocated in this paper with the following distinctive advantages.

- NetFlow record is composed of multiple time series with uneven collecting time stamps. Because of this feature, traditional time series methods such as ARIMA model (Box and Pierce, 1970), wavelet analysis (Percival and Walden, 2000), and exponential smoothing model (Bloomfield, 1972) are not applicable since they are designed for evenly collected time stamps and mainly deal with a single time series. Some researches (Box and Tiao, 1981, Box and Tiao, 1975 and Geweke, 1982) have extended their usage in two time series, but the complexity and inefficiency prevent them to go beyond. Thus, there is a need for a model which can handle multiple time series at same time without the constraint of even time stamps of collection. Facing these requirements, GLMM can fully utilize all variables in the dataset with no need for an even-spaced time variable.

- NetFlow record is a multivariate dataset showing mixed effects. In Figure 1, we see that as the number of packets in a data transfer increases, it takes longer time in general to finish the data transfer. This suggests that the number of packets can be a fixed effect to predict the duration of a data transfer. With a close look, we see that in different network transfer paths, the relationships between duration and the number of packets are different showing the fluctuation patterns such as slope rate and dispersion range. This suggests that the network path for data transfers can be considered as a random effect to explain the duration under varying conditions. Thus, in terms of modeling mixed effects, GLMM has the strength over Generalized Linear Model (GLM) that only considers fixed effects, and it has the flexibility over Linear Mixed Model(LMM) that can only model continuous response variables along with Gaussian assumption. GLMM is more flexible and general in the sense that it expands the choice of underlying distribution by relating the linear model to the response variable via a link function and categorizes the variance source by measuring the random effects.

- NetFlow measurements are a big data with large volume as well as high dimensions. It has millions of observations from a single router within a day. Each record contains 14 variables, and the candidate variables in the model can be as much as 30s or 40s when considering reasonable interaction terms. This giant set of data requires an efficient modeling. When identifying distinctive patterns within each group, a traditional hierarchical model (Lee and Nelder, 1996) divides the data according to the groups, and then generates a model for each group. However, there are three main reasons why hierarchical models are not feasible in this case:

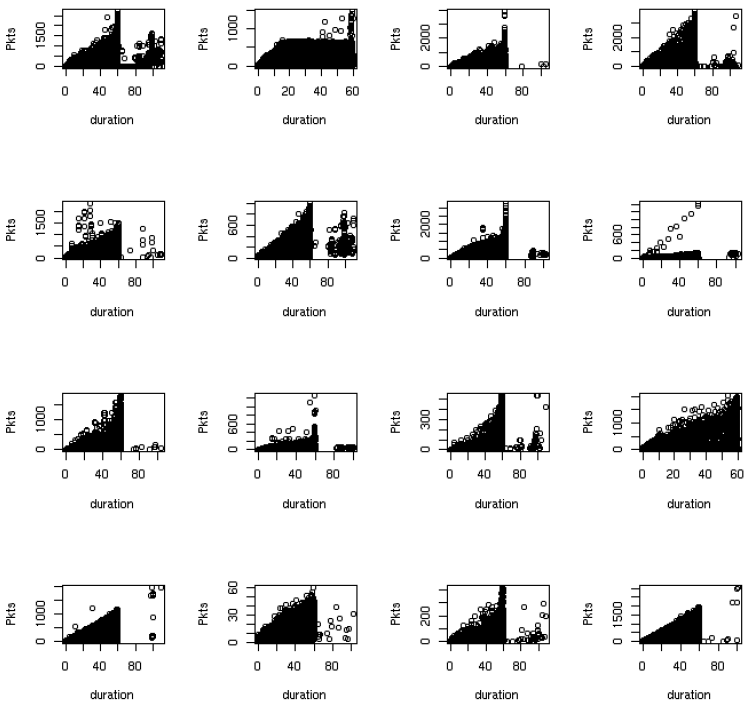

Figure 1. Relationship (Number of Packets v.s. Duration) on 16 different paths, showing mixed effect such as transfer path

- The grouping factor is not clear in network data, and it requires an investigation to identify the suitable variable sets that classify the observed data. Explorative data analysis shows that the grouping factor can be a path of the data transfer, the delivering time of the day, the transfer protocol used and all kinds of their combinations. If using hierarchical model, we need to first choose the correct grouping factor by modeling the data several times and examine which grouping factors make the most sense. Then we need to apply model selection techniques for each subgroup before we utilize the model to predict. However, when using GLMM, all work would be finished in one step. The GLMM conducts the model selection and model fitting at the same time. The best grouping factors are suggested by the automatically selected significant random effect, and the estimated model parameters are presented at the same time.
- Hierarchical models worsen the prediction accuracy since their errors converge to 0 in a slower rate compared with the one from GLMM.
- GLMM provides one model for all dataset. However, hierarchical models generate several models, one for each group. This makes the hierarchical models less efficient in use, compared to the GLMM approach.
- GLMM uses the entire information set to select and fit the model. However hierarchical models use only subgroup dataset, i.e. the partial information to generate each model and each subgroup model has less power in information.

2.2 Generalized Linear Mixed Model

The GLMM is defined with a vector of random effects v and the responses $y_1, ..., y_m$ of m groups that are conditionally independent such that the probability density function(pdf) of each response $f_i(y_i|v)$ follows the exponential family with

$$E(y_i|v) = \mu_i, g(\mu_i) = x_i'\beta + z_i'v, g^{-1} = h \tag{1}$$

where $v \sim N(0, \Phi)$, x_i is the observed fixed effect, and z_i is the index that indicates the group of random effect.

The $g(.)$ is the link function, and takes various forms such as Gaussian, Poisson and Logit with different assumptions of the model. In the NetFlow data, the only two types of variables are (1) continuous variables such as the size of the data transfer and the duration measured in milliseconds, and (2) count variables such as the number of congestions or the number of extreme large data transfers within a certain time window length. In order to predict these two types of response variables, the GLMM are constructed in the following two types.

- y_i is the continuous variable, with $g(x) = x$, where $y_i|v$ follows Gaussian distribution.
- y_i is the count variable, with $g(x) = log(x)$, where $y_i|v$ follows Poisson distribution.

The mixed effects θ of prediction interest and its Best Predictor (BP) $\check{\theta}$ under the assumed model M are

$$\theta = h(F'x\beta + R'v), \text{ where } F \text{ and } R \text{ are known matrices} \tag{2}$$

and

$$\check{\theta} = E_{M,\psi}(\theta_i|y) = h_{M,i}(\psi, y_i), \text{ where the parameter set } \psi = \{\beta, \Phi\} \tag{3}$$

M stands for the assumed model, and $h_{M,i}$ is the function showing the BP of θ connected with ψ and y_i. The MSPE to be minimized can be expressed as

$$\begin{aligned} \text{MSPE}(\check{\theta}) &= E(|\check{\theta} - \theta|^2) \\ &= \sum_{i=1}^{m} E\left(h_{M,i}(\psi, y_i) - \theta_i\right)^2 \\ &= E\left(\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\right) - 2\sum_{i=1}^{m} E\left(h_{M,i}(\psi, y_i)\theta_i\right) + \sum_{i=1}^{m} E(\theta_i^2) \\ &= I_1 + 2I_2 + I_3 \end{aligned} \tag{4}$$

Note that, unlike $E_{M,\psi}$ which depends on the assumed model as well as the parameter ψ, the E in Equation 4 is with respect to the true underlying distribution of y and θ, which may be unknown but not model dependent. This is the key feature of the Jiang's approach (Jiang et al. 2011).

First, consider the single case that Φ is known. Denote the MSPE in (4) by MSPE(β). Then, it is straightforward to apply the Lasso to select the fixed effects, that is,

$$\check{\beta} = argmin_{\beta}(\text{MSPE}(\beta) + \lambda|\beta|) \tag{5}$$

However, selecting the random effects using the Lasso is not simple. This is because the insignificant fixed effects are eliminated with its coefficient β diminishing exactly to 0; however, the insignificant random effects are eliminated with the corresponding whole columns and whole rows of the covariance matrix diminishing exactly to 0 (Bondell et al. 2010 and Ibrahim et al. 2011). After the covariance matrix is reformed, its positive definite property should also be maintained. In order to solve this difficulty, we use the Cholesky decomposition on the covariance matrix.

$$\Phi = D\Lambda\Lambda'D \tag{6}$$

where D is diagonal matrix $D = diag(d_1, d_2, ..., d_q)$, and Λ is the lower triangular matrix with 1's on the diagonal. The standardized model with the identity link function is

$$y_i = X_i\beta + Z_iD\Lambda v_i + e_i \text{ where } v_i \sim N(0, I),\ e \sim N(0, \Sigma) \tag{7}$$

The shrinkage penalty is imposed on d_i, the element of diagonal matrix D. When d_i is shrunk to 0, the corresponding random effect is eliminated. The covariance matrix $\Phi = D\Lambda\Lambda'D$ is still guaranteed to be positive definite. After the decomposition, the original random effect coefficients Z_i^*, Z^* change into

$$Z_i = Z_i^*D\Lambda, \quad Z = Z^*\tilde{D}\tilde{\Lambda}, \quad \tilde{D} = I_m \otimes D, \quad \tilde{\Lambda} = I_m \otimes \Lambda$$

The parameter set of prediction interest $\psi^* = \{\beta, \Phi\}$ changes into $\psi = \{\beta, d\}$.

2.3 Case 1: Gaussian distribution

LMM is a special case of the GLMM when the link function is identity, that is, $h(\mu) = \mu$ in (1), and the underlying exponential family is Gaussian. The mixed effect of prediction interest and its BP under assumed model M are

$$\theta = F'x\beta + R'v \text{ where } F \text{ and } R \text{ are known matrices} \tag{8}$$

$$\breve{\theta} = E_M(\theta|y) = F'X\beta + R'E_M(v|y) = F'X\beta + R'Z'V^{-1}(y - X\beta) \text{ where } V = var(y) = \Sigma + ZZ' \tag{9}$$

With previous notation R, Z, V and F, now write $B = R'Z'V^{-1}, \Gamma = F' - B$ and $H = Z'F - R$. For fixed effects without Lasso in Jiang's research (Jiang et al 2011), it shows:

$$\begin{aligned} \text{MSPE}(\breve{\theta}) &= E(|\breve{\theta} - \theta|^2) \\ &= E(|H'v + F'e|^2) - 2E((v'H + e'F)\Gamma(y - X\beta)) + E((y - X\beta)'\Gamma'\Gamma(y - X\beta)) \\ &= I_1 - 2I_2 + I_3 \end{aligned} \tag{10}$$

Since the true model tells $y = \mu + Zv + e$,

$$\begin{aligned} I_2 &= -2E((v'H + e'F)\Gamma(y - X\beta)) \\ &= -2E((v'H + e'F)\Gamma(y - \mu)) - 2E((v'H + e'F)\Gamma(\mu - X\beta)) \\ &= -2E((v'H + e'F)\Gamma(Zv + e)) \end{aligned} \tag{11}$$

Among the three components, I_1 and I_2 are not related to β. Since β is the only parameter that matters in the minimization of MSPE($\breve{\theta}$), the minimization is equivalent to

$$\breve{\beta} = argmin((y - X\beta)'\Gamma'\Gamma(y - X\beta))$$

It's important to note that 1) the MSPE is calculated with $E(.)$, 2) the expectation is under the true model rather than $E_M(.)$, and 3) the expectation is related to the assumed model M. This MSPE calculation feature in this method guarantees that $\breve{\beta}$ is immune to model misspecification, and proves to have better prediction accuracy than the estimates BLUP (Best Linear Unbiased Prediction) resulted from MLE (Maximum Likelihood Estimator). Besides the immunity to model misspecification, selecting the fixed effects via Lasso for efficient model selection is also imposed.

$$\breve{\beta} = argmin_\beta(y - X\beta)'\Gamma'\Gamma(y - X\beta) + \lambda \sum |\beta_i| \tag{12}$$

The selection of random effects is not addressed in the Jiang et.al (2011), and is now discussed in this paper. For random effects selection, the MSPE is minimized only with the part in Equation 4, related to the parameter of interest d.

$$\begin{aligned} \text{MSPE}(\breve{\theta}) = C &+ tr((Z'FF'Z - Z'FBZ - RF'Z + RBZ)G) \\ &- tr(FB\Sigma) + E((y - X\beta)'\Gamma'\Gamma(y - X\beta)) \\ &+ tr((Z'FF'Z - Z'FR' - RF'Z)G) \end{aligned} \tag{13}$$

where $C = 2tr(FF'\Sigma) + tr(RR'G)$ is not related to d.
Applying the L_1 penalty along with MSPE to achieve the efficient model selection with Lasso, the random effects are

$$\begin{aligned} \breve{d} = &argmin_d(y - X\beta)'\Gamma'\Gamma(y - X\beta) \\ &+ tr((2HBZ - HF'Z - Z'FR')G) - tr(FB\Sigma) \\ &+ \lambda \sum |d_i| \end{aligned} \tag{14}$$

The final model for prediction is built with the above equations (12) and (14), and has two distinctive advantages. First, it is mentioned in the paper that the minimization problem is not based on the assumed model M, and thus immune to the model misspecification errors. Second, the computation complexity is only a minimization problem with $O(nP)$, where n is the number of observations and P is the number of parameters in the full model. It is much simpler than the MCEM (Monte Carlo EM) algorithm that is required by Bondell et al. (2010) and Ibrahim et al. (2011) in order to handle the unobserved random effects in their estimation-based penalized maximum likelihood algorithms.

2.4 Case 2: Poisson Distribution

The second case is when the response variable is the counted data. Given the small area means $\mu_1, ..., \mu_m$, the observations $y_1, ..., y_m$ (with y_i being from the ith small area) are independent such that

$$y_i \sim Poisson(\mu_i); \quad log(\mu_i) = x_i'\beta + z_i v_i \tag{15}$$

The vector of prediction interest and its BP is

$$\theta = h(F'x\beta + R'v), \quad \breve{\theta} = E_{M,\psi}(\theta_i|y) = h_{M,i}(\psi, y_i) \tag{16}$$

Utilizing the properties of Poisson distribution as derived in the Appendix,

$$\begin{aligned} \text{MSPE}(\breve{\theta}) &= E\left(\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\right) - 2\sum_{i=1}^{m} E\left(h_{M,i}(\psi, y_i)\theta_i\right) + \sum_{i=1}^{m} E(\theta_i^2) \\ &= E\left(\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i + \sum_{i=1}^{m} E(\theta_i^2)\right) \end{aligned} \tag{17}$$

Since $\sum_{i=1}^{m} E(\theta_i^2)$ has no relationship with the parameter set $\phi = \{\beta, d\}$, minimizing MSPE is equivalent to minimizing

$$Q(\psi) = \sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i \tag{18}$$

The fixed and random effects under Poisson cases are

$$\beta = argmin_\beta Q(\psi) + \lambda_\beta \sum_{j=1}^{p} |\beta_j|, \quad d = argmin_d Q(\psi) + \lambda_d \sum_{i=1}^{m} |d_i|. \tag{19}$$

2.5 Selecting Penalty Parameters in Lasso

In Jiang et al. (2008), the adaptive fence procedure raises a method to select the tuning constant c_n that is used in the model selection. The idea is that it is ideal if the selecting tuning constant c_n maximizes the probability of choosing the optimal model. Suppose M is the set of candidate models which include the optimal model M_{opt}, and the selected model is $M_0(c_n) \in M$. Then, optimal c_n should be

$$c_n = argmax_{c_n} P(M_0(c_n) = M_{opt}) \tag{20}$$

In order to find c_n through equation (20), two keys must be known: (1) the underlying distribution to compute P; and (2) M_{opt}.
In Jiang et al. (2008), the first key, probability distribution P can be approximated by the bootstrapped samples under the full model M_f. The second key, M_{opt} can be found utilizing the idea of maximum likelihood. The optimal model is the model that generates data, and thus should be the model that is favored the most by the data. Since the bootstrapped samples almost duplicate the information from the original data, M_{opt} is the most supported by the bootstrapped samples, i.e. most frequent to be selected.

Extending the adaptive fence idea into the penalty parameter selection λ_n in Lasso, the procedures are:

- Step 1: Fit the full model M_f and bootstrap B samples from it
- Step 2: Select a grid of λ. For each λ, record $M_*(\lambda)$, the model that is selected the most, across B samples, i.e. $M_*(\lambda) = argmax_M P(M_0(\lambda) = M(\lambda))$ where $P(M_0(\lambda) = M(\lambda))$ is counted as the portion that the number of samples supports model $M(\lambda)$ as the selected model $M_0(\lambda)$ out of B.
  Note here that the final selected model $M_*(\lambda)$ is related to λ. When $\lambda \to 0$, the model that is selected the most $M_*(\lambda)$ will be the full model M_f, and when $\lambda \to \infty$, the model that is selected the most $M_*(\lambda)$ will be the empty model M_{empty}.

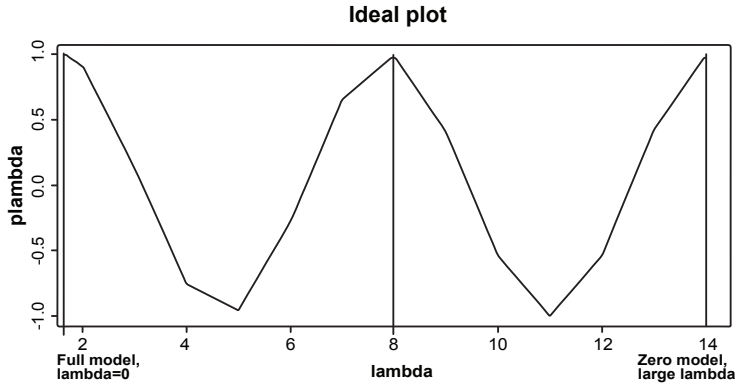

Figure 2. The Ideal (with unique peak in the middle) Plot for Selecting Tuning Parameter λ

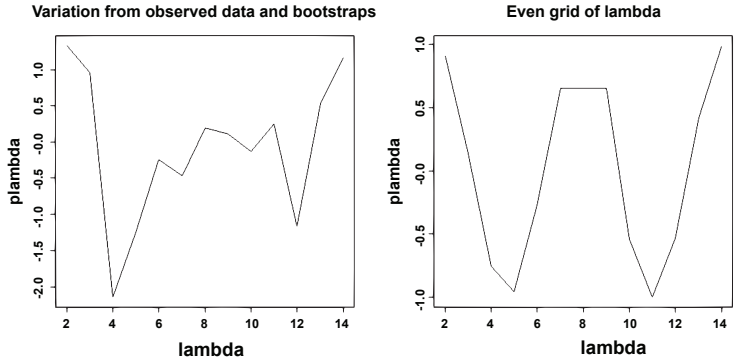

Figure 3. The Common (without unique peak in the middle) Wiggling and Platform Plots for Selecting Tuning Parameter λ

- Step 3: Denote $P_*(\lambda) = P(M_0(\lambda) = M_*(\lambda))$ as the support percentage of the favorite $M_*(\lambda)$ given λ. Plot $P_*(\lambda)$ against λ. In the ideal plot as Figure 2 with a unique peak in the middle, this peak is the favorite tuning parameter λ.

In Figure 2, the two ends have peaks because when λ is either too small or too large, only the full model M_f and the empty model M_{empty} will be selected. In Jiang's approach (Jiang et al. 2008), the λ corresponding to the peak in the middle of the plot should be the chosen λ, which maximizes the probability that the selected model is equal to the optimal model, $M_0(c_n) = M_{opt}$. The ideal situation does not always show, and in many times, one will end up in either of the cases shown in Figure 3. The fluctuation in the left case occurs due to the variation from the observed data and bootstraps. The platform in the right case occurs due to the fine cut in the grid of λ.

To solve these two problems, one no longer uses evenly-spaced grid of λ, but a dimension-related λ that λ_j corresponds to the j-predictors model. The detailed new approach goes through the following steps:

- Step 1: Start from smallest $\lambda_p = 0$. It returns M_p, a full model with p predictors. Keep increasing λ, until M_{p-1}, a model with $p-1$ predictors, is returned. Record the current value as λ_{p-1}
  $[\lambda_p, \lambda_{p-1})$ is the range that the model with p predictors is chosen.
- Step 2: Keep increasing λ, until one gets all the ranges for the dimension wise model selection. For $i = 0, ..., p, [\lambda_i, \lambda_{i-1})$ is the range that models with i predictors are chosen.
- Step 3: For each range $[\lambda_i, \lambda_{i-1})$, evenly separate the range into a grid by k candidates of λ. For each λ, compute the model across all bootstrap samples, and choose the optimal λ within this range as λ_i^* and the

corresponding $p_i^*(M_i^*, \lambda_i^*)$. Table 2 summaries the tuning parameters' range, the best λ and its supported percentage for each dimension.

- Step 4: Plot p_i^* against λ_i^* that are summarized in Table 2. The middle peak is selected as the overall optimal λ^*, and its corresponding model is selected as the final optimal model M^*.

The platform case is solved because each λ now selects the model with a different number of predictors, and the corresponding probability is not likely to be the same in the neighboring range. The variation case is solved because more robust and sophisticated choice of λ_j eliminates the unwanted wiggling in the plot. The resulted plot is more close to the shape of the ideal case shown in Fig. 2.

Table 2. The New Approach to Solve Wiggling and Platform: Dimensional Selection for Tuning Parameter λ

dim	p	p-1	...	i	...	0
λ range	$[\lambda_p, \lambda_{p-1})$	$[\lambda_{p-1}, \lambda_{p-2})$	...	$[\lambda_i, \lambda_{i-1})$	...	$[\lambda_0, \lambda_{-1})$
λ_i^*	λ_p^*	λ_{p-1}^*	...	λ_i^*	...	λ_0^*
p_i^*	p_p^*	p_{p-1}^*	...	p_i^*	...	p_0^*

3. NetFlow Data Study

The sample NetFlow data is provided by ESnet for the duration from May 1, 2013 to June 30, 2013. Considering the network users' interests, the established model should predict the duration of a data transfer so that users can expect how long the data transfer would take, given the size of their data, the start time of the transfer, selected path and protocols. Considering the network designers' interests, the established model should predict the long term usage of the network, so that the designer will know which link in the network is usually congested and requires more bandwidth or rerouting of the path. In the following, the models for these two interests are built, and its prediction accuracy is compared to two traditional GLMM algorithms: Backward-Forward selection and Estimation-based Lasso.

3.1 GLMM on Duration

The full model predicts the transfer duration, assuming influences from the fixed effects including the transfer start time, the transfer size (Octets and Packets) and the random effects including network transfer condition such as Flag and Protocol, source and destination Port numbers and transfer path such as source and destination IP addresses and source and destination Interfaces. After selecting and fitting the model via our Predictive Lasso procedure, the final model is

$$y = \beta_{start} s(x_{start}) + \beta_{pkt} x_{pkt} + Z_{ip-path} v_{ip-path} + e \tag{21}$$

Since the time variable is usually not linear related to the response variable, smoothing spline transformation $s(.)$ is implemented for the time variable, and the smoothing parameters are chosen automatically by cross-validation. P-value in the final model is all less than 2e-16, which stands for the significance of those variables in the model. The significant fixed effects in the model are start time and number of packets, as shown in Table 3. The transformed start time data is plotted in Figure 4, and the plot tells how the duration varies for different start time.

Table 3. Coefficient Estimates for Fixed Effects in GLMM (21) to Predict the Transfer Duration

Fixed Effects	Estimates	Standard Deviation	P-value
Intercept	-13.809	0.914	<2e-16
Start Time	0.574	0.0169	<2e-16
Packets	1.115	0.035	<2e-16

Table 4. Comparison of MSPE and Speed to Predict Duration

	Est.Lasso s	BF Selection	Pred. Lasso
MSPE	2306	42230	127.3
Modeling Time (in seconds)	6.26e+7	5.43e+10	142

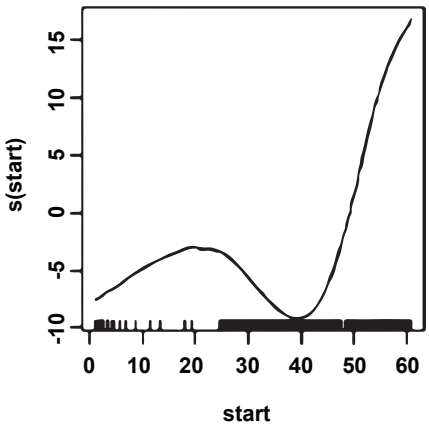

Figure 4. Smoothing Spline Transformed Start Time Variable, showing the nonlinear relationship between start time and transfer duration

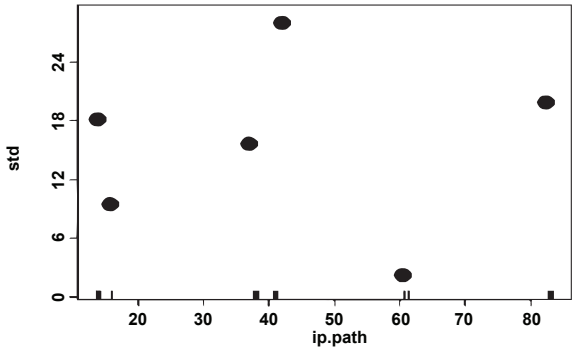

Figure 5. Standard Deviation Estimates for Random Effects in GLMM (21) to Predict The Transfer Duration, showing the busier paths bring higher variation to the transfer duration

The random effects' standard deviance estimates are plotted in Figure 5, and the plot shows that the traffic duration varies with the different IP paths. In our sample, there are six paths indexed as 83, 38, 41, 14, 16 and 61. The index is categorical representation of the IP path, and has no numerical values in the model. The busier paths, such as paths 83, 38, 41 and 14 that have dense area shown in the bottom of Figure 5, come with higher fluctuation rates in the transfer duration. On the other hand, the paths 16 and 61 are noticed to have less traffic and lower variation rates. Besides the variation resulted from each IP path, the background noise e in model (21) is estimated with a standard deviation of 11.2392 by the Predictive Lasso.

The model suggests the importance of variation in random effects such as IP path in the prediction of the duration. Besides the path, start time selection and assignment of packets are also significant in the prediction of the duration. Compared to the other two approaches, shown in Table 4, the Predictive Lasso shows that the best prediction accuracy is 18 times better than the Estimation Lasso and 330 times better than the Backward-Forward Selection, and the least computation time is 4e+5 times less than the Estimation Lasso and 3.8e+8 times less than the Backward-Forward Selection. The Predictive Lasso greatly improves the prediction accuracy which fits the interests of modeling and also provides efficient fast algorithm compared to the Estimation Lasso and Backward-Forward Selection, as analyzed in section 2.

3.2 GLMM on Frequency of Congestion

This model predicts the frequency of congestion occurred in each link of the network. The response variable y is the number of congestion measured by the speed, BytesPerSecs. Since slow speed is a sign of congestion, a congestion event is marked when BytesPerSecs is less than 50 which is the slowest 10% of network transfer speed. The full model to predict the number of congestion assuming influences from two sources: fixed effects and random effects. The fixed effects include transfer size (Octets and Packets), number of transfers with their Protocol 6, 17, 47 and 50 respectively, and number of transfers with their Flag is 0, 1, 2 and 4 respectively. The random effects include transfer path, and the source and destination IP addresses. After selecting and fitting the model via our Predictive Lasso procedure, the final model is

$$logE(y|v) = \beta_{pkts}x_{pkts} + \sum \beta_{p=i}x_{p_i} + Z_{ip-path}v_{ip-path} \tag{22}$$

The significant fixed effects in the selected model are the number of packets and the protocol used, as shown in Table 5.

Table 5. Coefficient Estimation for Fixed Effects in GLMM (22) to Predict The Frequency of Congestion

Fixed Effects	Estimates	Std	P-value
Packets	28.53924	0.02284	<2e-16
Protocol=6	45.43606	0.01608	<2e-16
Protocol=17	-1.58644	0.14088	<2e-16
Protocol=47	-8.39576	0.36338	<2e-16
Protocol=50	-4.96028	0.05175	<2e-16

Table 6. Comparison of MSPE and Speed to Predict Counts of Congestion

	Est.Lasso s	BF Selection	Pred. Lasso
MSPE	27.7	42.5	12.73
Modeling Time (in seconds)	7.31e+8	1.24e+10	10.06e+2

The random effects' standard deviance estimates are plotted in Figure 6, and the plot shows that the traffic duration in Y axis varies with different transfer IP paths in X axis. In our sample, there are 414 paths, and the busier path comes with the higher fluctuation rates. Besides the variation resulted from each IP path, the background noise is estimated with a standard deviation of 25.316 by Predictive Lasso.

The model suggests the importance of variation in random effects such as IP path in predicting the congestion frequency. Besides the path, the protocol selection and assignment of packets also significantly affect the congestion rate. Compared to the other two approaches shown in Table 6, the Predictive Lasso shows the best prediction accuracy which is twice better than the Estimation Lasso and four times better than the Backward-Forward Selection,

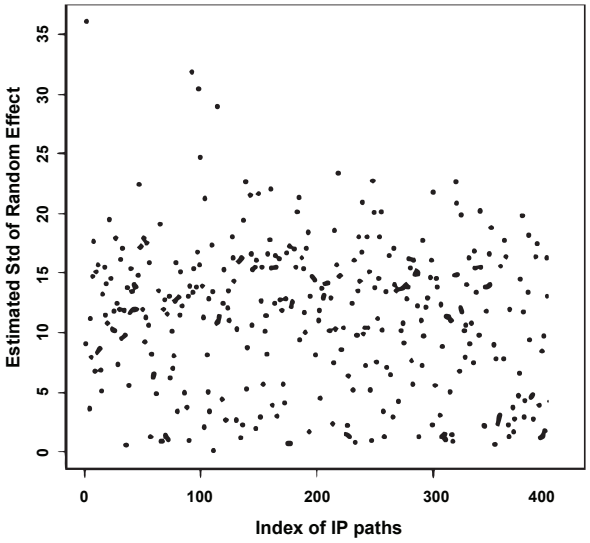

Figure 6. Standard Deviation Estimates for Random Effects in GLMM (22) to Predict The Frequency of Congestion, showing the busier paths bring higher variation to the frequency of congestion

and the least computation time in modeling step which saves 7e+8 seconds than the Estimation Lasso and 1e+10 seconds than the Backward-Forward Selection. Although the prediction accuracy improvement by the Predictive Lasso is not as dramatic as in the previous case, the saving in computing time is more impressive.

4. Simulation

The Predictive Lasso developed in section 2 is shown in section 3 to have two main advantages in terms of better prediction accuracy and less computational cost than the estimation-oriented methods. In this section, we use simulation studies to furthur support and illustrate these two advantages. The Predictive Lasso is compared with two model selection alternatives: (1) the Backward-Forward Selection, the classical model selection approach, and (2) the Estimation Lasso, the representative penalized model selection approach. The first advantage of better prediction accuracy is due to two reasons: First, the optimization is calculated without using E_M or the distribution of the assumed model. Thus the parameter estimates are not affected by the model misspecification error. Secondly, the Predictive Lasso minimizes the MSPE, while the estimation-oriented methods target to maximize likelihood. In this way, the Predictive Lasso gets smaller prediction error for both fixed effects prediction and random effects prediction.

In the simulation, we will first generate data from the true model. Then a proposed model is misspecified with several redundant fixed and random effects. The performance of the three model selection techniques are examined under three scenarios: increasing variance (sd), increasing number of observations in each groups n_i, and increasing number of groups N.
The simulation data is generated from the true model,

$$M = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + z_{ij1} v_{1i} + z_{ij2} v_{2i}$$

$$i = 1, ..., N;\ j = 1, ...n_i;\ \phi = diag(sd_1^2, sd_2^2);\ sd_1 = 3;\ sd_2 = 2$$

$$N = 20;\ n_i = 6;\ var(e_{ij} = sd^2 = 3)$$

The Gaussian model is given as $\mu = M$, and the Poisson model is given as $log(\mu) = M$. However, combined with redundant observed information and model misspecification, the assumed model is

$$\begin{aligned} M = {} & \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4} + \beta_5 x_{ij5} + \beta_6 x_{ij6} \\ & + z_{ij1} v_{1i} + z_{ij2} v_{2i} + z_{ij3} v_{3i} + z_{ij4} v_{4i} + z_{ij5} v_{5i} + z_{ij6} v_{6i} + z_{ij7} v_{7i} \end{aligned} \tag{23}$$

The assumed model is misspecified in terms of 4 redundant fixed effects and 5 redundant random effects. From Figure 7 showing the Gaussian Model and Figure 8 showing the Poisson Model, the plots on the first row show the

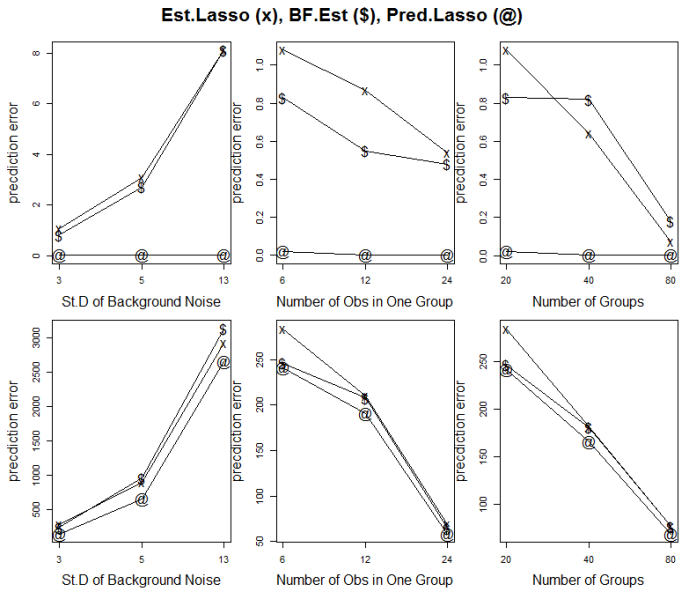

Figure 7. Prediction Accuracy under Case 1: Gaussian Model, showing the Predictive Lasso has the smallest prediction error

performance of fixed effect prediction accuracy, and the plots on the second row show the performance of random effect prediction accuracy. The plots on the left column, regarding increasing variance, show that the Predictive Lasso is not significantly worse in terms of the prediction error than the other two methods. The plots on the middle column, regarding increasing number of observations in each group, and the plots on the right column, regarding the increasing number of groups, both show that the Predictive Lasso always holds the most accuracy position no matter how the data is segmented into groups.

The second advantage of the Predictive Lasso is the dramatically reduced computational costs in reaching the final model. In the optimization steps, the Estimation Lasso and the Backward-Forward Selection require MCEM to estimate the expectation of the likelihood of the assumed model, since their target function involves non-observed random effects. However, the Predictive Lasso has an optimization function without the unobserved random effects which eliminate the costly MCEM. The computational complexity of the optimization problem in the Predictive Lasso is $O(np)$, where n is the observation and p is the number of predictors, as in Equation (24). The Estimation Lasso requires MCEM, an iterative algorithm that each iteration contains optimization and requires several iteration steps until it reaches the convergence in the final model. The Backward-Forward Selection requires l steps to reach the final model, and in each step it needs in trial-and-error to decide which variable to drop or add after using MCEM to fit each candidate model as shown in Equation (26).

$$\begin{aligned} Time(PredictiveLasso) &= Optimization \times 1 \\ Complexity(PredictiveLasso) &= O(np) \end{aligned} \tag{24}$$

$$\begin{aligned} Time(EstimationLasso) &= \text{MCEM} \times k \\ &= (\text{MC} + Optimization) \times k \\ Complexity(EstimationLasso) &= O(npk) \end{aligned} \tag{25}$$

where k is the number of iteration before algorithm converges.

$$\begin{aligned} Time(BF\ Selection) &= \text{MCEM} \times \sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij} \\ Complexity(BF\ Selection) &= O\left(np \sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij} \right) \end{aligned} \tag{26}$$

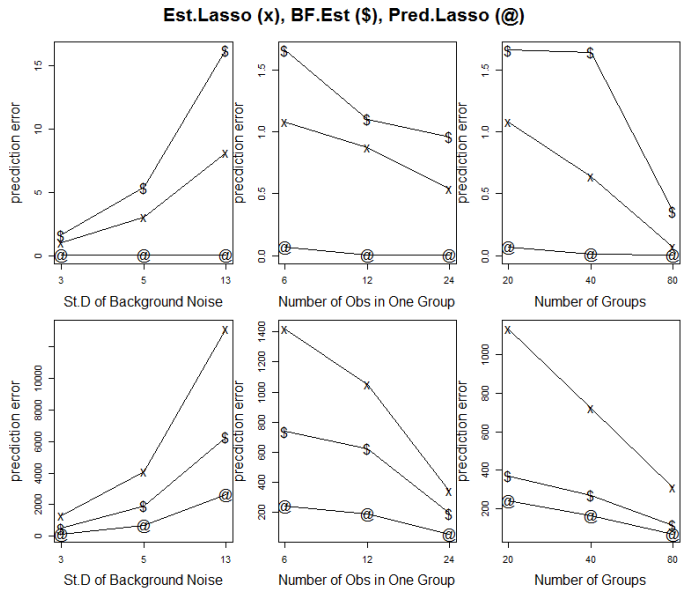

Figure 8. Prediction Accuracy under Case 2: Poisson Model, showing the Predictive Lasso has the smallest prediction error

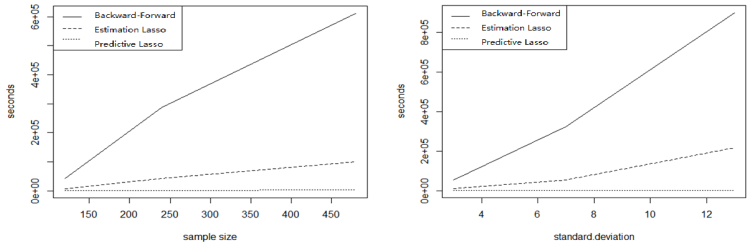

Figure 9. Computational Costs of Three Methods, showing the Predictive Lasso has the least computation time for both the size of the data increases (left) and the uncertainty in the data increases (right)

where l is the number of steps before reaching to the final model; k_i is the number of trial-and-error in the i-th step before moving to the $i+1$-th step; J_i is the number of remaining variables in the model at i-th step, and n_{ij} is the number of iteration before MCEM converges for the i-th trial with j-th predictors omitted.

From the Equation (24), (25) and (26), it is clear that the Predictive Lasso saves large computational costs compared to the Estimation Lasso and the Backward-Forward Selection. Moreover, the simulation supports the computational advantage of the Predictive Lasso. In Figure 9, the left graph shows the increasing computational time when the sample size is increased, and the right graph shows the increasing computational time, when the variation of data is increased. The Predictive Lasso costs the least time in computation, when the data volume is increased and uncertainty in the data is increased. This feature perfectly meets the need of large volume and high fluctuation from the NetFlow measurements.

The simulation examines the two advantages of the Predictive Lasso. Firstly, under Gaussian model and Poisson model, the results show that the Predictive Lasso has much smaller prediction error than the Estimation Lasso and the Backward-Forward Selection. Secondly, the computational complexity listed in the equations and the simulation results both show that the magnitude of the computational time by the Predictive Lasso is many times less than the other two methods.

5. Summary of Discoveries and Discussion

Large scientific data movements require efficient utilization of the network bandwidth. Network performance prediction helps scheduling and estimation of the large network usage. Some of challenges in the prediction of large data movement are the computational cost and the large number of features in the data. The conventional methods such as Estimation Lasso and the Backward-Forward Selection are very computationally costly. Computational complexity is $O(npk)$ for Estimation Lasso shown in (25) and $O(np \times \sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij})$ for Backward-Forward Selection shown in (26), thus may not handle large data set with n observations and p dimensions. To solve this problem, we developed an efficient statistical method, the Predictive Lasso which is scalable for the big data. The computational complexity of the proposed Predictive Lasso is $O(np)$ shown in (24). Hence it enables the Predictive Lasso to handle large volume of data.

Moreover, large data sets usually include multiple features. The features degenerate the input data set into numerous, smaller partitions. As the number of features grows, it becomes intractable to handle such large amount of individual partitions. To solve this issue, we proposed the GLMM with Predictive Lasso. It not only prevents the input data set degenerating by specifying common features, but also selects the models with best prediction accuracy.

We presented the analysis of network measurement data to predict the network traffic for efficient utilization of the network bandwidth for large scientific data transfers as well as capacity planning of the network infrastructure for the future bandwidth needs. Our Predictive Lasso combines the best prediction in GLMM and the efficient model selection of Lasso. The method is designed by minimizing the MSPE plus the L-1 penalty on the coefficients of fixed effects and random effects. Compared to the Estimation Lasso and Backward-Forward Selection, our method results the best prediction accuracy and the least computational costs, supported by the simulation study and real application on the NetFlow measurement data. In addition, we developed an innovative approach for selecting tuning parameters, based on dimensional modeling with bootstrapping. The Predictive Lasso method will be used to model the performance of the data flow to predict the network traffic bandwidth in support of efficient utilization of the network infrastructure.

Acknowledgements

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Jiming Jiang's research is partially supported by the NSF grants SES-1121794 and NIH grant R01-GM085205A1. Jaesik Choi's research is partially supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2014R1A1A1002662), the NRF grant funded by the MSIP (NRF-2014M2A8A2074096). We thank Chris Tracy, Jon Dugan, Brian Tierney, Inder Monga and Gregory Bell at ESnet; Arie Shoshani, Joy Bonaguro and Jay Krous at LBNL; Richard Carlson at Dept. of Energy; Constantine Dovrolis at Georgia Tech; and Demetris Antoniades at University of Cyprus for their support.

References

Antoniades, D., Hu, K., Sim, A., & Dovrolis, C. (2013). What SNMP Data Can Tell Us about Edge-to-Edge Network Performance. *Passive and Active Measurement Conference 2013.* 267-269. http://dx.doi.org/10.1007/978-3-642-36516-4_30

Bondell, H. D., Krishna, A., & Ghosh, S. K. (2010). Joint variable selection for fixed and random effects in linear mixed effect models. *Biometrics., 66*, 10691077 http://dx.doi.org/10.1111/j.1541-0420.2010.01391.x

Box, G. E. P., & Tiao, G. C. (1981). Modeling Multiple Time Series with Applications. *Journal of the American Statistical Association,* 376-76.

Box, G. E. P., & Tiao, G. C. (1981). A canonical analysis of multiple time series. *Biometrika,* 64-2.

Box, G. E. P., & Pierce, David A. (1970). Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association.* 65-332

Bloomfield, P. (1972). An exponential model for the spectrum of a scalar time series. *Biometrika* 60-2.

Cisco Systems Inc. (2007). *NetFlow Services and Applications - White paper.*

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society.* 1-38.

Geweke, J. (1982). Measurement of Linear Dependence and Feedback between Multiple Time Series. *Journal of the American Statistical Association,* 378-77.

Hu, K., Choi, J., Jiang, J., & Sim, A. (2013). Best Predictive GLMM using LASSO with Application on High-Speed Network. *LBNL Tech Report.* 6327E.

Hu, K., Sim, A., Antoniades, D., & Dovrolis, C. (2013). Estimating and Forecasting Network Traffic Performance Based on Statistical Patterns Observed in SNMP Data. *Machine Learning and Data Mining Conference 2013,* 601-615. http://dx.doi.org/10.1007/978-3-642-39712-7_46

Ibrahim, J. G., Zhu, H., Garcia, R. I., & Guo, R. (2011). Fixed and Random Effects Selection in Mixed Effects Models. *Biometrics, 67*, 495-503. http://dx.doi.org/10.1111/j.1541-0420.2010.01463.x

Jiang, J., Nguyena, T., & Rao, J. S. (2011). Best Predictive Small Area Estimation. *Journal of the American Statistical Association, 106*, 494, 732-745. http://dx.doi.org/10.1198/jasa.2011.tm10221

Jiang, J., Rao, J. S., Gu, Z., & Nguyena, T. (2008). Fence methods for mixed model selection. *Annals of Statistics, 36-4*, 1669-1692. http://dx.doi.org/10.1214/07-AOS517

Lee, Y., & Nelder, J. A. (1996). Hierarchical Generalized Linear Models. *Journal of the Royal Statistical Society.* 58-4.

Percival, D. B., & Walden, A. T. (2000). Wavelet Methods for Time Series Analysis. *Cambridge University Press.*

Stallings, W. (1999). *SNMP, SNMthp2, SNMPv3 and RMON 1 and 2.* Addison-Wesley.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B.* 58,267288.

Appendix: Deriviation of Conditional Expectation under Poisson Case

Under the Poisson case of GLMM, the overall MSPE can be expressed as

$$\text{MSPE}(\check{\theta}) = E\left(\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\right) - 2\sum_{i=1}^{m} E\left(h_{M,i}(\psi, y_i)\theta_i\right) + \sum_{i=1}^{m} E(\theta_i^2)$$

Utilizing the property of Poisson distribution, the second part of MSPE can be written as the following

$$\begin{aligned} E\left(h_{M,i}(\psi, y_i)\theta_i\right) &= E[\theta_i E\left(h_{M,i}(\psi, y_i)|\theta\right)] \\ &= \sum_{k=0}^{\infty} h_{M,i}(\psi, k) E(e^{-\theta_i}\theta_i^{k+1}/k!) \\ &= \sum_{k=0}^{\infty} h_{M,i}(\psi, k)(k+1) E\left(e^{-\theta_i}\theta_i^{k+1}/(k+1)!\right) \end{aligned} \tag{27}$$

where $\theta = (\theta_i)_{1\le i\le m}$. Furthermore,

$$E\left(e^{-\theta_i}\theta_i^{(k+1)}/(k+1)!\right) = E\left(1_{(y_i=k+1)}\right).$$

Thus, with $h_{M,i}(\psi, -1) = 0$,

$$\begin{aligned} E\left(h_{M,i}(\psi, y_i)\theta_i\right) &= E\left(\sum_{k=0}^{\infty} h_{M,i}(\psi, k)(k+1)1_{(y_i=k+1)}\right) \\ &= E\left(h_{M,i}(\psi, y_i - 1)y_i\right) \end{aligned} \tag{28}$$

And the overall MSPE is

$$\text{MSPE}(\check{\theta}) = E\left(\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i + \sum_{i=1}^{m} E(\theta_i^2)\right)$$