2016 IEEE 24th International Conference on Network Protocols (ICNP)
Workshop on Machine Learning in Computer Networks (NetworkML 2016)

Predicting Future Traffic using Hidden Markov Models

Zhitang Chen
Huawei Technologies
Shatin, Hong Kong
Email: chenzhitang2@huawei.com

Jiayao Wen
The University of Hong Kong
Pokfulam, Hong Kong
Email: jywen@hku.hk

Yanhui Geng
Huawei Technologies
Shatin, Hong Kong
Email: geng.yanhui@huawei.com

***Abstract*—Network traffic volume estimation and prediction is an important research topic that attracts persistent attention from the networking community and the machine learning community. Although there has been extensive work on estimating or predicting the traffic matrix using time series models, low rank matrix decomposition et. al, to the best of our knowledge, there is few work investigating the problem whether we are able to estimate and predict the traffic volume based on some statistics of the traffic which are much less costly to collect, for example, the flow counts. In this paper, we propose to model the relationship between the traffic volume and simple statistics about flows using a Hidden Markov Model based on which we can avoid direct measurement of the traffic volume but instead we estimate and predict the hidden traffic volume based on those simple flow statistics which are collected by some sketch techniques. We demonstrate the feasibility and effectiveness of our proposed method using some semi-simulation and real data experimental results.**

I. Introduction

Network traffic volume estimation and prediction is a very important research problem in networking. Accurate estimation and prediction of the traffic volume, especially the traffic matrix, is beneficial to network routing control, congestion control, network resource allocation and long term planning, and thus it has attracted extensive attention in the networking community and the machine learning community. There are mainly two main streams of research in existing works. The first main stream assumes that at any given time slot, the aggregated traffic volume (the number of bytes) transmitted between a certain source destination pair can be measured, and then a time series analysis model such as linear models including AR,ARMA,ARIMA,FARIMA [6], [7], [16], [14] and nonlinear models including ANN,RNN,GARCH [10], [17], [2], [8] is applied to predict the future traffic. The limitation of this category of approaches is that we need to directly measure the traffic volume of the previous time intervals in order to predict the traffic volumes for the future time interval. However, direct measurement of the traffic volume is too expensive to be feasible, especially in the large scale high speed network, and thus although this approach is simple but in practice, it is not well scalable. The other main stream of approaches is usually termed network tomography [3], [1], [9], [4] which is complementary to the first main stream approach. The idea of network tomography is to estimate the network traffic volume based on other observations such as the link utilizations. Link utilization is the aggregated traffic volume of those flows going through that link. Consequently, usually there is a deterministic linear system to describe the relationship between the link utilizations and the hidden traffic volumes. However, one of the fatal limitation of the network tomography approach is that the linear system is always underdetermined as the number of links is far less than the number of source destination pairs in a network. Recovering the hidden traffic volume from the limited amount of link utilizations is extremely difficult.

Realizing the strong limitations of existing works, in this paper, we investigate the possibility of inferring the hidden traffic volume based some flow statistics which are much easier to collect such as the number of flows in a certain time interval. To the best of our knowledge, our work is the pioneer work exploiting the dependence between the flow counts and the flow volume to estimate and predict the traffic volume. We propose to use a Hidden Markov Model to describe the relationship of the flow count and the flow volume and also the temporal dynamic behavior of both. We use the state-of-the-art algorithms such as Kernel Bayes Rule (KBR) and Recurrent Neural Network (RNN) with Long Short Term Memory unit(LSTM unit) to train the model and apply the model to predict the future traffic.

In the rest of this paper, in Section II, we discuss the existing works which are contributed to estimating and predicting the network traffic; in Section III, we propose our hidden Markov model to estimate and predict the future traffic using Kernel Bayes Rule [5], [15] as well as Recurrent Neural Network; in Section IV, we conduct experiments using semi-simulated data and real network traffic data; in Section V, we conclude this paper.

II. Network Traffic Estimation and Prediction

In the past decades, there are large amounts of works published to solve the network traffic estimation and prediction problem. As discussed in the introduction, those works are mainly divided into two main categories. One is to assume that we are able to observe the network traffic aggregated in sequential time intervals and we build mathematical model to predict the future traffic in a rolling way[6], [7], [16], [14], [10], [17], [2], [8]. The other category of methods termed network tomography avoid direct measurement of network

978-1-5090-3281-5/16/$31.00 ©2016 IEEE

traffic transmitted between any two end hosts in which we are interested, but instead try to recover the hidden traffic volume using some link utilizations. In this section, we will briefly discuss the formulations as well as the limitations of these two categories of methods.

A. Rolling Prediction Using Past Observations

This category of methods [3], [1], [9], [4] assume that we are able to observe the traffic volume in a sequential way. Our goal is to predict the future traffic based on the past observations. The foundation of this category of methods is the self-similarity of the network traffic. In general, we can use the following formulation to describe the prediction process:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \cdots, \mathbf{x}_{t-p+1}, \boldsymbol{\epsilon}_t, \cdots, \boldsymbol{\epsilon}_{t-q+1}) + \boldsymbol{\epsilon}_{t+1}, \quad (1)$$

where $\mathbf{x}_t$ is the traffic volume transmitted at time t of those source destination pairs in which we are interested, $\boldsymbol{\epsilon}_t$ is the prediction error at time t, p is the number of past observations that are used for prediction and q is the number of past prediction errors that are used to correct the prediction.

The training phase is to learn a best function that minimizes the prediction error as follows:

$$f^* = \underset{f}{argmin}\, \mathbb{E}[(\mathbf{x}_{t+1} - f(\mathbf{x}_t, \cdots, \mathbf{x}_{t-p+1}, \boldsymbol{\epsilon}_t, \cdots, \boldsymbol{\epsilon}_{t-q+1}))^2].$$

Many models are used to approximate f. A simple case is the linear model such as ARMA and its variants like ARIMA and FARIMA. In ARMA models, the relationship between the predictor and the target variable is simply described using a linear model as follows:

$$\mathbf{x}_{t+1} = \sum_{i=0}^{p-1} \alpha_i \mathbf{x}_{t-i} + \sum_{j=0}^{q-1} \beta_j \boldsymbol{\epsilon}_{t-j} + \boldsymbol{\epsilon}_{t+1}, \quad (2)$$

where α_i and β_j are coefficients that can be easily learnt by Least Square Regression.

Linear models are easy to implement and have good interpretation and thus are widely used in many real work time series analysis problems. However, linear models are shown not sufficient to describe some nonlinear behaviors of the network traffic. To make the model more flexible, there are also works using Artificial Neural Network (ANN) to approximate the nonlinear function f.

ANN is a very powerful nonlinear function approximator given sufficient number of hidden neurons. As illustrated by Fig. 1, we feedward the past observations and the prediction errors to the neural network and the neural network outputs a predicted future traffic volume. The training phase of the neural network is to adjust the weights of connections between two adjacent layers of neurons in order to minimize the prediction errors. Back prorogation using batch gradient descend or stochastic gradient descend is commonly used to train a neural network.

Although the idea of rolling prediction using the past observations looks simple and efficient, the main limitation is that one should be able to collect the traffic volumes in sequential time intervals which could be very expensive especially in a large scale high speed network. To avoid direct measurement of the traffic volumes, there are methods termed network tomography proposed to estimate the traffic volume from link utilization data and then use the estimated traffic volume to do prediction as described in the next section.

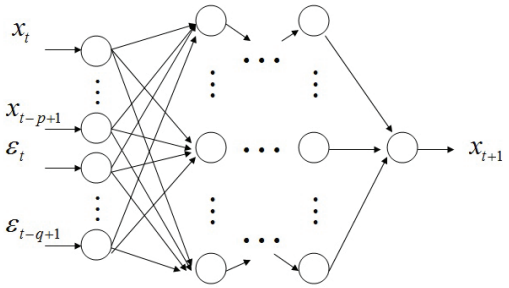

Fig. 1: Artificial Neural Network

B. Network Tomography

The idea of network tomography is to exploit the relationship between the link load data and the traffic demand among the end hosts of the network. Denote by $\mathbf{X}_t$ a vector collecting all traffic volumes transmitted at time slot t between any two end hosts of the network and denote a routing matrix by $\mathbf{A}$ containing the routing information, i.e. $\mathbf{A}_{i,j} = 1$ means that the link i belongs to the path that the source destination pair j used to transmit their traffic; otherwise $\mathbf{A}_{i,j} = 0$. We also denote by $\mathbf{Y}_t$ a vector collecting all link loads at time slot t. Then we can formulate the relationship between the link load and the traffic volume by the following linear system:

$$\mathbf{Y}_t = \mathbf{A}\mathbf{X}_t, \ \forall t. \quad (3)$$

Note that we assume during the observation period, the routing matrix is not changed. Even with this assumption, the system described in Eq. 3 is very difficult to solve because the system is highly under-determined since the number of link is usually far less than the number of end host pairs in a network.

There has been extensive research using compressive sensing, expectation-maximization (EM) algorithms to solve this system. In general, those algorithms are complicated and the results are not satisfying.

Realizing the strong limitations of those existing works, we propose a new framework of estimating the traffic volume based on some simple flow level statistics using Hidden Markov Models in the following section.

III. Inferring and Predicting Traffic Volume Using HMMs

In this section, we discuss the possibility of inferring and predicting traffic volume based on some simple flow level statistics. This idea is based on the observation that there exists strong statistical dependence between those simple flow level statistics and the total traffic volume as illustrated by the

following figure which is obtained by analyzing a time series of real network traffic.

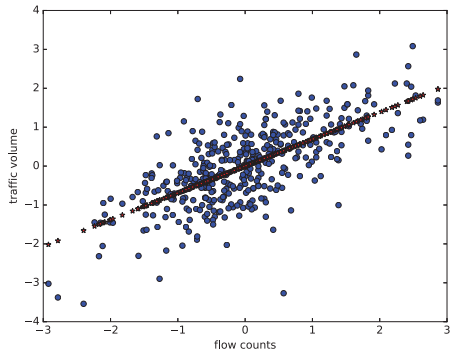

Fig. 2: Relation between flow counts and flow volumes

Fig. 2 shows the relationship between the flow numbers at different time slots and the corresponding total flow volumes (in terms of bytes). Here we normalize the time series such that they have zero means and standard deviations. We can see that it is possible to exploit this relationship to infer the volume based on the flow count as there is significant correlation between the flow count and the traffic volume.

A. Simple Flow Statistics and Sketch Technique

We argue that collecting the simple flow statistics is much less expensive than direct measurement of the traffic volume. Here we can define the flow statistics as the following statistics:

- $C_{f,t}$ the number of flows at time interval t;
- $C_{tcp,t}$ the number of TCP flows at time interval t;
- $C_{R(i),t}$ the number of flows using a port number falling into the range $R(i),\ \forall i$ at time t.

There is no limitation of using more information that are beneficial to the estimation of the traffic volume in addition to those we define here. We put all flow statistics which are easy to collect into a vector of observations $\mathbf{y}_t = [C_{f,t}, C_{tcp,t}, C_{R(i),t}, \cdots]^T$.

Collecting those flow level statistics is by nature a problem of counting distinct items is a data stream, which has been extensively studied in literature.

B. Hidden Markov Models

Hidden Markov Models are commonly used time invariant state-space models as follows:

$$p(\mathbf{X}, \mathbf{Y}) = \pi(\mathbf{x}_0) \prod_{i=0}^{T} p(\mathbf{y}_i|\mathbf{x}_i) \prod_{i=0}^{T-1} p(\mathbf{x}_{i+1}|\mathbf{x}_i), \tag{4}$$

where $\mathbf{x}_i$ is the hidden variable and $\mathbf{y}_i$ is the observed variable, $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ is the transition probability which describes the dynamic behavior of the system and $p(\mathbf{y}_i|\mathbf{x}_i)$ is the emission probability which describes the how the system generates the observation based on the hidden variable. In our problem, the traffic volume is the hidden variable $\mathbf{x}_i$ and the flow statistics is the observed variable $\mathbf{y}_i$ and thus $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ describes how the traffic volume changes along time and $p(\mathbf{y}_i|\mathbf{x}_i)$ describes the relationship between the traffic volume and the flow statistics such the the flow count.

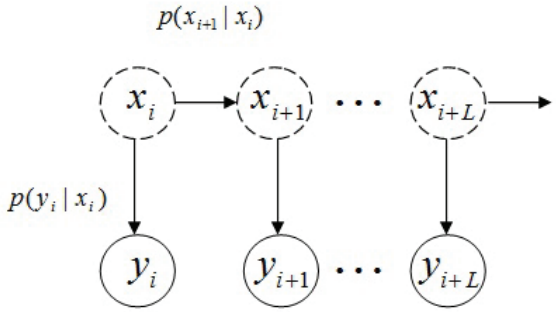

Fig. 3: Hidden Markov Model

Generally, $p(\mathbf{x}_{i+1}|\mathbf{x}_i)$ and $p(\mathbf{y}_i|\mathbf{x}_i)$ are unknown and thus we need to estimate them either by approximation using some parametric approaches or learning it from data. In this paper, we assume that we are able to collect some training data $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_1, \mathbf{y}_1, \cdots, \mathbf{x}_L, \mathbf{y}_L)$ such that we are able to learn the transition probability and the emission probability. In the networking problem, we assume that given a limited amount of budget of storage and measurement capability, we can collect some packet traces which enable us to learn the how the traffic volume evolves and the relationship between the volume and the flow statistics such as the flow count. This is done once and for the future estimation and prediction, we only need to collect the flow statistics based on which we infer and predict the traffic volume.

Suppose we have a sequence of new observed flow statistics $(\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \cdots, \tilde{\mathbf{y}}_t)$, we would like to infer the corresponding hidden variable $\tilde{\mathbf{x}}_t$, i.e.

$$\tilde{\mathbf{x}}_t \sim p(\mathbf{x}_t|\tilde{\mathbf{y}}_t, \cdots, \tilde{\mathbf{y}}_0), \tag{5}$$

furthermore, we would like to predict the future traffic volume:

$$\tilde{\mathbf{x}}_{t+1} \sim p(\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t, \cdots, \tilde{\mathbf{y}}_0). \tag{6}$$

Here we follow Kernel Bayes Rule to obtain a point estimation of $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{x}}_{t+1}$. The basic idea is to embed the transition probability and the emission probability into the RKHS as the condition mean embeddings as follows:

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t) \mapsto \hat{C}_{X_{+1}X}(\hat{C}_{XX} + \epsilon I)^{-1}, \tag{7}$$

$$p(\mathbf{y}_t|\mathbf{x}_t) \mapsto \hat{C}_{YX}(\hat{C}_{XX} + \epsilon I)^{-1}, \tag{8}$$

where

$$\hat{C}_{UV} = \frac{1}{L} \sum_{i=1}^{L} k(\cdot, U_i) \otimes k(\cdot, V_i),$$

where $k(\cdot,\cdot)$ is a kernel function such as the square exponential function $k(x,x') = \exp(-(x-x')/\sigma^2)$. Suppose we embed $p(\mathbf{x}_t|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0)$ into the Reproducing Kernel Hilbert Space [11] as the conditional mean embedding $\hat{m}_{\mathbf{x}_t|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0}$ [13], [12] , then we can estimate $\tilde{\mathbf{x}}_t$ by finding the value that minimizes the following objective:

$$\tilde{\mathbf{x}}_t = \underset{\mathbf{x}}{argmin}\, \|k(\cdot,\mathbf{x}) - \hat{m}_{\mathbf{x}_t|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0}\|^2_{\mathcal{H}}.$$

We can further embed $p(\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0)$ into the Reproducing Kernel Hilbert Space as the conditional mean embedding $\hat{m}_{\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0}$ using the following Kernel Bayes Rule as follows:

$$\hat{m}_{\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0} = \hat{C}_{X_{+1}X}(\hat{C}_{XX} + \epsilon I)^{-1}\hat{m}_{\mathbf{x}_t|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0},$$

since

$$p(\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0)d\mathbf{x}_t$$

and in RKHS, the integration simply reduces to matrix multiplication. Similarly, we can predict $\tilde{\mathbf{x}}_{t+1}$ by finding the value that minimizes the following objective:

$$\tilde{\mathbf{x}}_{t+1} = \underset{\mathbf{x}}{argmin}\, \|k(\cdot,\mathbf{x}) - \hat{m}_{\mathbf{x}_{t+1}|\tilde{\mathbf{y}}_t,\cdots,\tilde{\mathbf{y}}_0}\|^2_{\mathcal{H}}.$$

The computational complexity of KBR is $\mathcal{O}(n^3)$ where n is the training sample size. The computational complexity can further be reduced to $\mathcal{O}(nr^2)$ where $r << n$ is the cardinality of the subset of regressors. For more details, please refer to [5], [15].

An alternative is to use Recurrent Neural Networks to predict the future traffic volume based on the flow counts. Different from feedforward neural networks in which there

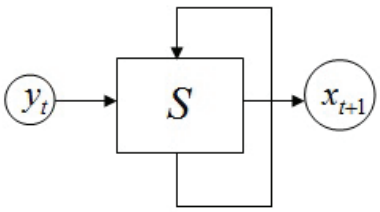

Fig. 4: Future traffic prediction using RNN

is no interconnection among the neurons in the same hidden layer, RNNs have interconnected neurons in the same hidden layers which means that RNNs maintain internal memory to store previous states of neurons and feed the previous states of those neurons to themselves in addition to the inputs from the previous layer such that it is very suitable to process arbitrary sequences of inputs and thus applicable to tasks such as time series analysis.

In our problem, we only need to train a RNN which takes the flow count at any time interval t and then output an estimate of the future traffic volume at time interval $t+1$. The objective of the train phase is to learn the optimal weights of connections in the RNN such that the prediction error is minimized. Deep recurrent neural network could be used if necessary.

IV. Experiment

In this section, we conduct experiments using semi-simulated data and real network traffic data to demonstrate the feasibility of inferring and predicting network traffic volume based on simple flow statistics such as flow counts. In the following experiments, we normalize both time series such that they have zero means and standard deviations.

A. Semi-Simulation

In this section, we conduct semi-simulation where we use the public benchmark data named 2004 Abilene data from the Internet[1].

This dataset contains 24 weeks of 5 minute averages for 12 routers (12×12 matrices). In this experiment, we only use the traffic of the $(3,3)$-th entries of the matrix .

Denote by x_t the traffic volume at time interval t and y_t the corresponding flow count. We generate the flow counts from the traffic volume using the following mechanisms (conditional distributions) which exhibit certain kinds of nonlinearity and stochasticity.

- M1: $y_t = 0.01 * (x_t + 0.05 * \xi_1 + 0.05 * \xi_2)$,
- M2: $y_t = 0.01 * ((x_t)^{0.1} + 0.25 * \xi_1 + 0.25 * \xi_2)$,

where ξ_1 follows standard Gamma distribution and ξ_2 follows standard Gaussian distribution.

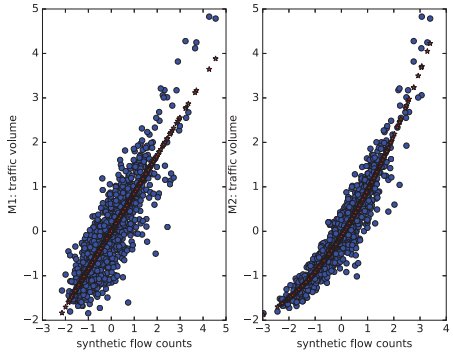

Fig. 5: synthetic flow count vs traffic volume

We conduct the experiments under different settings of training and testing sample size (S_{tr}, S_{tst})=(200, 700), (300, 600), (400, 500). The results are summarized in the following tables for both M_1 and M_2.

Table I and II show the Mean Square Error of prediction in both scenarios of conditional distributions of the flow count given the flow volume under different settings of training sample size and testing sample size.

[1]http://www.maths.adelaide.edu.au/matthew.roughan/Stuff/Abilene.tar.gz

TABLE I: Prediction Mean Square Error for M_1

Algo	(200, 700)	(300, 600)	(400, 500)
KBR	0.2215	0.1684	0.1991
RNN	0.2396	0.2151	0.2538

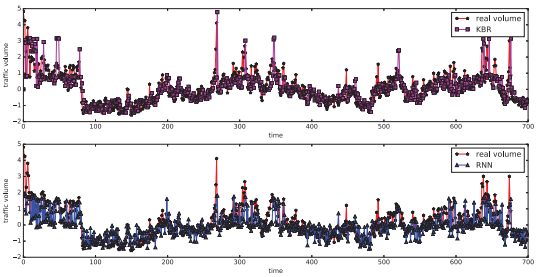

Fig. 6: Traffic Volume Prediction based on Flow Counts under M1

We find that the prediction error is quite small as here we normalize the time series such that the original time series is with unit variance.

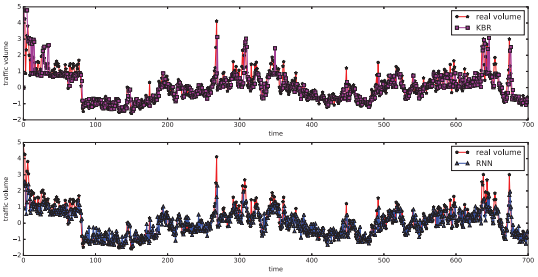

Fig. 7: Traffic Volume Prediction based on Flow Counts under M2

Fig. 6 and 7 show the predicted traffic volume vs. the real traffic volume for both conditionals M_1 with the training sample equals to 200. We can see the predicted traffic volumes are very close to the real value.

B. Real Network Traffic

In this section, we conduct experiments using the real network traffic from the Internet. We divide the whole trace into 400 time interval according to the timestamps of the flow records and count the number of flows as well as the total number of bytes within each time interval.

The relationship between the flow counts and the total traffic volume is shown in Fig. 2. We also conduct experiments

TABLE II: Prediction Mean Square Error for M_2

Algo	(200, 700)	(300, 600)	(400, 500)
KBR	0.1624	0.1454	0.1599
RNN	0.1686	0.1545	0.1608

TABLE III: Prediction Mean Square Error

Algo	(100, 300)	(200, 200)	(300, 100)
KBR	0.3545	0.3092	0.2926
RNN	0.4285	0.4126	0.3441
RNN (volume observed)	0.6524	0.2202	0.1517

using different training sample size and testing sample size (S_{tr}, S_{tst}) = (100, 300), (200, 200) and (300, 100).

In this experiment, we also compare the accuracy of prediction based on flow counts with the accuracy of prediction based on observed traffic volumes in previous timestamps. Note that in this paper, we advocate the idea of prediction based on simple statistics such as flow counts since direct measurement of traffic volume is too expensive. In this experiment, we include time series prediction based on past observations of traffic volume just in order to show the gap between prediction based on noisy observations and prediction based on perfect information. We use RNN again but the input to RNN is the past observed traffic volume other than the past flow count.

The prediction accuracies are shown in Table III.

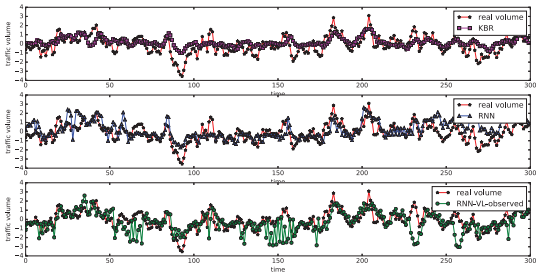

Fig. 8: Traffic Volume Prediction based on Flow Counts(Training:Testing = 1:3)

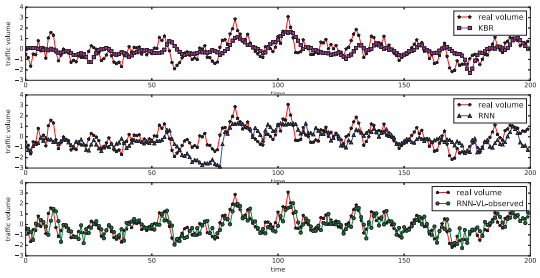

Fig. 9: Traffic Volume Prediction based on Flow Counts(Training:Testing = 2:2)

We can see that generally, when the sample size is larger, the prediction error gets smaller. Even when the ratio between the testing sample size and the training sample size is as large as 3, the prediction errors for both KBR and RNN are less than 0.5 which are less than half of the variance of the time series. We can see the gap between prediction based on noisy observations and prediction based on perfect information is not

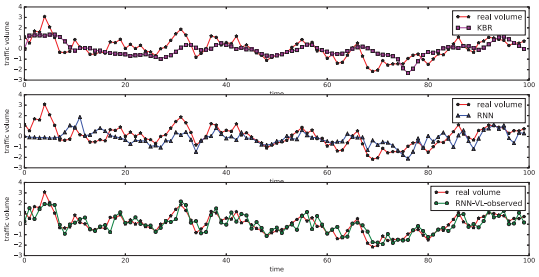

Fig. 10: Traffic Volume Prediction based on Flow Counts(Training:Testing = 3:1)

large which shows there should be a solution that balances the prediction accuracy and monitoring costs.

Fig. 8, 9 and 10 also show the predicted time series vs the real one under different settings of training and testing sample size ratios. We can see that is generally, the predicted time series match the real one quite well.

V. Conclusion

In this work, we described how to use several machine learning techniques including Hidden Markov Model based on Kernel Bayes Rule as well as Recurrent Neural Network to estimate traffic volume as well as to predict the future traffic volume based on some simple flow level statistics that can be much easier collected using sketching techniques. This approach avoids direct measurement of the traffic volume and thus is much less costly in terms of the complexity and the storage requirement. This is particularly useful in large scale high speed network where direct measurement of traffic volume for all source destination pairs is nearly impossible and the estimation of the network traffic volume from the link load is extremely difficult. By conducting semi-simulation and experiments using real network traffic data, we show the use of simple flow level statistics such as the flow counts, provides useful information to predict traffic volume. In the future work, we plan to apply the proposed framework to real network monitoring and traffic engineering.

There are also remaining issues to be addressed. One of them is whether the dependence between the traffic volume and simple flow level statistics such as flow counts is significant enough in all networks such as WAN and inter-data center traffic. The second issue is the nonstationarity of the network traffic. Since the network traffic is changing dynamically which means that the behaviors of the transition function and the emission function could also change. In this case, we need to develop online learning algorithms for KBR as well as RNN such that the model adjusts and adapts itself to the dynamic network traffic. The third open question is besides the flow count, what other flow level statistics we can also include in order to improve the prediction accuracy.

References

[1] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: router link data. *Journal of the American statistical association*, 95(452):1063–1075, 2000.

[2] S. Chabaa, A. Zeroual, J. Antari, et al. Identification and prediction of internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems and Applications*, 2(03):147, 2010.

[3] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. *IEEE Transactions on Signal Processing*, 58(12):6029–6039, 2010.

[4] M. H. Firooz and S. Roy. Network tomography via compressed sensing. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[5] K. Fukumizu, L. Song, and A. Gretton. Kernel bayes' rule. In *Advances in neural information processing systems*, pages 1737–1745, 2011.

[6] N. K. Hoong, P. K. Hoong, I. K. Tan, N. Muthuvelu, and L. C. Seng. Impact of utilizing forecasted network traffic for data transfers. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 1199–1204. IEEE, 2011.

[7] P. K. Hoong, I. K. Tan, and C. Y. Keong. Bittorrent network traffic forecasting with arma. *arXiv preprint arXiv:1208.1896*, 2012.

[8] W. Junsong, W. Jiukun, Z. Maohua, and W. Junjie. Prediction of internet traffic based on elman neural network. In *2009 Chinese Control and Decision Conference*, pages 1248–1252. IEEE, 2009.

[9] G. Liang and B. Yu. Maximum pseudo likelihood estimation in network tomography. *IEEE Transactions on Signal Processing*, 51(8):2043–2053, 2003.

[10] D.-C. Park and D.-M. Woo. Prediction of network traffic using dynamic bilinear recurrent neural network. In *2009 Fifth International Conference on Natural Computation*, volume 2, pages 419–423. IEEE, 2009.

[11] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[12] L. Song, K. Fukumizu, and A. Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

[13] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968. ACM, 2009.

[14] A. R. Syed et al. Forecasting network traffic load usingwavelet filters and seasonal autoregressivemoving average model. *International Journal of Computer and Electrical Engineering*, 2(6):979, 2010.

[15] Y. Xu. Kernel bayes rule. *Journal of Machine Learning Research*, 14, 2013.

[16] S. Yantai, Y. Minfang, Y. Oliver, L. Jiakun, and F. Huifang. Wireless traffic modeling and prediction using seasonal arima models. *IEICE transactions on communications*, 88(10):3992–3999, 2005.

[17] E. Yu and C. R. Chen. Traffic prediction using neural networks. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM'93., IEEE*, pages 991–995. IEEE, 1993.