Journal of Network and Computer Applications 60 (2016) 220–232

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Traffic matrix estimation: A neural network approach with extended input and expectation maximization iteration ☆

Haifeng Zhou [a], Liansheng Tan [b,*], Qian Zeng [c], Chunming Wu [a]

[a] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, PR China
[b] Computer Science Department, Central China Normal University, Wuhan 430079, PR China
[c] School of Information Management, Wuhan University, Wuhan 430072, PR China

ARTICLE INFO

Article history:
Received 15 May 2013
Received in revised form
24 August 2015
Accepted 6 November 2015
Available online 17 December 2015

Keywords:
Traffic matrix
Neural network
EM algorithm
Moore–Penrose inverse
Singular value decomposition (SVD)
IP network

ABSTRACT

Accurately estimating of IP Traffic matrix (TM) is still a challenging task and it has wide applications in network management, load-balancing, traffic detecting and so on. In this paper, we propose an accurate method, i.e., the Moore–Penrose inverse based neural network approach for the estimation of IP network traffic matrix with extended input and expectation maximization iteration, which is termed as MNETME for short. Firstly, MNETME adopts the extended input component, i.e., the product of routing matrix's Moore–Penrose inverse and the link load vector, as the input to the neural network. Secondly, the EM algorithm is incorporated into its architecture to deal with the output data of the neural network. Therefore, MNETME manifests itself with the advantages that it needs less input data, but has better accuracy of estimation. We theoretically analyze the algorithm and then study its performance using the real data from the Abilene Network. The simulation results show that MNETME leads to a more accurate estimation in contrast to the previous methods, meanwhile it holds better robustness and can well track the traffic fluctuations. We finally extend MNETME to random routing networks by proposing a new model of random routing which overcomes three fatal deficiencies of the existing model and it is easier, more practical and more precise.

1. Introduction

Traffic matrix (TM) reveals flow's volume in a IP network globally with every entry giving the volume of one origin–destination (OD) pair's flow. TM plays an important role in network engineering and administration. For example, network designer and manager can take advantage of TM to optimize the network's performance by load-balancing, to detect the abnormal traffic flows and to restrict some traffic flows for security reason and so on. Unfortunately, direct accurate measurement is time-consuming and costly, since the limited and inconsistent router support for flow level measurement capability in current networks (Zhang et al., 2003). What's worse, the accurate estimation of TM is difficult due to the estimation problem is under-constrained. As a result, given the fact that enormous research efforts have been directed to this subject in recent decades, the accurate estimation to TM remains a challenge.

Vardi (1996a) first applied the statistical approach to model TM estimation problem, which can be categorized into tomography method. It assumes that there are n nodes and M directed links in a network, with the number of OD pairs $N=n\times(n-1)$. The fundamental model of TM estimation is depicted as follows:

$$Y=AX, \tag{1}$$

where X denotes the traffic matrix written as a column vector with N elements, Y denotes the link load column vector with M elements, and A denotes the $M\times N$ routing matrix of a network with fixed routing. A is a zero-one matrix: its rows represent the directed links, its columns correspond to the OD pairs, and whether its entry is one or zero depends on whether the directed link belongs to the path of the OD pair (see Section 2.1 for detail). Because the number of OD pairs is bigger than the directed link's (i.e., $M<N$), Eq. (1) is under-constrained and its solution is not unique. To determine a solution from the solution space, Vardi's work assumes that each OD pair's flow follows Poisson distribution. Afterwards, the expectation maximization (EM) iteration (see

☆This research is supported by the National Natural Science Foundation of China (Nos. 61070197, 61370107, 61379118), the National Basic Research Program of China (No. 2012CB315903).

* Corresponding author. Tel.: +86 27 6786 8318.
E-mail addresses: zhouhaifeng@zju.edu.cn (H. Zhou), l.tan@mail.ccnu.edu.cn (L. Tan), zengqianhn@whu.edu.cn (Q. Zeng), wuchunming@zju.edu.cn (C. Wu).
URL: http://cs.ccnu.edu.cn/teacher/tls/index.html (L. Tan).

http://dx.doi.org/10.1016/j.jnca.2015.11.013
1084-8045/

Section 2.4 for detail) is used to attain the estimation of TM. Cao et al. (2000) made subsequent contributions, where the EM algorithm is modified to use in the Gaussian model. The essence of tomography method is using models of the higher order statistics of OD flow to construct additional constraints for ascertaining a solution. Thus making assumptions of the distribution of OD flow is inevitable. However, Medina et al. (2002) shown that these basic assumptions of OD flow which underlie the statistical model are not validated. What's worse, the above tomography methods would perform badly when these assumptions are not founded.

To overcome the defect of tomography method, based on gravity model and tomography method, Zhang et al. (2003) proposed the Tomogravity method which is suitable for practical network. It uses the gravity model to construct the relationships between OD flows and thus obtain the gravity model solution of TM. Then, determine a best solution which has the least Euclidean distance to the gravity model solution from the solution space of (1). Finally, exploit the iterative proportional fitting procedure (IPFP) (Cao et al., 2000) to deal with the data to keep non-negativity. Tomogravity attains a better accuracy than tomography method and needs less time to make computation. However, Tomogravity also makes its assumption that the OD flows satisfy the gravity model, and it also has big errors, especially to those OD flows whose volumes are small. What's worse, Juva (2007) found that if the OD flows do not satisfy the above assumption, the accuracy of Tomogravity declines faster than tomography method. Compared Tomogravity with tomography method, we find that both of these two methods make assumption of OD flows, yet the gravity model's assumption is more consistent with the flows in the real network. As a result, Tomogravity is superior than tomography method and is a relatively accurate method so far, which has been used in practical network engineering (Zhang et al., 2003).

Jiang et al. (2011) proposed BPTME method based on back-propagation neural network (BPNN) (see Section 2.3). In view of Eq. (1), it adopts the link load vector Y as the input of BPNN and treats the real traffic matrix X as the output to train the BPNN. After the training phase, Jiang's work uses the trained BPNN to predict the future TM. Finally, exploit IPFP to deal with the output data of BPNN to satisfy (1). In contrast to previous methods, BPTME avoids complex mathematical computation, and covers both spatial and temporal correlations of TM rather than the only spatial correlation attained by Tomogravity and tomography method. However, the estimation accuracy of BPTMP is still not satisfying.

Besides, Tan and Wang (2007a,b) study the TM estimation problem from a view of optimization, Tchrakian et al. (2012), Caggiani et al. (2012) and Djukic et al. (2012) shed light on real-time estimation of traffic matrix. Fang et al. (2007) provides a method that is valid even when the measured data are incomplete. Jiang et al. (2011), Zhang et al. (2009), Nie et al. (2013), Conti et al. (2010), Conti et al. (2012), Juva et al. (2006) and Juva () contribute to the recent development of this problem.

In this paper, we propose a new method (called MNETME) for traffic matrix estimation which is more accurate compared with previous representative methods, and even extend it to the random routing network by proposing a new advanced random routing model.

Our method is based on neural network and EM algorithm. Through mathematical analysis, we adopt the extended input, i.e., the product of routing matrix's Moore-Penrose inverse and link load vector, as the input of neural network and the corresponding TM as the output. Because the routing information is transmitted to the neural network by the Moore–Penrose inverse in the input component, the neural network need not to get such information from training input–output data. As a result, the neural network has more space to cover other important information. Consequently, compared with the simple way only using the link load vector Y as the input part of neural network, which is adopted by the previous neural network estimation methods, like BPTME, obviously, MNETME is more accurate, more capable to track the fluctuations at the same number of data, which is verified by simulations. What's more, the BPNN's output data post-treated by EM algorithm further makes the final estimation value more accurate.

We make two simulations using the real data from the Abilene Network to justify MNETME by comparing with two previous representative methods, i.e., Tomogravity and BPTME. The first simulation result shows that MNETME makes a more accurate estimation in contrast to the previous two methods, especially to BPTME, meanwhile holds better robustness and well track the fluctuations. The second simulation verifies the superiority of the extended input by comparing with the simple one.

Afterwards, based on the simple introduction of the existing random routing model (see Section 2.1.2) by Vardi (1996a), we show its three fatal deficiencies: first, the data needed is impossible to search in the current network architecture; second, this model cannot cover all possible paths of each OD flow; third, its assumption that there is no ring in the path of OD flow dose not satisfy the situation of real network. To overcome all these deficiencies, we propose a new random routing model. It can well avoid the above three deficiencies and is easier, more practical and more precise. At last, we extend MNETME to random routing networks.

This paper is organized as follows. In Section 2, we introduce relevant background and basic concepts. In Section 3, we respectively introduce the principle of MNETME, its four phases and the corresponding procedures. In Section 4, we introduce our two simulations and analyze the results. In Section 5, we propose the random routing model and extend MNETME to the random routing networks. In Section 6, we conclude the paper.

2. The TM estimation problem: fixed routing, random routing and related background

2.1. Routing matrix

The network routing is divided into two categories: fixed routing and random routing. We respectively introduce them in the following two subsections.

2.1.1. Routing matrix in fixed routing networks

Let n denotes the number of router nodes and M denotes the number of directed links in a network. If the network topology is strongly directly connected and every OD pair's flow between each two router nodes is permitted, the number of OD pairs is $N = n \times (n-1)$, or that $N < n \times (n-1)$. Let A denotes the fixed routing matrix with M rows and N columns that respectively corresponding to directed links and OD pairs. It is a zero-one matrix and whether its entry is one or zero depends on whether the directed link belongs to the path of the OD pair. For instance, if the entry a_{ij} equals to one, it means the ith directed link belongs to the path of the jth OD pair.

Example 1. Consider the network topology of Fig. 1. There are $4 \times 3 = 12$ OD pairs and 9 directed links. The traffic matrix A is in Table 1 (the blank entries are equal to zero). The routing of each OD pair is pre-specified in the fixed routing network. For example, although in Fig. 1 both of paths $c \rightarrow b \rightarrow a$ and $c \rightarrow d \rightarrow b \rightarrow a$, connect