*Advanced Science Letters*
Vol. 23, 3934–3938, 2017

# Internet Traffic Matrix Estimation Based on Compressive Sampling

Indrarini Dyah Irawati[1,2,*], Andriyan Bayu Suksmono[1], and Ian Joseph Matheus Edward[1]

[1] *School of Electrical and Informatics, Institut Teknologi Bandung, Indonesia*
[2] *Telkom Applied Science School, Telkom University, Bandung, Indonesia*

A method for estimating traffic matrix is needed for network operator. Traffic matrix is an essential parameter for network maintenance, network planning, and network monitoring. However, it is very difficult to measure it directly. Hence, traffic matrix is inferred from direct link measurement by estimation. There are various techniques for traffic matrix estimation. This paper proposed SVD-$l_1$ (Singular Value Decomposition-$l_1$) for traffic matrix estimation based on compressive sampling. The proposed method is compared with IRLS (Iteratively Reweighted Least Square), SRSVD (Sparsity Regularized Singular Value Decomposition), OMP (Orthogonal Matching Pursuit), and interpolation techniques. The result showed that the NMAE (Normalized Mean Absolute Error) performance parameter of SVD-$l_1$ as good as SRSVD and better than the other algorithms. The computational time of SVD-$l_1$ is the longest compared to others.

**Keywords:** Internet Traffic Matrix, Estimation, Low-Rank, Compressive Sampling, Interpolation.



## 1. INTRODUCTION

Traffic Matrix (TM) is a crucial information of network operator for network management, planning, and traffic engineering. It represents the volume of traffic flowing between a source to a destination in a network.[1] It is not easy for network operator to measure traffic matrix directly, especially for a large-scale network. The problem can be solved by estimating traffic matrix based on SNMP (Simple Network Management Protocol) link-level measurement from network devices.

In a real-network, a significant missing value of traffic matrix measurement can happen. Currently, there are number of research for estimating a traffic matrix. Usually, the problem is solved as a liner-inverse problem using Poisson distribution.[2,3] Further research on the TM estimation is developed using normal distribution.[4] Following studies consider the properties of real-traffic network and the correlation between nodes in a network. This research has resulted in the development of several gravity models such as simple gravity model,[5] generalized gravity model,[6] tomo-gravity model.[6,7] Another estimation method is the PCA (Principle Component Analysis) that overcome the problem of huge traffic matrix dimension.[1,8] The Compressive Sampling (CS) model was also introduced to approximate traffic matrix by solving an optimization problem as a low-rank matrix.[9–11]

CS is a new paradigm for acquisition and reconstruction in signal processing.[12–14] CS works on signal that has sparse representation or low-rank nature and incoherence property.[15,16] Interpolation is a technique for completion to the missing value. In the term of matrix, interpolation is referred to as matrix completion.[15]

*Contributions*: Both the CS and interpolation methods can solve the problem of traffic matrix estimation. In this paper, we use Singular Value Decomposition (SVD) to obtain a low rank matrix and represent it as a temporal structure. The researches modelled traffic matrix as purely spatial,[1,6,10] purely temporal,[2,10] and combining spatio-temporal.[9,11] We use a routing matrix whose elements are '0' or '1' as the measurement matrix for estimating TM. A real dataset from Abilene (Internet2) is used for this simulation.[17] We compare traffic matrix estimations using linear interpolation, CS-SVD$-l_1$, CS-IRLS,[18] CS-SRSVD,[9,11] and CS-OMP.[19]

## 2. PRELIMINARIES

### 2.1. Traffic Matrix

A traffic matrix is a non-negative matrix $X = (x_{ij})$ that expresses the volume of traffics flowing between source $i$ to destination $j$. For a network with $n$-nodes, the traffic matrix consists of $N = n \times n$ source-destination pairs at a moment. If TM works on $T$ time intervals, it can be considered as a 3-dimensional array $X \in R^n \times R^n \times R^T$. TM can be expressed as column vector, $X_t$ (where $t$ indicates time point). As the time changes, the TM changes from 3-dimensional array into 2-dimensional array.

As an example, the Abilene network topology consisting of 12 nodes is shown in Figure 1.[17] The TM is a 2-dimensional array $X \in R^n \times R^n$ where $X$ represents at one time as shown in (1). $X$ can be converted into a column vector $(N \times 1)$ where

*Author to whom correspondence should be addressed.


1936-6612/2017/23/3934/005 doi:10.1166/asl.2017.8283

Adv. Sci. Lett. 23, 3934–3938, 2017



**Fig. 1.** Abilene topology.

$N = n^2$ and $x_i$ $(i = 1, 2, \ldots, N)$ as traffic volume for all pairs from source to destination. For the TM that is measured during the time $T$, the dimension of TM is $(N \times T)$ as shown in (2). In this equation $X(n, t)$ denotes the traffic measurement of $n$-th source-destination at time $t$, where $n = 1, 2, \ldots 12 \times 12$, $t = 1, 2, \ldots, T$.

$$X = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{11} \\ s_{12} \end{matrix} \begin{bmatrix} x_1 & x_{13} & \cdots & x_{133} \\ x_2 & x_{14} & \cdots & x_{134} \\ x_3 & x_{15} & \cdots & x_{135} \\ \vdots & \vdots & \cdots & \vdots \\ x_{11} & x_{23} & \cdots & x_{143} \\ x_{12} & x_{24} & \cdots & x_{144} \end{bmatrix} \tag{1}$$



$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & & x_{1,T} \\ x_{2,1} & x_{2,2} & & x_{2,T} \\ x_{3,1} & x_{3,2} & & x_{3,T} \\ \vdots & \vdots & & \vdots \\ x_{13,1} & x_{13,2} & \cdots & x_{13,T} \\ \vdots & \vdots & & \vdots \\ x_{25,1} & x_{25,2} & & x_{25,T} \\ \vdots & \vdots & & \vdots \\ x_{144,1} & x_{144,2} & & x_{144,T} \end{bmatrix} \tag{2}$$

This paper aims to estimate the actual-TM based on measurement results at link-level. The relationship between the link-loads $Y$ and TM $(X)$ can be expressed by the following linear matrix equation:

$$Y = AX \tag{3}$$

where $A$ is a routing matrix, which describes links that are used by the nodes.[1] The estimation of the real-TM is solved by finding an optimal solution $\hat{X}$ to (3) from link-load measurement $Y$ that approximates the original $X$ as closely as possible.

### 2.2. Compressive Sampling (CS)

In the compressive sampling, there are 2 requirements: the TM is low-rank[15] and the routing matrix is incoherent.[16] Because of the TM is not a low-rank matrix, we use SVD as a tool to obtain a low-rank approximation. SVD decomposes $N \times T$ according to (4).[8,21]

$$X = U\Sigma V^T \tag{4}$$

where $U$ is a $N \times N$ unitary or orthogonal matrix (i.e., $UU^T = U^T U = I$), and $V$ is a $T \times T$ unitary or orthogonal matrix (i.e., $VV^T = V^T V = I$), with $V^T$ is the transpose of $V$, and $\Sigma$ is an $N \times T$ diagonal matrix containing non-zero elements as the singular values $\sigma_R$ of $X$. The singular values are arranged so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. The number of non-zero singular values represent the rank of matrix that indicates the number of linearly independent rows or columns. If $\ll \min(N, T)$, the TM is low-rank. The TM is assumed to have a multiplicative low-rank structure, it can be rewritten as:

$$X = U\Sigma V^T = \sum_{r=1}^{\min(N,T)} \sigma_r u_r v_r^T \tag{5}$$

where $u_r$ is the $ith$ columns of $U$ and $v_r$ is the $ith$ columns of $V$. The approximation of $X$ from the SVD by keeping only the largest singular values in the summation and dropping the others is equivalent to

$$\hat{X} = \sum_{r=1}^{R} \sigma_r u_r v_r^T = \sum_{r=1}^{R} \sigma_r B_r \tag{6}$$

where $\hat{X}$ is the best rank-approximation of $X$ and $B_r$ is matrix that constructed by rank-1.

A routing matrix $(A)$ is generated by random Boolean.[10,20] It is constructed to obey the GUP (Generalized Uncertainty Property).[16] According to the CS theory, the minimum number of sample $m$ is given by the following expression (7).[10,22]

$$M \geq C\left(R \log \frac{N}{R}\right) \tag{7}$$

where, $C$ is a positive constant from 1 to 2, and whereas $R$ is the number of non-zero singular values.

## 3. ESTIMATION METHOD OF REAL-TM

### 3.1. Interpolation

Interpolation is a method that uses the nearest neighbor values to construct an unknown value. The TM $(X)$ interpolation is performed for each column with length of $X$ rows. The interpolation formula is shown in the following Eq. (8).[23]

$$X_{\text{row}}(i) = \text{mean } (X_{\text{row}}(i+) + X_{\text{row}}(i-)) \tag{8}$$

where $X_{\text{row}}(i)$ is estimation row matrix of $X$, $X_{\text{row}}(i+)$ is a row matrix neighbors above $X_{\text{row}}(i)$, and $X_{\text{row}}(i-))$ is a row matrix neighbor below $X_{\text{row}}(i)$.

### 3.2. CS-SVD-$l_1$

The TM estimation $\hat{X}$ must have an optimal solution that approximates the original $X$ as closely as possible with respect to $l_1$ norm, following Eq. (9).[24]

$$\min \|X - \hat{X}\|_1, \quad \text{subject to } AX = Y \tag{9}$$

where $\|\cdot\|_1$ is the $l_1$ norm used to measure the error between TM $X$ and $\hat{X}$.

### 3.3. CS-Iteratively Reweighted Least Square (IRLS)

IRLS is used to solve optimization problem of Eq. (3) with $l_p$ norm estimation, where $0 < p < 1$.[25] That is

$$\min \|X - \hat{X}\|_p^p, \quad \text{subject to } AX = Y \tag{10}$$

Adv. Sci. Lett. 23, 3934–3938, 2017

IRLS uses weights to estimate TM that is defined as follows:

$$W_p = [(X'X)^2 + \gamma I]^{(p/2)-1} \tag{11}$$

where $\gamma$ is a regularization parameter added to ensure that $W_p$ is well defined, and $\gamma > 10^{-9}$. The estimation TM can be obtained from the equation:

$$\hat{X} = QX^T(XQX^T)^{-1}Y \tag{12}$$

where $Q$ as a diagonal matrix with the following value:

$$Q = 1/W_p \tag{13}$$

### 3.4. CS-OMP

OMP estimates TM ($X$) which determines columns of $A$ that participates in the measurement of $Y$. The idea behind the algorithm is to choose columns in a greedy fashion. At each iteration, a column of $A$ which is highly correlated with the residual part of $Y$ is chosen. Then reducing any contribution to the $Y$ yields residue. After $k$ iteration, algorithm will identify the right set from the columns. The procedure of the algorithm are as follows:[19]

(1) Initialize the residual $R_0 = Y$, index set $\Delta_0$, and the iteration counter $t = 1$. And $A_o$ is an empty matrix.

(2) Find the index $\lambda_t$ that solves an easy optimization problem:

$$\lambda_t = \arg\max_{j=1\ldots M} |\langle R_{t-1}, A_j\rangle| \tag{14}$$

If the maximum occurs of the inner product, break the tie deterministically.

(3) Augment the index set:

$$\Delta_t = \Delta_{t-1} \cup \{\lambda_t\} \tag{15}$$

The matrix of chosen atoms:

$$A_t = [A_{t-1} a_{\lambda_t}] \tag{16}$$

(4) Solve a least squares problem to obtain the estimation:

$$x_t = \arg\min_x \|Y - A_t X\|_2 \tag{17}$$

(5) Calculate the new approximation of the data and the residual:

$$a_t = A_t - X_t \text{ and } R_t = Y - a_t \tag{18}$$

Increments $t$ and returns to Eq. (3) until $t < k$. The estimate $\hat{X}$ for the ideal matrix has non zero indicates at the component listed $\Delta_k$. The value of the estimate $\hat{X}$ in component $\lambda_j$ equal the $jth$ component of $X_t$.

### 3.5. CS-SRSVD

SVD can created as a factorization of a matrix $X$, that can be expressed as:

$$X = U\Sigma V^T = LR^T \tag{19}$$

where $L = U\Sigma^{1/2}$ and $R = V\Sigma^{1/2}$. The low rank problem can be formulated as the following rank minimization:

$$\min rank(X), \quad \text{subject to } A(X) = B \tag{20}$$

where $rank(\cdot)$ is the rank of a matrix, $A(\cdot)$ is a linear operator that works on matrix $X$, $B$ denotes the set of direct measurements. Because $L$ and $R$ have low Frobenius norm, then (20) is equivalent to:

$$\min rank\|L\|_F^2 + \|R\|_F^2, \quad \text{st } A(LR^T) = B \tag{21}$$

Furthermore, because of the real TM ($X$) is not exactly low rank, the regularized parameter ($\lambda$) is added to solve the optimization problem:

$$\min \|A(LR^T) - B\| + \lambda(\|L\|_F^2 + \|R\|_F^2) \tag{22}$$

where $\lambda$ is parameter that control tradeoff between a precise fit to the measured data and the purpose of low rank target. It names SRSVD interpolation.[9]

## 4. EXPERIMENTAL RESULT

### 4.1. Data

We use real TM data from Abilene (Internet2)[17] that used previously in various studies.[9–11] The Abilene network consists of $n = 12$ nodes and $r = 54$ links. There are 144 source-destination pairs ($N = 144$). The real TM is collected every 5 minutes. We examine for a day, so the Abilene TM data is denoted by $X$ is a $144 \times 288$ matrix. Each column represents a 5-minute traffic matrix snapshot and each Source-Destination row represents a time-series of traffic intensities for each Source-Destination pair.

Figure 2 shows low-rank structure from Abilene TM that resulted from SVD process. This figure says represents that the top several singular values occupy most of the energy.



### 4.2. Metric

The metric used to measure the error accuracy is Normalized Mean Absolute Error (NMAE). That is,[9,11]

$$\text{NMAE}(X, \hat{X}) = \frac{\sum_{i,j:A(i,j)=0}^{N} |X(i,j) - \hat{X}(i,j)|}{\sum_{i,j:A(i,j)=0}^{N} |X(i,j)|} \tag{23}$$

where $\hat{X}$ is the estimated error, and $A(i, j) = 0$ means the missing value in $X(i, j)$. The NMAE only calculate errors on the missing values.

### 4.3. Performance Comparison

We compare performance of estimation algorithm such us SVD$-l_1$, SRSVD, IRLS, OMP, and Interpolation. These

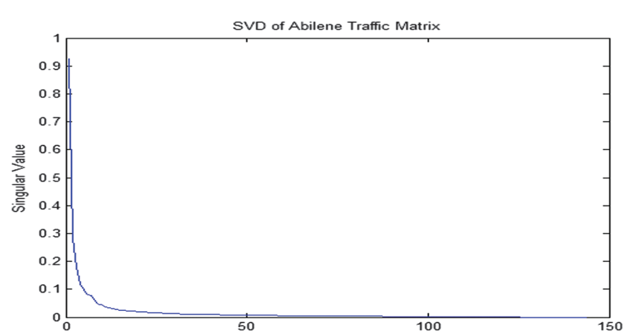


**Fig. 2.** Low-rank structure of abilene TM.

(a) Loss probability (ρ = 0.1) (b) Loss probability (ρ = 0.7)

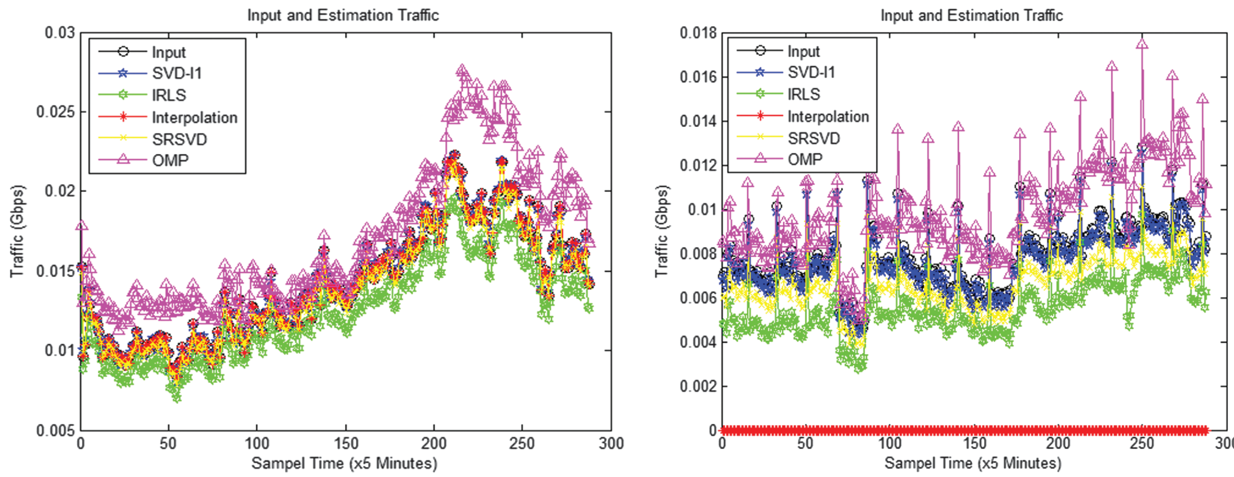


**Fig. 3.** Input and estimation traffic, (a) Loss probability ($p = 0.1$), (b) Loss probability ($p = 0.7$).

algorithms use the same parameters such as: $R = 30$ and $C = 2$. Following a data loss pattern, a series of row random missing value from low loss probability (0.01) to high loss probability (0.9) is implemented.

To evaluate the accuracy of the algorithms, we select a maximum traffic from the network to be tested. There are two scenarios:

(1) loss probability = 0.1 and
(2) loss probability = 0.7.

Figure 3 show the input (original) traffic by the black lines and the estimation results of TM estimation algorithms. Figure 3(a) shows the TM estimation results for loss probability 0.1, while Figure 3(b) shows the loss probability 0.7. We can see from Figure 3(a), that the SVD-$l_1$, SRSVD, Interpolation can estimate the TM precisely for low loss probability, but the OMP and IRLS can not perform well. In Figure 3(b), we point out that the proposed method SVD-$l_1$ is the best estimator than the others for loss probability ($p = 0.7$).

Figure 4 showed the absolute error that is difference between the original traffic and estimation results. Figure 4(a) illustrates the absolute error for loss probability 0.1, while Figure 4(b) illustrates the loss probability 0.7. We can see from Figure 4, that the OMP algorithm produces estimation with large absolute errors both in low and high loss probability, followed by IRLS algorithm. Interpolation method generates small absolute errors in low loss probability, but large absolute errors in high loss probability. We observe that the SVD-$l_1$ and SRSVD have small absolute error for low and high loss probability.



Figure 5 showed NMAE parameter for row random loss TM with loss probability from 0.01 to 0.9. The NMAE of all estimation algorithms have an ascending trend along with increment of the loss probability. Interpolation achieves best performance for low loss probability less than 0.1. In essence, interpolation is not suitable for resolving the problems of row random loss. This is because the row of TM represents a link in the network, and between rows are not correlated with each other. This is indicated on the increasing of NMAE for loss probability greater than 0.1. SVD-$l_1$ and SRSVD have a similar performance and outperform all algorithms for whole loss probability.

Figure 6 showed the computational time for all estimation algorithms. In general, the computational time increases with the increasing of loss probability. The SVD-$l_1$ has the longest

(a) Loss probability (ρ = 0.1) (b) Loss probability (ρ = 0.7)

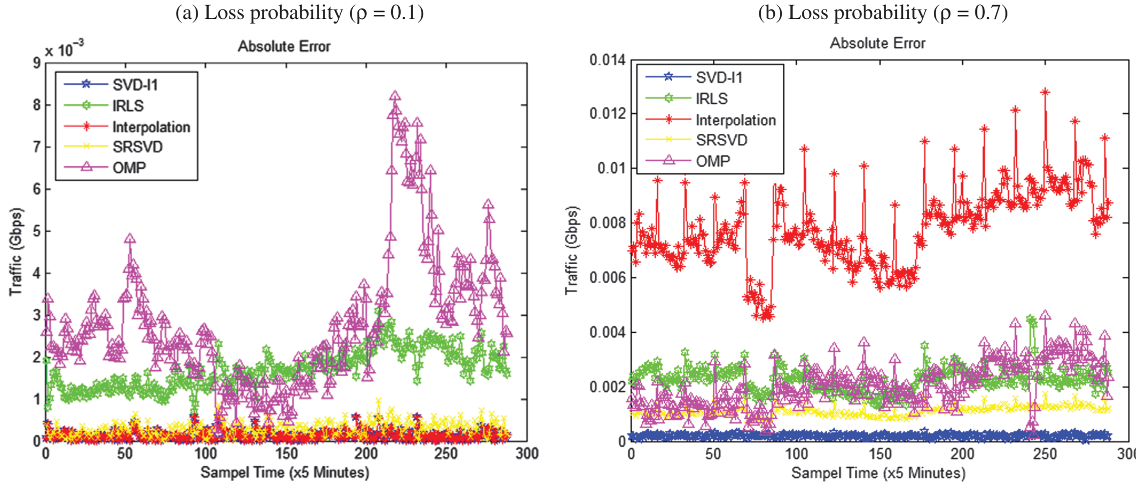


**Fig. 4.** Absolute error, (a) loss probability ($p = 0.1$), loss probability ($p = 0.7$).

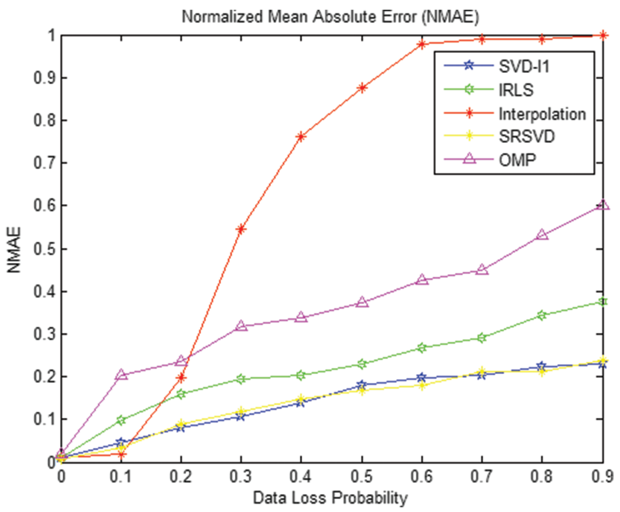


**Fig. 5.** NMAE comparison between estimation algorithms for different loss probability.

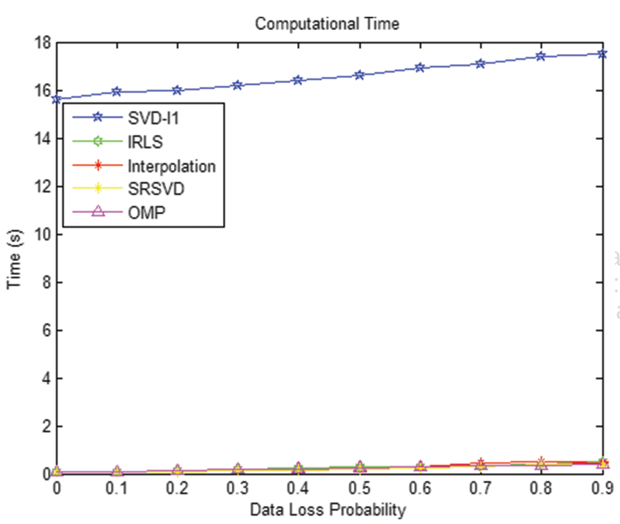


**Fig. 6.** Computational time comparison between estimation algorithms for different loss probability.

computing time, which is about 15.7–17.8 s, while the other algorithms have less than 1 s. Based on this results, the OMP algorithm has the fastest computational time.

## 5. CONCLUSIONS

Internet traffic matrix estimation based on compressive sampling such us SVD$-l_1$, IRLS, SRSVD, OMP produce a better performance than the conventional method (Interpolation). The SVD$-l_1$ and SRSVD exceed the other algorithms for all data loss probability. While the Interpolation has good performance only for low loss probability ($<0.1$). The OMP computing time is fastest, while SVD is slowest.

## References and Notes

1. A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, Traffic matrices: Balancing measurements, inference and modeling, *Proceedings of the 2005 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, Banff, AB **(2005)**, pp. 192–199.
2. Y. Vardi, *Journal of the American Statistical Association* 91, 365 **(1996)**.
3. C. Tebaldi and M. West, *Journal of the American Statistical Association* 93, 557 **(1998)**.
4. J. Cao, D. Davis, S. V. Wiel, and B. Yu, *Journal of the American Statistical Association* 95, 1063 **(2000)**.
5. M. Roughan, First order characterization of internet traffic matrices, *Proceedings of the 55th session of the International Statistics Institute*, Sydney, Australia **(2005)**.
6. Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, *ACM Sigmetrics Performance Evaluation Review* 31, 206 **(2003)**.
7. Y. Zhang, M. Roughan, C. Lund, and D. Donoho, An information-theoretic approach to traffic matrix estimation, *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe, Germany **(2003)**.
8. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, Structural analysis of network traffic flows, *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, New York **(2004)**.
9. M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, *IEEE/ACM Transaction on Networking* 20, 662 **(2012)**.
10. L. Nie, D. Jiang, and L. Guo, *Journal Computers and Electrical Engineering* 39, 1422 **(2013)**.
11. Z. Huibin, Z. Dafang, X. Kun, and W. Xiaoyang, *IEEE Journal and Magazine, China Communication* 11, 1 **(2014)**.
12. D. Donoho, *IEEE Transaction Information Theory* 52, 1289 **(2006)**.
13. R. Baraniuk, E. Candes, R. Nowak, and M. Vetterli, *IEEE Signal Processing Magazine* 118 **(2008)**.
14. E. Candes and T. Tao, *IEEE Transaction Information Theory* 52, 5406 **(2006)**.
15. E. Candes and B. Recht, *Found. Comput. Math* 9, 717 **(2009)**.
16. M. Elad and A. Bruckstein, *IEEE Transaction on Information Theory* 48, 2558 **(2002)**.
17. Y. Zhang, Abilene Dataset. [Online]. Available: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/.
18. R. Chartrand and W. Yin, Iteratively reweighted algorithms for compressive sampling, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **(2008)**, pp. 3869–3872.
19. J. Troppand and A. Gilbert, *IEEE Transaction Information Theory* 53, 4655 **(2007)**.
20. R. E. Bryant, *IEEE Trans. on Computers* 35, 677 **(1986)**.
21. D. Yang, Singular Value Decomposition for High Dimensional Data, University of Pennsylvania **(2012)**.
22. W. Xu, E. Mallada, and A. Tang, Compressive sensing over graph, *IEEE Proceedings of Infocom.* **(2011)**, pp. 2087–2095.
23. E. Kreysig, Advanced Engineering Mathematics, John Wiley and Sons, Inc. **(1972)**.
24. E. Candes and J. Romberg, Caltech. $l_1$ magic: Recovery of Sparse Signals via Convex Programming **(2005)**.
25. K. Mohan and M. Fazel, *Journal of Machine Learning Research* 3441 **(2012)**.

Received: 22 September 2016. Accepted: 10 February 2017.