Best Predictive GLMM using LASSO with Application on High-Speed Network

Kejia Hu^{1,2}, Jaesik Choi¹, Jiming Jiang², Alex Sim¹

¹Lawrence Berkeley National Laboratory, Berkeley, CA, USA ²University of California at Davis, Davis, CA, USA



Disclaimers

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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 its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Best Predictive GLMM using LASSO with Application on High-Speed Network

Kejia Hu^{1,2}, Jaesik Choi¹, Jiming Jiang², Alex Sim¹ ¹Lawrence Berkeley National Laboratory, Berkeley, CA, USA ²University of California at Davis, Davis, CA, USA

kjhu@lbl.gov, jaesikchoi@lbl.gov, asim@lbl.gov, jiang@wald.ucdavis.edu

ABSTRACT

Efficient data access is essential for sharing massive amounts of data with many geographically distributed collaborators. Better data routing and transfers are possible for large data transfers by accurate estimations of the network traffic performance with a probabilistic tolerance. Such estimations become non-trivial when amounts of network measurement data grow in unprecedented speed and volumes, and misspeicifed models are given. We present a statistical prediction method for network traffic performance by analyzing network traffic patterns and variation with the network conditions via the Generalized Linear Mixed Models (GLMMs), which relax the distributional assumption to that only involving the mean and variance of the errors. The method allows "borrowing strength" in the data by adopting fixed effects for shared relationship, random effects for subject variation, and errors for additional unexplained variation. The main contributions of the proposed method include: (1) best prediction accuracy even under a model misspecification; and (2) least computation time among all existing predicative algorithms under the GLMM setting.

Keywords

Generalized Linear Mixed Models; Mean squared prediction error (MSPE); Model misspecification; LASSO regularization

1. INTRODUCTION

The analysis of network traffic is getting more and more important today to efficiently utilize the limited resources offered by network infrastructure and wisely plan the large data transfers. Estimating the network traffic for a given time window with a given probabilistic tolerance error enables better data routing and transfers, which is particularly important for large scientific data movements. Short-term prediction of network traffic guides the several immediate scientific data placements. Long-term forecast of network traffic evaluates the performance of network and enables the capacity planning of the network infrastructure up to the future needs.

The Best Predictive GLMM (Generalize Linear Mixed Model) using LASSO fits tightly with the needs of quick generating accurate prediction with massive amount of network data from the following perspectives.

First, network data usually has a complicated data structure rather than a single time series and shows a complicated distribution model. For example, the Netflow data used in this study is a multivariate data that a single observation is composed of 12 variables with different types such as categorical, continuous, discrete and time series. Also, Netflow data is multiple time series data that measures both start time and end time of each data transfer. Moreover, it is a network data that shares some of features over the whole network as well as holds unique features for each path. Previous researches about network traffic highlight the usage of ARIMA model [1, 2] and spectral analysis [3, 4]. Those models are feasible with single time series data and complicated yet still manageable with multiple time series. However, it cannot fit data in one model with time series as well as other types of variables, representing the unique as well as the shared network structure.

However, GLMM allows us to represent the complicated multivariate structure of Netflow data in the following models:

$$y = g(x\beta + Zv) + e \tag{1}$$

For example, model (1) can be used to model how the duration y of one data transfer is related to multiple explanatory variables in the right hand side of the equation such as transfer size (in bytes), the number of packets it contains, the time that the transfer was started and also which path the transfer goes through.

By using more information from the dataset rather than a single time series, we can generate more accurate prediction by *"borrowing strength"* from other explanatory variables happened in the same network condition.

Since LM (Linear Models) and GLM (Generalized Linear Models) can also handle multivariate cases, a question may arise about the advantage of GLMM against LM and GLM. LM and GLM are just special cases of GLMM, and the following formula shows their relationship clearly:

GLMM:
$$y = g(x\beta + Zv) + e, E(e) = 0, Var(e) = \Sigma$$

GLM: $y = g(x\beta) + e, E(e) = 0, Var(e) = \Sigma$, indicating GLMM with 0 random effects

LM: $y = x\beta + e, e \sim N(0, \Sigma)$, indicating GLMM with 0 random effects, and limiting the error following a Gaussian Distribution.

It's obvious that GLMM is more general and flexible than GLM and LM at least in two terms: 1. It releases the Gaussian assumption of error term, and closely represents the real network distribution by modeling it generally with E(e) and Var(e). 2. It adopts the random effect that particularly addresses the subgroup variance in the data, and it quite fits the real situation of the Netflow data that in different transfer path the relationship is different as shown in Figure 1.



Figure 1: 3D plot of Bytes (Transfer Size), Duration and link (path for the data transfer) of Netflow data.

In Figure 1, it illustrates the relationship among Bytes (Transfer Size), Duration and link (path for the data transfer) in Netflow data. Overall, when the transfer size is large, it takes longer to finish the transfer. So the plot suggests an upward slope between Bytes and Duration. However, it is noticeable for different links that the slope rate is different. This means that Netflow data cannot be simply represented by LM and GLM considering only the universal relationship, but it should be modeled with mixed effects by using GLMM to capture shared features among network as well unique feature on different path.

Back to model (1), a GLMM for the relationship in Figure 1 can be written as

$$y = x\beta + Zv + e$$

where y is the observed duration time;

x represents fixed effects such as the transfer size of the data. The influence of bytes x to duration y is shown by β and in this case β is positive so larger bytes leads to longer duration;

Z represents the path for the data transfer. It is an index matrix that for each column only the ith position will be nonzero when the ith path is used for transfer;

v represents random effects that measure the different variation rate of duration when transferring data through different link. So var(v) = Ψ estimates the variation led from different transfer path;

e is the universal variance that measures the background noise in the network, E(e) = 0, $Var(e) = \Sigma$.

Up to this point, it is shown that GLMM properly models the network data with complicated data structure and distribution with its mixed effects and generalized distribution assumption.

On another issue, network data with multivariate data structures are usually high-dimensional and massive amount in size, and thus the algorithm should be efficient in handling a large amount of data. For example, the Netflow data has 18 millions of observations in 9 months and each observation has 12 variables which interact with each other. Thus GLMM alone is not sufficient for efficient selection of the final model, and we include LASSO [11]. LASSO selects the model using L1-penalty such as $|\beta|$ in the following expression. The simplest usage of LASSO is to select the significant variable based on likelihood and penalty:

$$\hat{\beta} = argmin\{|y - x\beta|^2 + \lambda|\beta|\}$$

It is proved that insignificant variables will be discarded from the final model by shrinking its corresponding coefficient in $\beta = 0$ while the significant ones are kept in the model with its

estimates $\hat{\beta}$ derived from the above expression. LASSO has a well-known advantage that it fits and selects the final model at the same time. However there are more benefits of LASSO in this study. It dramatically decreases the time taken to reach the final model. Firstly, since LASSO reaches the final model in one step with its shrinkage-to-0 property, it saves time from traditional stepwise model selection that takes several rounds of computation until the final model is obtained. Secondly, the time is saved by reducing dimensions from the original complicated model. In section 2 and 3, the computational advantage is shown from both theoretical side and simulation work. In our study with Netflow data, the final model only requires 5 or 6 significant variables such as path, start time and transfer size to be calculated in the final computation.

In order to achieve our goal for predicting network throughput, a method with a good prediction and a quick response is needed. The traditional parameter estimation of GLMM always focuses on the likelihood function. However, our main interest is in the prediction of mixed effects, e.g., linear combinations of fixed and random effects, under a GLMM. As an example, a user wants to find out the duration of a data transfer with $6e^{+05}$ bytes through path #1.

$$\theta = x\beta + R'v$$

where x are the fixed effects of transfer size, and β measure the influence of transfer size on duration. R would be a vector to indicate with path used. In this case, R' = (1,0,...,0) and v records the variance from different links.

$$\theta = (6e^{+05})\beta + v_1$$
 with $v_1 \sim N(0, \psi_1)$

Therefore, our approach focuses on the predictive accuracy by minimization of the mean squared prediction error (MSPE) of the mixed effects θ . The final model is proved to have the best prediction accuracy from theoretical results and simulation results (see sections 2 and 3) as well as with the real Netflow data (see section 4).

A main contribution of this study is combining the ideas of best predictive estimation with LASSO in order to deal with three major features of today's network prediction problem: 1. Algorithm particularly developed for prediction accuracy. 2. Model with sufficient efficiency and effectiveness for the complicated structure of multivariable and multi-type data. 3. GLMM with LASSO in handling the high-dimensional big data.

2. Best Predictive GLMM using LASSO

This section defines a best predictive estimator based on GLMM using LASSO to do model selection. Moreover, we introduce two existing GLMM approaches: Estimation LASSO(LASSO selection for estimation purpose) and Backward-Forward Selection (Backward-forward Selection based on log-likelihood ratio test) to compare with our approach and Predictive LASSO (LASSO selection for prediction purpose). Following the comparison, two distinctive advantages of Predictive LASSO are highlighted: 1. Accurate Prediction immune to model misspecification error and 2. Fast computation speed.

2.1 Generalized Linear Mixed Models (GLMMs)

$$y = g(\mu + v) + e$$

where y is the observed response variable, $E(y) = g(\mu + v)$

 μ, v is respectively the fixed effects and random effects

e is the error term with E(e) = 0 and $Var(e) = \Sigma$

g(.) is called a link function and can take forms such as

log(.), logit(.), probit(.), expit(.) And the most common one is called identity link that $g(\mu + v) = \mu + v$.

Suppose that the true underlying model is $y = \mu + Zv + e$

where μ =E(y). Here E without a subscript represents expectation with respect to the true distribution, which may be unknown but is not model dependent.

Mixed Models represent the relationship in the data by two types of features: fixed effects β capture the general linear relationship among variables, and random effects v capture the different characteristic of subgroups. Random effects v are usually categorical variables in the data, and we want to know how the response variable y fluctuates while the subgroups change. One of its usages is to examine in Netflow data how data transfer's volume and speed fluctuate with a variance $\psi_i (i = 1, ..., n)$ with different choice of path for its transfer. The model estimates the general linear relationship from $\hat{\beta}$ and the variation led from the ith subgroup's behavior as $\psi_i = var(v_i^*)$. Thus for the ith group, the relationship can be written as

$$y_i = X_i \beta + Z_i^* v_i^* + e_i, i = 1, ... n$$

Where y_i is a vector of $(y_{i1,\dots,}y_{in_i})$, altogether n_i observations are collected. In the first term, X_i is a $n_i \times q$ matrix contains where each row corresponds to feature vector of one of n_i observations. β is a $q \times 1$ vector that measures the fixed effects of the q predictors on y_i and needs to be estimated. The second term represents random effects. v_i^* is an unobserved random effect. It measures the ith group variation/uncertainty. It has $E(v_i^*) = 0$ and $Var(v_i^*) = \psi_i$ which need to be estimated. Z_i^* is an incidence matrix that contains observed information of the group category with 0s and 1s. The last term, e is the error term that measures the universal variance/background noise in the dataset. It has E(e) = 0 and $Var(e) = \Sigma$. In other words, with GLMM, the data's uncertainty/variation source is modeled more deliberately. $Var(e) = \Sigma$ captures the universal variation in the dataset, and $Var(v_i^*) = \psi_i$ features the fluctuation led from subgroup.

The model is called unstandardized if $\psi_i \neq I$. In order to apply LASSO for the mixed effects, the model needs to be standardized first, and the transformation can be made with Cholesky decomposion on the covariance matrix of random effects. $\psi = D\Gamma\Gamma'D$ where D is diagonal matrix D = diag(d₁, d₂, ..., d_q) and Γ is lower triangular matrix with 1's on the diagonal.

In the standardized model where $Var(v_i^*) = I_q$, Z_i is transformed as

$$Z_{i} = Z_{i}^{*} D \Gamma, Z = Z^{*} \widetilde{D} \widetilde{\Gamma}$$

where $\widetilde{D} = I_m \otimes D$, $\widetilde{\Gamma} = I_m \otimes \Gamma$. The \otimes means kronecker product.

Our interest is prediction of a vector of mixed effects that can be expressed as

$$\theta = F'\mu + R'v \tag{2}$$

where F and R are known matrices. Selecting F and R enables us to predict any combination of fixed and random effects. For example, if we want to predict the influence of the jth predictor's fixed effect and ith group's random effect, we can choose F to be a vector with 1 on the jth position and 0s on the rest, and R to be a vector with 1 on the ith position and 0s on the rest. In Netflow data, if we want to know the behavior of data transfer from a particular source IP address i, we can set F = I and R as a vector with 1 on the ith position and 0 on the rest. Then θ will be the prediction of mixed effect of all fixed effects and random effect of the ith source IP address. And, if we want to estimate all the possible mixed effects, we can set F = R = I. Then $\theta = \mu + \nu$.

2.2 Mean Square Prediction Error (MSPE) under GLMM

Our focus is on the best prediction accuracy. Thus, the target for the minimization is MSPE (mean square prediction error) of θ .

$$MSPE(\check{\theta}) = E(|\check{\theta} - \theta|^2)$$
(3)

The best predictor (BP) can be expressed as

$$\begin{split} \check{\theta} &= \mathrm{E}_{\mathrm{M}}(\theta|\mathbf{y}) = F'X\beta + R'E_{M}(v|\mathbf{y}) \\ &= F'X\beta + \mathrm{R'Z'V^{-1}}(Y - X\beta) \end{split}$$

where E_M denotes expectation under the assumed model.

To simplify the notation, we write

 $B = R'Z'V^{-1}$, M = F' - B and H = Z'F - R

then the BP can be expressed in a slightly different way as

$$\hat{\theta} = E_{M}(\theta|y) = F'y - M(y - X\beta)$$
(4)

By (5) and (8), we have

$$\check{\theta} - \theta = H'v + F'e - M(y - X\beta)$$

Thus we have MSPE as

$$MSPE(\check{\theta}) = E(|\check{\theta} - \theta|^2) = E(|H'v + F'e - M(y - X\beta)|^2)$$
(5)

2.3 Best Fixed Effects predictors via LASSO

To learn the fixed effect, we need to obtain the BPE of $\hat{\beta}$ in the model (3) to minimize the expression inside the expectation of MSPE. To prevent over fitting, we utilize LASSO (least absolute shrinkage and selection operator) as a regularized least square. LASSO prefers models with small L₁ norm $|\hat{\beta}|$ while minimizing MSPE. That is, LASSO puts L₁ penalty for the estimates in order to shrinkage the coefficients of insignificant predictors to zero.

Recently, Jiang (2011) [10] shows that β is related to the parts of terms in MSPE. Here, we show that the claim is still valid for the GLMM with LASSO models.

$$MSPE(\check{\theta}) = E(|H'v + F'e|^2) - 2E((v'H + e'F)M(y - X\beta))$$
$$+E((y - X\beta)'M'M(y - X\beta))$$
$$= I_1 - 2I_2 + I_3$$

We see that I_1 does not related to β and neither is I_2 since

$$I_2 = E((v'H + e'F)M(y - X\beta))$$

= $E((v'H + e'F)M(y - \mu)) + E((v'H + e'F)M(\mu - X\beta))$
= $E((v'H + e'F)M(Zv + e))$

Thus the BPE of $\hat{\beta}$ is obtained by minimizing the expression inside the expectation plus the penalty on the coefficients, that is

$$\beta = \operatorname{argmin} Q(\beta) = (y - X\beta)'M'M(y - X\beta) + \lambda \sum |\beta_i|.$$

2.4 Best Random Effects predictors via LASSO

Now, we focus on learning random effects. In order to eliminate one random effect, it requires the corresponding row and column of the covariance matrix having all zero elements. Furthermore, due to the fact as a covariance matrix, it needs to be guaranteed to be at least non-negative definite after removing the zero row and column. In Bondell(2010) [9], it mentions the use of the Cholesky decomposition for random effect selection in Linear Mixed-effects Models. In this paper, we extend this approach under GLMM with a focus on the prediction.

The approach is to put L_1 penalty on the elements of the diagonal matrix D which are $(d_1, d_2, ..., d_q)$. When d_i shrinks to zero, the whole column and row of $\psi = D\Gamma\Gamma'D$ will be eliminated, and it still remains non-negative definite since it is a matrix quadratic product. With this setting, we can now using LASSO to select the random effect by shrinking the corresponding row and column in the covariance matrix of insignificant random effects into 0.

Again we break the MSPE into three components.

$$MSPE(\tilde{\theta}) = E(|H'v + F'e|^2) - 2E((v'H + e'F)M(y - X\beta))$$

+ $E((y - X\beta)'M'M(y - X\beta))$
= $I_1 - 2I_2 + I_3$
$$I_1 = E(|H'v + F'e|^2) = E((H'v + F'e)'(H'v + F'e))$$

= $E(v'HH'v + e'FF'e)$
= $tr(HH') + tr(FF'\Sigma)$
= $tr(Z'F - R)(Z'F - R)') + tr(FF'\Sigma)$
= $tr(Z'FF'Z - Z'FR' - RF'Z) + C_1$
Note that C_1 is not related to d.
$$I_2 = E((v'H + e'F)M(y - X\beta))$$

= $E((v'H + e'F)M(y - \mu)) + E((v'H + e'F)M(\mu - X\beta))$
= $E((v'H + e'F)M(Zv + e))$
= $E(v'HMZv + e'FMe)$
= $tr(HMZ) + tr(FM\Sigma)$
= $tr(Z'FF'Z - Z'FBZ - RF'Z + RBZ) + tr(F(F' - B)\Sigma)$
= $tr(Z'FF'Z - Z'FBZ'RF'Z + RBZ) - tr(FB\Sigma) + C_2$

Note that
$$C_2$$
 is not related to d.

Thus the BPE of \hat{d} is obtained by minimizing the expression inside the expectation plus the penalty on the coefficients, that is

$$\hat{d} = \operatorname{argmin} Q(d) = (y - X\beta)'M'M(y - X\beta) + \operatorname{tr}(2\text{HBZ} - HF'Z - Z'FR') - \operatorname{tr}(FB\Sigma) + \lambda \sum |d_i|$$

2.5 Two Major Advantages

We discuss the whole procedure to obtain the Predictive LASSO for fixed effect and random effect in the previous sections. To summarize, for fixed effects selection,

$$\widehat{\beta} = \operatorname{argmin} Q(\beta) = (y - X\beta)' M' M(y - X\beta) + \lambda \sum |\beta_i|$$

where $B = R'Z'V^{-1}, M = F' - B$ (6)

and for random effects selection

$$\hat{d} = \operatorname{argmin} Q(d) = (y - X\beta)'M'M(y - X\beta) + \operatorname{tr}(2\text{HBZ} - HF'Z - Z'FR') - \operatorname{tr}(FB\Sigma) + \lambda \sum |d_i|.$$

where
$$H = Z'F - R$$
 (7)

Now we compare this Predictive LASSO with two methods: one is the traditional approach Backward and Forward selection that estimate the model based on likelihood,

$$(\hat{\beta}, \hat{d}) = \operatorname{argmax} E_{M}[loglikelihood_{n}(x_{i}|\beta, d)]$$
 (8)

The other is the Estimation LASSO, which inputs LASSO selection into likelihood,

$\hat{\beta} = \operatorname{argmax}$	$\mathbb{E}_{M}[loglikelihood_{n}(x_{i} \beta, d)] + \lambda \sum \beta_{i} $	(9)
$\hat{d} = \operatorname{argmax}$	$E_{M}[loglikelihood_{n}(x_{i} \beta, d)] + \lambda \sum d_{i} $	(10)

Both methods are estimation-oriented, finding the estimates which maximize the likelihood while our algorithm is prediction-oriented, finding the estimates which minimize the prediction error. The MSPE of our approach should be at least as good as these two methods. Moreover, while the model is misspecified and uncertainty increases, the Predictive LASSO has much better prediction accuracy than these two as you will see in the simulation study.

2.5.1 Immune to misspecification error

The first advantage of Predictive LASSO is immune to misspecification error. In other words, even if the assumed model is not correctly set or different from the underlying true model, the Predictive LASSO still achieves satisfied prediction accuracy, while the penalized MLE and Backward-Forward estimators will be biased. This is caused by their minimization function.

For the Estimation LASSO, the estimates are obtained from the formulas (9) and (10). It is noticed that the optimization problem is related to the assumed model, since the target function E_M is the expectation under the assumed model. For the Backward-Forward Selection, in each step the estimates are obtained from the E_M function, which is the expectation under the assumed model as shown in formula (8).

Thus, the Estimation LASSO and Backward-Forward Selection are obtained by maximized expectation of likelihood function under the assumed model. When the assumed model is misspecified, the estimators are biased at the same time. However, the Predictive LASSO is obtained by minimizing the inside part of expectation and thus is not influenced by the model –related expectation any more, as shown in (6) and (7).

The unique derivation of Predictive LASSO makes it superior than penalized MLE and Backward-Forward estimators under the misspecification case.

2.5.2 Much fast computation speed

Another very satisfying achievement of Predictive LASSO is the dramatically reduced computational cost. In order to estimate the expectation of the likelihood function of the GLMM, we need to use the Monte-Carlo Expectation Maximization (MCEM) algorithm to iteratively update the likelihood function, due to the unobserved random effect. The procedure is first to generate the unobserved random effect based on the initial guess of the parameters, then to find the estimators of random effect and fixed effect that maximize the expectation of the likelihood, based on the new estimators to generate random effect again. The iteratively steps stop when it is converged. This MCEM algorithm is very time-consuming, and can take days to reach convergence due to the curse of dimensionality.

In the penalized MLE and Backward-Forward estimators, they all require MCEM to estimate the expectation of the likelihood of the assumed model, and it makes the efficiency of these two methods largely decreased. However, the Predictive LASSO does not require likelihood function calculation, and the simple minimization problem is much less time-consuming than the MCEM algorithm. The unique derivation of Predictive LASSO makes it desirable in achieving fast computational speed.

Time(Predictive LASSO) = Optimization×1;

Time(Penalized MLE) = MCEM \times n

= (Monte Carlo + Optimization)×n

n is the number of iteration before algorithm converges, $n \geq 5$

Time (Predictive LASSO) = MCEM× $\sum_{i=1}^{I} k_i \sum_{j=1}^{J_i} n_{ij}$

 k_i is the trails of reducing one dimension of the model

I is the number of steps before reaching to the selected model

 n_{ij} is the number of iteration before MCEM converges for the ith trails and the jth variables omitted.

 J_i is the number of remaining variables in the model at step i.

The cost for optimization and that for Monte Carlo are at the same level. Thus we can see that the penalized MLE needs to pay at least 2n times of the computation cost of Predictive LASSO. For the Backward-Forward selection, the cost is even many times more than the penalized MLE. Thus it is very clear that the Predictive LASSO is computationally advantageous.

3. Simulation

In the above section 2, Predictive LASSO is derived along with introduction of two existing approaches: Estimation LASSO and Backward-Forward Selection. The two main advantages of Predictive LASSO claimed in the section 2 are: 1. Prediction is immune to model misspecification error, and 2. much faster computational speed is supported by the simulation results. The simulation compares the efficiency and accuracy of the prediction among three approaches in terms of MSPE as well as computational cost.

The simulation data is generated from the true model,

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

$$i = 1, ..., N \text{ and } j = 1, ..., n_i, N = 20, n_i = 6.$$

$$Var(v_i) = sd_1^2, Var(v_2) = sd_2^2$$

$$\psi = \text{diag}(\text{sd}_1^2, \text{sd}_2^2), sd_1 = 3, sd_2 = 2$$

$$var(e_{ij}) = \text{sd}, \text{sd} = 1$$

Then, as in reality, a model is assumed with redundant fixed effect and mixed effects, and this model is misspecified.

$$y = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4} + \beta_5 x_{ij5} + \beta_6 x_{ij6} + z_{ij1} v_{1i} + z_{ij2} v_{2i} + z_{ij3} v_{3i} + z_{ij4} v_{4i} + z_{ij5} v_{5i} + z_{ij6} v_{6i} + z_{ij7} v_{7i} + e_{ij}$$

The assumed model is misspecified with redundant variables. There are altogether 4 redundant fixed effects and 5 redundant random effects. The simulation is carried out in two types: one is the best prediction accuracy for fixed effects, and the other is the best prediction accuracy for random effects.

For each type, under three scenarios, the three approaches are compared. The first scenario is that changing variance is considered because it is crucial to know increasing uncertainty/variation/noise in the network data, which reflects the high fluctuation existing in the real Netflow data. The Predictive LASSO compared to the other two approaches still holds lower MSPE and fast computation speed in generating future prediction. Since there are multiple ways to categorize the network data such as by the source/destination IP addresses, by the network path, by the time block of start transfer and by the level of transfer size, we need to examine whether Predictive LASSO still holds its two advantages no matter how the segmentation is designed. The second scenario is that changing observational size in each group is considered because it is important to know whether the Predictive LASSO is always better than the other two approaches no matter how much data is segmented into each sub group. The third scenario is that changing number of groups is considered to examine whether Predictive LASSO is always superior than the other two no matter how many subgroups are used to categorize the whole data.

3.1 Best Predictive Fixed Effects

First, the prediction accuracy is examined among three approaches: Estimation LASSO, Backward-Forward Selection, and Predictive LASSO, under three scenarios: changing variance, changing observational size in one group, and changing number of groups. The MSPE is the criterion to evaluate prediction accuracy in Table 1.

Table 1. MSPE with Increasing Variance

Variation	Est.LASSO	BF Selection	Pred.LASSO
Sd=3	1.08	0.83	0.023
Sd=5	3.09	2.71	0.026
Sd=13	8.12	8.12	0.028

In Table 1, with the increasing variance, it becomes harder to achieve a good prediction. For all three methods, the MSPE has an increasing trend with larger uncertainty in the data. However, among those three methods, Predictive LASSO performs better with the smallest MSPE. Also, while Estimation LASSO and Backward-Forward Selection become worse with larger variance, Predictive LASSO only slightly increases its MSPE. This suggests an advantage of our approach in the real-world applications since the variance usually are very large in the data transfers for a large volume of data over high-speed network, but it does not decrease MSPE for Predictive.

Table 2. MSPE with increasing observational size in a group

Sample Size (number of observations in one group)	Est.LASSO	BF Selection	Pred.LASSO
$120(n_i = 6)$	1.08	0.83	0.023
$240(n_i = 12)$	0.87	0.55	0.0019
$480(n_i = 24)$	0.54	0.48	0.0011

In Table 2, with the increasing observational size in one group, prediction is more likely to be accurate since there is more

information with the larger dataset. Even though all three methods improve their accuracy with the varying condition, Predictive LASSO certainly performs better. Furthermore, it is noticeable that the decreasing speed for prediction model via Predictive LASSO is much faster than the other two approaches. There are millions of observations in Netflow data, and even though data is segmented by seconds, there are thousands of observations within one second. An approach like Predictive LASSO greatly improves its accuracy with increasing sample size such as in network measurement data.

Table 3. MSPE with increasing groups

Sample Size (number of groups)	Est.LASSO	BF Selection	Pred.LASSO
120(N = 20)	1.08	0.83	0.023
240(N = 40)	0.64	0.82	0.0044
480(N = 80)	0.07	0.18	0.0015

In Table 3, with the increasing observational groups, prediction is more likely to be accurate since more samples mean more available information. Predictive LASSO not only performs better than the other two approaches with the minimum MSPE, but also reduces the MSPE fastest with increasing number of groups.

With more available information from network, the data volume is increasing from its length and its width. The length is increasing when network data can be monitored for a long period. For example, the sample data in our study is from the last one year with a 1/1000 second frequency. The width is increasing since different kinds of variables related to the network condition can be monitored and collected from network devices. For example, the sample data in our study has 10 variables for each observation and allows us to understand the network behavior from different perspectives/dimensions. Our approach in Predictive LASSO is a preferred method in analyzing network data due to not only its best prediction accuracy but also for its dramatic improvement in reducing MSPE when more data is available for modeling.

3.2 Best Predictive Random Effects

In this section, we examine the best prediction accuracy of random effects for the three approaches: Estimated LASSO, Backward-Forward Selection and Predictive LASSO. Similarly, we test the prediction accuracy based on MSPE under three varying conditions: changing variance, changing number of observational size in one group, and changing number of groups. One thing we note is that because the random effect contains more uncertainty than the fixed effect, its MSPE is usually larger.

 Table 4. MSPE with Increasing Variance

Variance	Est.LASSO	BF Selection	Pred.LASSO
Sd=3	284.31	246.15	241.63
Sd=5	907.80	961.11	904.88
Sd=13	2923.37	3139.13	2902.16

In Table 4, with the increasing variance, the prediction accuracy is getting worse. It is observed that for all three approaches, MSPE is increasing with rising uncertainty in the data. However, among the three approaches, our approach performs better than the other two with the smallest MSPE. In real-world applications with large variation like in the network measurement data, our approach in

Predictive LASSO performs better in random effect prediction with the higher prediction accuracy.

Table 5. MSPE with increasing observational size in group

Sample Size (number of observations in one group)	Est.LASSO	BF Selection	Pred.LASSO
$120(n_i = 6)$	284.31	246.15	241.63
$240(n_i=12)$	210.32	207.82	190.67
$480(n_i = 24)$	68.07	64.40	58.26

In Table 5, while the observational size in one group is increasing, more accurate prediction can be produced because of more available information. Even though all three approaches improve their accuracy with more information, prediction model via Predictive LASSO is the best among them in terms of the smallest MSPE. As described in the previous simulation regarding fixed effect, the network data transfer throughput for a large volume of data is needed to be predicted. Acquiring the lowest prediction error from Predictive LASSO is beneficial in analyzing and predicting the high-speed network.

Table 6. MSPE with increasing groups

Sample Size (number of groups)	Est.LASSO	BF Selection	Pred.LASSO
120(N = 20)	284.31	246.15	241.63
240(N = 40)	181.09	180.29	165.12
480(N = 80)	77.01	76.17	68.64

In Table 6, as the number of groups is increased in the simulation, the prediction accuracy is improved with more information in the dataset. Predictive LASSO performs better than the other two approaches with the minimum MSPE. With the increasing amount of accessible network measurement data, Predictive LASSO performs better with its best prediction accuracy, compared to the other two existing approaches.

3.3 Computation Cost

In sections 3.1 and 3.2, the advantage of Predictive LASSO with the accurate prediction is validated by the simulation results, and leads us to apply it in the real-world network analysis. The second advantage for the computation cost is again compared among three approaches: Estimation LASSO, Backward-Forward Selection and Predictive LASSO.

In this simulation, it is examined under two cases: increasing sample size and increasing variance, and the computational speed for Predictive LASSO is examined compared to the other two approaches. The increasing sample size is considered to explore the influence of the data volume on the computation cost. The increasing variance is considered to examine whether the computational advantage of Predictive LASSO always holds regardless of how large the uncertainty from the data is. These two cases are practical in real-world applications as the network data usually comes with a large variation in a massive amount.

3.3.1 Computation Cost with Increasing Sample Size First, the simulation compared three approaches with increasing sample size. The consumed time is recorded and plotted in Figure 2. For the three approaches, the computation cost is increasing overall because more calculation is intrigued by more samples.

Consistent with the theoretical computation cost in section 2, the Backward-Forward estimators are the slowest since it requires not only stepwise selection to reach the final model but also MCEM in each step with multiple candidate model comparison based on likelihood ratio. Thus the computation cost for Backward-Forward Selection is very large. The Estimation LASSO is faster due to the property of LASSO to fit and select the final model at the same time, and reduces the time because MCEM is only needed to be computed once. For the Predictive LASSO, it is the fastest approach since the only computation required is for the minimization and neither MCEM nor iterative steps is needed.



Figure 2. Consumed Time vs. Sample Size

When predicting the network performance, a large sample size is common. From the promising computation cost shown in Figure 2, our approach in Predictive LASSO is more beneficial than the other two approaches.

3.3.2 Computation Cost with Increasing Variation

The second comparison for the computation cost is with different levels of uncertainty. In general, with more uncertainties in the data, the calculation takes more time to locate the globally optimal point since there are more variations in the target function now. However, in Figure 3, it is observed that the least time consuming algorithm is Predictive LASSO since it doesn't require MCEM and iterative steps. Moreover, it is noticeable that Predictive LASSO is least influenced by the increasing uncertainties in the data.

High variation is very likely in the network measurement data. Maintaining fast computation speed for Predictive LASSO is more crucial in handling network measurement data. While the other two approaches perform worse as uncertainty increases, the Predictive LASSO ensures that even though high variation exists in the Netflow data, its computation cost won't increase too much.



Figure 3. Computation Cost vs. Uncertainty.

Overall, the simulation results support the two advantages of our Predictive LASSO approach, and are consistent with the theoretical derivation in section 2. Predictive LASSO performs the best prediction with the minimum MSPE, and requires the least computation cost. These advantages are even more obvious, when the sample size increases or variation increases in the collected data. With these two advantages, our Predictive LASSO approach optimizes the network prediction.

4. Application to Netflow Data

The sample Netflow data is from ESnet for one year. The data has the following features, and Predictive LASSO fits our needs to generate the best prediction.

First, Netflow data is multivariate, and the relationship needs to be identified between multiple predictors and the response variables. In this case, GLMM provides a suitable form to combine influence from many variables. Secondly, the data has a very complicated distribution for noise. GLMM compared to the linear regression relieves the restriction of Gaussian assumption by only requiring the knowledge of the first moment and second moment of the noise. Thus the generalization of this model fits the reality of the data. Thirdly, the data shows evidences of fixed effect and random effect. As in the Left plot of Figure 4, the 3D plot shows the relationship among the octets, link (path for the transfer) and duration of the transfer. We observed that for every link, the general trend is that larger size of transfer takes longer time to finish. This is an evidence of the fixed effect of bytes on duration that for larger value of octets, we expect longer duration. Furthermore, it is observed that for different links, the slope is different, which indicates that the relationship between octets and duration is different. Also, the spread and variation of data points on different links are not the same. Thus it suggests an evidence of random effect that for different links, the behavior of the relationship is varied, and the grouping effect should be considered in the model. In the Right plot of Figure 4, it shows the relationship among octets, start time and link. Again, it is observed that for different links, the behavior of data points' variation and range is different. It supports the needs of random effect in the model, and GLMM is our choice. Fourthly, the data has millions of observations, and the dimension of the model is very high. Within the data, the directly observed variables are the following 12:

- Time Group: Start time, End time, Duration
- Transfer Path: Start/Destination Ports, Start/Destination IP addresses, Start/Destination Interfaces
- Octets
- Packets
- Frequency of collection



Figure 4. 3D plots of Netflow data. (Left) Bytes in transfer size, duration and network link for the data transfer, (Right) Bytes in transfer size, duration and start time for the data transfer.

If we consider the interaction and nested terms for both fixed effects and random effects, the dimension of the model can reach as high as 30 or 40. Therefore, for these large dataset and high dimensions, the computational speed of the algorithm becomes very crucial. For example, if we need to forecast the next 1 minute behavior of the network performance but the algorithm requires days to reach the final result, then the result is no longer useful for generating instant prediction. The computational advantage of our Predictive LASSO approach satisfies the need for the short-prediction of network, and it can react much more quickly than other methods such as Estimation LASSO and Backward-Forward Selection as shown in our simulation study.

Finally, our goal is to predict the network performance in the future, and we require a model with high prediction accuracy which means the smallest MSPE. The best prediction accuracy from the Predictive LASSO approach against the other approaches satisfies our need for network performance prediction.

The Predictive LASSO approach enables the best network performance prediction from the large Netflow dataset with quick computational time and flexibility in the model.

We observed that there are 8 unique interfaces, and an index 1,2,...,8 is used to represent them. When a data transfer comes out from interface 1 and goes into interface 8 on a device, the interface path is denoted as 18. In our sample Netflow data, we had 10 interface paths, and a variable called *interface path* is composed of 10 possible values. Similarly, we used 1-25 to represent the 25 unique IP addresses in our sample Netflow data, and built a new variable called *IP path* with 15 possible values.

4.1. GLMM on Octets (Bytes)

The first model is to study the relationship between Octet (traffic volume in the network) and the timestamps and path. A prediction can be made based on the traffic volume at a certain future time point through a certain path, and can be beneficial to know in advance whether there will be a traffic congestion.

The model is $y = X\beta + Z_1v_1 + Z_2v_2 + e$

where y is a N×1 vector of the observed Octets, and N is the total number of observations.

x is a N×q matrix of the observed variables: Start time, End time, Duration and Packets. Time variables such as start time and end time are modeled with smoothing splines, since linear relationship between time variables and Octets are very rare, and certain curvy transformation is usually needed.

 β is a q×1 vector of fixed effect of the variables in x. v_1 is a 10×1 vector, the random effect, measuring the variation of traffic volume coming through a certain interface path.

 Z_1 is a N×10 incidence matrices of 0s and 1s to tell the corresponding interface path for each observation.

 v_2 is a 15×1 vector, the random effect, measuring the variation of traffic volume coming through a certain IP path.

 Z_2 is a N×15 incidence matrices of 0s and 1s to tell the corresponding IP path for each observation.

With this model, prediction of future network performance condition notifies us the congestion situation at certain time through a specific link. With this knowledge, we can avoid the slow data transfers, and we can also adjust the network infrastructure based on the long-term prediction of network performance behavior in order to optimize the allocation of the limited network bandwidth.

With Predictive LASSO, the selected predictive model for prediction of Octets is the following:

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

As mentioned above, time series variables are usually non-linear relative to the bytes. Thus we allow smooth spline transformation on the variables of start time and duration to model the possible curvy relationship. The smooth spline parameters are chosen by the model automatically based on the cross validation.

From our model, the significant fixed effects chosen by Predictive LASSO are start time, duration and number of packets. The significant random effect chosen by Predictive LASSO is IP path.

To address the influence of fixed effects, the coefficients of β_{start} , β_{pkt} and $\beta_{duration}$ along with the smoothing function $s(x_{start})$ and $s(x_{duration})$ are estimated.

Applying our Predictive LASSO approach to 1-minute snapshot of Netflow data, the sample has 2932 observations with start time ranging from 0 seconds to 60th seconds, 6 unique IP paths and 6 interfaces paths. In the following Table 7, Predictive LASSO provides the coefficient estimates, and shows that variables start time, duration and packets are very significant in predicting the octets. As expected, the duration and packets hold positive relationships with the transfer size. Large octets are usually connected with longer duration to complete transfer and also more packets in transfer.

Table 7. Coefficient estimation for fixed effects

Fixed effect	Estimates	std	p-value
Intercept	-507.4132	1063.23702	0.6
Start time	-417.6627	28.56410	8e-47
Duration	683.0074	34.00287	3e-84
Packets	8479.8077	66.55375	0

The Figure 5 shows the smoothing function, and it is observed that from the 1^{st} to the 30^{th} seconds the transfer size is getting larger and larger as time goes by. However, after the 30^{th} second, octets become smaller. This behavior shows the frequent changes in the network traffic volume. The smoothing function on duration suggests that a larger data transfer matches with a longer duration time even though there is a chance that the transfer takes longer time with sudden traffic congestion even for smaller transfers. As for packets, we observe that it always holds an upward relationship with octets that larger transfer size requires more packets.



Figure 5: Smooth function of the fixed effect

The random effects plot in the next Figure 6 shows that the traffic volume varies in different IP paths. In our sample data, there are

six paths: 83, 38, 14, 41, 16 and 61. The busiest path is observed at 83, and 66% of data transfers in our sample data is through this channel. The fluctuation rate on this path is also the highest with a standard deviation as high as 64946.04. The second largest variance source is through path 38 with standard deviation equal to 45446.48. From Figure 6, the uncertainty source for the data prediction is very clear. It is also observed that data transfers are easy to predict through IP path with low variance such as path 41 and path 14, but difficult to predict through IP path with high variance such as path 83, path 16, path 38 and path 61. Besides the uncertainty coming from each IP path, the background noise/universal uncertainty has a standard deviation of 17331.42.



Figure 6: standard deviation of the random effect

Comparing with the other two approaches, Estimation LASSO and Backward-Forward Selection, our Predictive LASSO approaches shows the best prediction accuracy and least computation time, as shown in Table 8.

 Table 8. Comparison of MSPE and Speed to Predict Traffic

 Volume (Octets)

	Est.LASSO	BF Selection	Pred.LASSO
MSPE	1.47e+5	1.64e+6	1.84e+4
Computation(in seconds)	3.56e+8	1.53e+12	2.23

4.2. GLMM on Duration

The second model is to predict the transfer duration for a certain size of the data at a selected path starting at certain time points. A prediction can be made based on the expected duration time of transfer on given paths and a selected starting time. The data transfers can be scheduled according to the prediction by selecting the path and starting time with the shortest predicted transfer time.

The model is $y = X\beta + Z_1v_1 + Z_2v_2 + e$

where y is a N×1 vector of observed duration, and N is the total number of observations.

x is a $N \times q$ matrix of the observed variables: Start time, Octets and Packets. Time variable such as start time will be modeled with smoothing splines since linear relationship is scarcely existed with time variable, and curvy trend is close to the reality.

 β is a q×1 vector of fixed effects of the variables in x.

 v_1 is a 10×1 vector, the random effects, measuring the variation of traffic volume coming through a certain interface path.

 Z_1 is a N×10 incidence matrices of 0s and 1s to tell the corresponding interface path for each observation.

 v_2 is a 15×1 vector, the random effects, measuring the variation of traffic volume coming through a certain IP path.

 Z_2 is a N×15 incidence matrices of 0s and 1s to tell the corresponding IP path for each observation.

With this model, the expectation of duration for a data transfer is obtained once the path and start time are chosen. This is especially useful when the data transfer needs to be completed within a certain period of time or to achieve a designated transfer speed. With the model, we select the path and start time that satisfy the time limits or speed requirement.

With Predictive LASSO, the selected predictive model for prediction of Duration is the following:

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

As mentioned above, time series variables are usually not linear relative to the bytes, and we allow smooth spline transformation on the variable of start time, and the smooth spline parameters are chosen automatically by the model based on the cross validation.

The significant fixed effects chosen by our Predictive LASSO approach are start time, octets and number of packets. The significant random effect chosen by Predictive LASSO is IP path.

To address the influence of fixed effects, the coefficients of β_{start} , β_{pkt} and β_{octets} along with the smoothing function $s(x_{start})$ are estimated. Applying our approach to 1-minute snapshot of Netflow data, the sample has 2932 observations with start time ranging from 0 seconds to 60th seconds, 6 unique IP paths and 6 interfaces paths.

In the following Table 9, Predictive LASSO provides the coefficient estimates, and shows that variables of start time, octets and packets are very significant in predicting the octets. As expected, the number of packets holds positive relationship with the duration that more packets require longer transfer time to be completed. The coefficient estimation for fixed effects is shown in Table 9.

Table 9. Coefficient estimation for fixed effects

Fixed effect	Estimates	Std	p-value
Intercept	-13.80878104	0.91421760	1e-49
Start time	0.57404659	0.01694813	2e-212
Packets	1.11494182	0.03472477	5e-194

The following Figure 7 shows the smoothing functions, and it is observed that from the 1st to the 20th seconds, the duration is getting larger as time goes by. However, from the 20th to the 40th second, duration is reduced probably due to less traffic within this time period. After the 40th second, more traffic volume is in the network, and data transfer takes longer time during this period. This behavior shows the frequent changes in the network traffic volume. As for packets, we observed that it always holds an upward relationship with duration that larger transfer size requires longer delivery time.



Figure 7: Smooth function of the fixed effect

The random effects plot in the following Figure 8 shows that the traffic volume varies in different IP paths. In our sample data, there are six paths: 83, 38, 14, 41, 16 and 61. The busier paths such as paths 83, 38, 41 and 14 have higher fluctuation rates in duration, while those paths with less traffic such as paths 16 and 61 have lower variations. From Figure 8, the uncertainty source of the data prediction is very clear. Besides the uncertainty coming from each IP path, the background noise/universal uncertainty has a standard deviation of 11.2392.



Figure 8: Standard Deviation of the random effect

The model suggests the importance of variation in random effect such as IP path in predicting duration. Besides the path, start time selection and assignment of packets are also significant in prediction of the duration.

Compared with the other two approaches in Table 10, Estimation LASSO and Backward-Forward Selection, our Predictive LASSO approach shows the best prediction accuracy and least computation time.

 Table 10. Comparison of MSPE and Speed to Predict

 Duration

	Est.LASSO	BF Selection	Pred.LASSO
MSPE	23.06	42.23	12.73
Computation(in seconds)	6.26e+7	5.43e+10	1.42

5. Conclusion

In this paper, a new approach Predictive LASSO, the best predictive GLMM with LASSO, is discussed. The computational algorithm of the best predictor is discussed in section 2. Our Predictive LASSO approach is more informative in prediction, since it is generated based on the analysis of all variables and their

interactions in the complicated structure of network measurement data. With focus on minimizing the Mean Squared Prediction Error (MSPE), we showed that the derived predictors are superior in acquiring high accuracy of forecast. Moreover, with the involvement of LASSO, the new approach is efficient in handling high dimensionality and large volume of data with fast computation speed. We showed simulations with three scenarios with changing variance, observational size in one group and number groups. Compared to two existing approaches, Backward-Forward Estimators and Penalized MLE, both theory and simulation support two distinguished advantages from our Predictive LASSO approach: 1. Accurate prediction without misspecification error, and 2. Fast computation speed to achieve final results. We also applied Predictive LASSO to Netflow measurement data in the form of two models. The two models have different interests in prediction: one is to predict the relationship between Octets (size of the traffic) and the timestamps and path, and the other is to predict the transfer duration for a certain size of the data volume for a selected path and a start time. This statistical prediction for network performance captures network traffic patterns and variation with the network conditions.

The advantages of our Predictive LASSO approach include that (1) Generalized Linear Mixed Models (GLMMs) fully enables utilization of the complicated structure of network data and relaxes the distributional assumption to that only involving the mean and variance of the errors, (2) "borrowing strength" in the data with sophisticate analysis of fixed effects for shared relationship, random effects for sub-group variation, and errors for background noise, (3) the best prediction accuracy even under a model misspecification, and (4) the least computation time among existing predicative algorithms under the GLMM setting.

6. ACKNOWLEDGMENTS

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.

7. REFERENCES

- Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825.
- [2] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
- [3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531.
- [4] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
- [5] Sannella, M. J. 1994. Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398, University of Washington.
- [6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.
- [7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In

Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10.

- [8] Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. J. Syst. Softw. 79, 5 (May. 2006), 577-590. Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33.
- [9] Bondell, H. D., Krishna, A. and Ghosh, S.K. (2010) Joint variable selection for fixed and random effects in linear mixed-effects models, Bioetrics 66, 1069-1077
- [10] Jiang, J., Nguye, T. and Rao, J.S. (2011a) Best predictive small area estimation, J. Amer. Statist. Assoc. 106, 732-745
- [11] Tibshirani, R. (1996), Regression Shrinkage and Selection via the LASSO, J. Roy. Statisti. Soc. B 58, 267-288
- [12] Hu, K., Sim, A., Antoniades, D. and Dovrolis, C. Estimating and Forecasting Network Traffic Performance based on Statistical Patterns Observed in SNMP data, MLDM 2013.