Network Tomography and Partial Least Squares for Traffic Matrix Estimation

Francisco J. Cuberos
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
francicm92@correo.ugr.es

Irene Herrera
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
irenehl@correo.ugr.es

Katarzyna Wasielewska
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
k.wasielewska@ugr.es

José Camacho
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
josecamacho@ugr.es

***Abstract*—The traffic matrix is a useful data structure in network management, monitoring, optimization and traffic forecast. A recurrent problem is to obtain accurate traffic matrices in real time from the traffic of a network, specially when this network is large (e.g., a Tier 1 Internet Service Provider), and without causing a relevant overhead in network computing, storage and communication resources. A solution deeply investigated in the past is the network tomography: the estimation of a traffic matrix from the volume of traffic traversing the links (a.k.a. link counts), which measurement implies a minimum overhead. This estimation entails relevant challenges. In this paper, we propose the application of the Partial Least Squares method to this problem. We illustrate the proposal with the Abilene network dataset, and report promising results in comparison to traditional methods like General Tomogravity and the Structural Analysis based on Principal Component Analysis.**

***Index Terms*—traffic matrix, network tomography, link counts, partial least squares, Abilene data set**

I. Introduction

There is an increasing interest in the development of new data analysis methods to improve the performance of communication networks by providing them with some level of autonomy, in tasks like network monitoring, troubleshooting and optimization [1]. In this context, the measurement of the volume of network traffic has become a crucial need in order to accommodate high quality services at a reasonable investment in technology. A Traffic Matrix (TM) is a widely used data structure in important network problems such as resources optimization, planning and monitoring, resilience analysis, anomaly detection and more [2]. A TM represents the volume of traffic per interval of time between each pair of ingress and egress nodes in the network and provides essential information about the current network state.

The computation of the TM is challenging. A TM can be directly measured using a traffic flow registering protocol such as Netflow [3]. However, this requires a non-negligible consumption of processing time, storage capacity, and network bandwidth. An alternative is to estimate the TM, reducing the overhead. There are several methods that attempt to derive the TM from link counts, which can be collected via the Simple Network Management Protocol (SNMP). This is generally referred to as network tomography, and was proposed two decades ago [4]. A link count makes reference to the number of incoming and outcoming packets or bytes in a link of the network. Unlike Netflow, SNMP neither describes traffic at flow level nor provides origin-destination (OD) information, and it is much less resource consuming.

Many techniques for estimating the TM have been proposed, but these methods are not ideal and new approaches are being sought. One of the simplest approaches is to use a gravity model that is based on the probability theory [5]. The General Tomogravity (GT) approach uses the gravity model and estimates the TM solely from link load data [6]. The Structural Analysis based on Principal Component Analysis (SA-PCA) is a predictive technique based on a regression model that captures the relationship between the link load and the TM [7]. For this purpose, the origin-destination measurements at the flow level must be combined with link counts during a so-called calibration period to build a prediction model. This calibration entails some challenges. In particular, the link counts may show a collinear behavior and low signal-to-noise ratio to accurately estimate the TM. One way to tackle these problems is to use multivariate methods such as PCA. However, PCA is not a regression technique and therefore its application to regression problems is sub-optimal.

In this paper, we propose to use the Partial Least Squares (PLS) regression method [8] to construct a prediction model for TM estimation. PLS, a widely developed technique in fields such as econometrics, psychometrics, and chemometrics, can be seen as the regression counterpart of PCA, more suited to the TM estimation problem. To our knowledge, this is the first application of PLS in this problem.

This work was supported by the Agencia Estatal de Investigación in Spain, grant No PID2020-113462RB-I00, and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 893146.

The rest of the paper is organized as follows. Section II reviews the principal and recent works on TM estimation. Section III introduces the GT and SA-PCA TM estimation approaches. Section IV presents the PLS approach. Section V introduces the calibration strategy, the Abilene network dataset and the performance metrics considered in the experimental part. Section VI presents the evaluation results. Finally, Section VII draws conclusions and establishes future research plans.

II. Related Work

The main problem in the TM estimation with network tomography is that the link counts do not provide enough information to estimate all TM coefficients. One of the strategies developed to improve the TM estimation performance was based on assumptions. A direct consequence of this approach is that the estimation results depend on how accurately these assumptions correspond to reality. An alternative approach to the one based on assumptions is the use of additional information, which unfortunately means an additional overhead. This approach can be found in the tomogravity method, which is based on a combination of link counts from the inner network and the incoming and outgoing traffic counts [6].

Another strategy is to create a prediction model between the counts and the TM. For this purpose, Netflow measurements (or the like) should be collected during the calibration process. Nevertheless, the derivation of the predictive model is not straightforward as there are numerical problems related to the fact that the link count information is insufficient to estimate the TM. Collinearity between the link counts contributes to this problem. Lakhina *et al.* proposed the use of PCA to reduce the dimension of the traffic matrix estimation problem, which in turn reduces the related problems [9]. Another proposal is to use a deep learning architecture [10]. Yang *et al.* introduced a method to improve the accuracy of the estimation based on the sparsity of TM and a grey predictive model [11]. Zhou *et al.* proposed an estimation method called MNETME, which consists of a Moore-Penrose inverse based neural network approach combined with an expectation maximization approach [12].

An alternative trend to TM estimation is to reduce the TM measurement overhead by, for example, using partial direct measurements. This is a problem very similar to the link count estimate (the network tomography) and there are similar difficulties too. Zhou *et al.* proposed a method based on multi-Gaussian models and Bayesian inference to estimate the missing traffic data [13]. The idea of measuring a small number of OD flows at a time, where the measured OD flows change in time and are randomly selected, was proposed [14]. Nie proposed an compressive sensing measurement model based on a stochastic Bernoulli matrix and partial measurements of the TM [15].

The search for new methods that could increase the efficiency and effectiveness of TM measurements continues to this day. Recently, there has also been some work on TM estimation in the Software Defined Networking (SDN) area. Polverini *et al.* defined an effective criterion based on a flow spread concept that allows to select the optimum set of OD flows to be measured in SDN [16]. Other performance improvements for TM estimation were introduced in references [17]–[19].

Our work lies on the group of methods that create a prediction model between link counts and the TM. We use the PLS technique to create this model. PLS is a regression technique that is very efficient in prediction tasks in highly multivariate problems.

III. Traffic Matrix Estimation based on GT and SA-PCA

A TM can be represented as a temporal series of matrices denoted as $\mathbf{X}_t$. Each matrix $\mathbf{X}_t$ in the series has dimension $n \times n$, where n is the number of ingress/egress nodes in the network. Let each element of $\mathbf{X}_t$ in row i and column j be denoted $x_t^{i,j}$, and contain the volume of traffic between the OD pair $\{i, j\}$ at time interval *t*. As we mentioned in Section 1, the TM can be directly obtained using Netflow, which stores flow-level data. For the sake of simplicity of notation, we drop the suffix *t* in the remaining unless necessary.

Let us denote $\mathbf{x}$ as the column vectorized form of $\mathbf{X}$, and $\mathbf{y}$ as the column vector with bytes/packets counts for the L link interfaces of the network at a certain sample interval *t*. We can regress $\mathbf{x}$ on $\mathbf{y}$ to create a prediction model for the former, so that in future time we only need to measure $\mathbf{y}$ to estimate $\mathbf{x}$, reducing the overhead.

The estimation of $\mathbf{x}$ from $\mathbf{y}$ can be improved by introducing first-principles information in the prediction model. For instance, one extensively used result is the so-called tomography relationship between the TM and the link counts using routing information [4]. The expression is as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1}$$

where $\mathbf{A}$ is the routing matrix. When non-bifurcating routing is imposed, $\mathbf{A}$ is a binary matrix where each element $a^{i,j}$ equals to 1 if the traffic of the *i*-th OD pair is routed through the *j*-th link in the network. In large-scale IP networks, the number of links L is often much lower than the number of OD pairs $N = n^2$. For this reason, obtaining $\mathbf{x}$ from $\mathbf{y}$ is an under-determined problem.

The Gravity model assumes that the volume of traffic between two nodes is proportional to the traffic entering through the ingress node and the traffic leaving through the egress node. Authors in [6] define the Generalized Gravity model as the extension of the Gravity model where specific network characteristics are considered. In particular, they consider large ISPs with non-bifurcating connections between networks clients and with other, peer networks. The Tomogravity model is based on the combination of the Gravity model and tomography relationship [6], [20], [21]. The TM estimate is initialized with the Gravity model using the link counts. Then, the estimate is re-adjusted using the Moore-Penrose pseudo-inverse to minimize the network tomography equivalence $||\mathbf{A}\mathbf{x} - \mathbf{y}||$, where $||.||$ is the L_2 norm of a vector. Depending on whether the Simple Gravity model or the Generalized Gravity model is used to estimate the initial solution, the resulting model is called Simple Tomogravity or General Tomogravity, respectively.

Using pairs of measurements of $\mathbf{x}$ and $\mathbf{y}$ collected in coincident time intervals $t = 1, ..., T$, we can establish a data-driven predictive model. This model can be used to infer the value of future TMs only from the link counts, reducing the overhead. Let us denote[1] $\underline{\mathbf{X}}$ as a $T \times N$ matrix containing T instances of $\mathbf{x}$ measurements in the rows, and $\mathbf{Y}$ a $T \times L$ matrix with the corresponding set of $\mathbf{y}$ measurements in the same intervals. The relationship between $\underline{\mathbf{X}}$ and $\mathbf{Y}$ can be established with a purely data-driven approach using a linear regression model as follows:

$$\underline{\mathbf{X}} = \mathbf{YB} + \mathbf{F} \tag{2}$$

where $\mathbf{B}$ is the matrix of regression coefficients and $\mathbf{F}$ is the residual matrix.

The SA-PCA method employs PCA to provide an estimate for $\mathbf{B}$ from the tomography relationship in Eq. (1). PCA transforms a matrix into a number of uncorrelated linear combinations of the original variables, each combination orthogonal to the rest. These linear combinations are obtained to maximize variance, and are referred to as Principal Components (PCs). These PCs capture most of the information in a matrix, and allow to overcome inversion problems. A version of PCA is based on the Singular Value Decomposition (SVD) and follows:

$$\underline{\mathbf{X}} = \mathbf{USV}^T + \mathbf{E} \tag{3}$$

where $\mathbf{S}$ is a AxA diagonal matrix with the singular values of $\underline{\mathbf{X}}$, for A the number of PCs, $\mathbf{U}$ is a TxA matrix that contains the eigenvectors of $\underline{\mathbf{X}} \cdot \underline{\mathbf{X}}^T$, $\mathbf{V}$ is a NxA matrix that contains the eigenvectors of $\underline{\mathbf{X}}^T \cdot \underline{\mathbf{X}}$, and $\mathbf{E}$ is the residual matrix. The number of PCs can be selected by cross-validation [23].

Let us extend the tomography relationship to the time-series matrices:

$$\mathbf{Y} = \underline{\mathbf{X}}\mathbf{A}^T + \mathbf{G} \tag{4}$$

The combination of Eqs. (4) and (3) yields:

$$\mathbf{Y} = \mathbf{USV}^T\mathbf{A}^T + \mathbf{H} \tag{5}$$

where $\mathbf{H}$ combines the error terms of PCA and the tomography relationship. Let us rename $\mathbf{Q} = \mathbf{SV}^T\mathbf{A}^T$. Using the pseudo-inverse of this matrix leads to the regression term of SA-PCA:

$$\mathbf{B}_{SP} = \mathbf{Q}^{+}\mathbf{SV}^T \tag{6}$$

[1]The underline notation is often used for 3-way arrays or tensors, which is suitable given $\underline{\mathbf{X}}$ can also be seen as a time-wise unfolded matrix [22] with three modes: time intervals $\times$ ingress/egress nodes $\times$ ingress/egress nodes.

IV. Partial Least Squares for Traffic Matrix Estimation

The least-squares solution for $\mathbf{B}$ in Eq. (2) is:

$$\mathbf{B}_{LS} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\underline{\mathbf{X}} \tag{7}$$

which may suffer from numerical problems if the inversion is ill-conditioned, or be even unfeasible if $\mathbf{Y}^T\mathbf{Y}$ is rank-deficient. Additionally, this solution does not take into account the tomography relationship. However, by definition, $\mathbf{B}_{LS}$ minimizes the Frobenious norm of the error $\mathbf{F}$ in Eq. (2).

A well-known solution to the numerical problems in least squares is the PLS regression method [8]. The PLS model, particularized to the problem under study, can be stated as follows:

$$\mathbf{Y} = \mathbf{TP}^T + \mathbf{F}_y \tag{8}$$

$$\underline{\mathbf{X}} = \mathbf{TQ}^T + \mathbf{F}_x \tag{9}$$

where $\mathbf{T}$ is the $T \times A$ score matrix, for A now the number of LVs, $\mathbf{P}$ and $\mathbf{Q}$ are the $L \times A$ and $N \times A$ loading matrices, and $\mathbf{F}_y$ and $\mathbf{F}_x$ are the $T \times L$ and $T \times N$ residual matrices of $\mathbf{Y}$ and $\underline{\mathbf{X}}$, respectively[2]. This can be re-arranged into a regression equation as follows:

$$\underline{\mathbf{X}} = \mathbf{Y}\hat{\mathbf{B}}_{PLS} + \mathbf{F}_x \tag{10}$$

with:

$$\hat{\mathbf{B}}_{PLS} = \mathbf{W}(\mathbf{P}^T \cdot \mathbf{W})^{-1}\mathbf{Q}^T \tag{11}$$

where $\mathbf{W}$ is the $N \times A$ matrix of weights.

A PLS model can be obtained, among others, using the kernel algorithm [25], NIPALS [26] or SIMPLS [27]. Cross-validation can be used to determine the optimum A.

V. Materials and Methods

In this section, we present the calibration strategy, the dataset and performance metrics used in the experimentation.

A. Calibration strategy

A predictive model in a time series tends to reduce performance in time. In order to maintain an adequate performance of the predictive model between the links and the TM, we can define a strategy to re-calibrate the model after fixed periods of time, every $\triangle_t$ sampling times. The re-calibration is based on C_t samples, in which we need to measure both the TM and the links traffic. In the remaining $\triangle_t - C_t$ sampling times of each period of time, the TM is estimated but not measured. If, for example, the combination $C_t = 2$ and $\triangle_t = 72$ is selected, in 2 samples out of every 72 we measure both the TM (through, e.g., Netflow) and the links traffic, and in the remaining 70 we only measure the latter. We use this calibration strategy in SA-PCA and PLS.

[2]Note that an alternative and equivalent expression for the PLS model makes use of slightly different scores and an additional inner regression parameter [24].

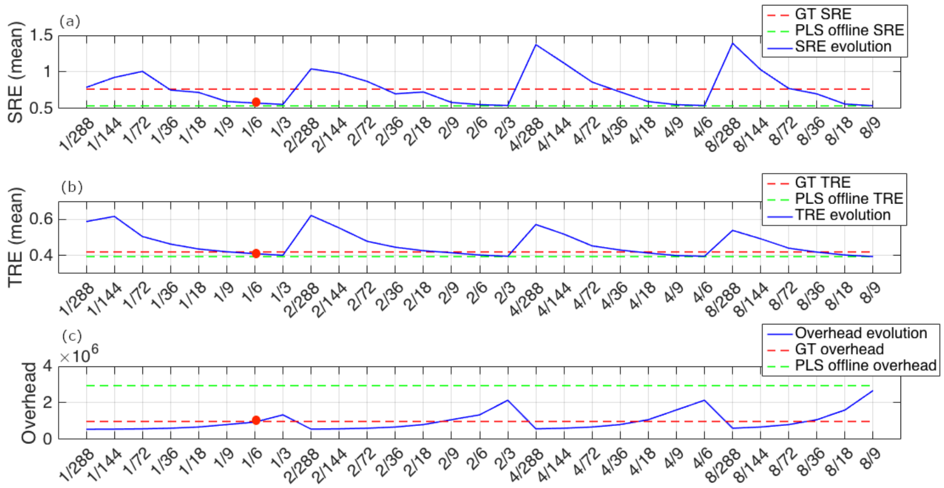

Fig. 1. SRE (a), TRE (b) and OVH (c) evolutions of all the possible C_t and $\triangle_t$ combinations (x-axis) versus GT and *offline* PLS.

B. Dataset

The Abilene network data [28] has been widely used in the TM estimation area to test the performance of new algorithms. The dataset was formed in 2004 by 24 weeks of traffic (X01-X24 files) and contains 144 OD pairs. The Abilene network includes 54 links classified into three categories: internal, inbound and outbound. In our research, we only considered internal links. Due to some time discontinuity in the measurements we used 20 weeks of the total of 24. In order to validate the PLS-based TM estimation, the dataset was divided into two subsets: one for calibration and one for testing the results. The calibration set consisted of the first ten weeks and was used to select the optimal calibration strategy. The last ten weeks were used to evaluate the performance of the selected algorithm.

C. Performance metrics

We used several metrics to evaluate the proposed method: the Temporal Relative Error (TRE), the Spatial Relative Error (SRE) [29] and the information overhead. The $TRE(t)$ computes the average sum-of-squares error between the true TM $\mathbf{X}_t$ and its estimate $\hat{\mathbf{X}}_t$ at time t, using any of the aforementioned methods. The $SRE(k)$ computes the average sum-of-squares error between the true traffic and its estimate at the k-th flow. The *overhead* (OVH) refers to the amount of information (in bytes) that has to be sent through the network to the collector station. The combination of TRE and SRE allows to understand the estimation error in the time and OD pair dimension, respectively. The combination of the error metrics with the OVH allows us to assess methods as a trade-off between the estimation performance and the network overhead, which is relevant for real practice.

VI. Experiments and Results

In this section, the performance evaluation results are presented. In a first step, we used the first 10 weeks of data to select the configuration (calibration strategy) in PLS. The remaining data is used to evaluate the performance and compare PLS with GT and SA-PCA.

A. PLS model selection: First ten weeks

The mean performance values of the different calibration strategies in PLS and for the first ten weeks are shown in Fig. 1. For the sake of a fair comparison, we select a PLS calibration strategy with a similar overhead to GT ($OVH_{GT} = 18144$ samples $* \ 54$ links $= 979776$ Bps)[3]. The closest overhead value is the one given by the combination $C_t = 1$ and $\triangle_t = 6$ (marked as a red filled circle in the figure). The blue solid line represents the performance of the variants of PLS, for each possible combination of the values of C_t and $\triangle_t$. We can see that as the overhead increases, the SRE and TRE decrease. We compare these results with the one of GT, and the hypothetical (but not useful in practice) PLS model calibrated with all sampling times (i.e., for $C_t = \triangle_t$). We call this a PLS offline model, and it represents the optimal achievable performance in PLS in terms of SRE and TRE. We can see that the selected model (red circle) clearly outperforms GT in terms of SRE, approaching the performance of the PLS offline model. In the case of TRE, the three methods provide a similar result.

B. Performance evaluation: Last ten weeks

Once the parameters of the algorithm had been selected, they were applied on the second half of the dataset to compare the performance of the methods. In this case, we included GT, SA-PCA and PLS, the latter two for the configuration $C_t = 1$ and $\triangle_t = 6$ and their corresponding offline versions ($C_t = \triangle_t$). Recall that offline versions are used to show the limit of achievable performance, but they cannot be used in practice. The numerical results can be found in Table I. We

[3]Later on, we will choose a SA-PCA variant with the same overhead

can see that for a similar overhead, PLS outperforms the other methods.

TABLE I
PERFORMANCE RESULTS

	Calibration	*TRE*	*SRE*	*OVH*
GT	–	0.4474	0.8456	979796
SA-PCA	$C_t = 1$, $\triangle_t = 6$	0.4748	0.5789	943488
PLS	$C_t = 1$, $\triangle_t = 6$	0.4172	0.5713	943488
SA-PCA	*offline*	0.4780	0.5697	2939328
PLS	*offline*	0.3971	0.5146	2939328

VII. CONCLUSIONS

In this paper, we present for the first time the application of Partial Least Squares to the problem of estimation of the traffic matrix using a tomography approach. We also propose a model calibration strategy to find a compromise between network overhead associated to traffic matrix measurements and estimation performance. A reduction in overhead can be advantageous when implementing the proposed methodology in a real network. Our approach inherits the advantages and disadvantages of the tomography approach: we reduce the overhead to compute the traffic matrix at the cost of introducing some estimation error. The results of the experiments show that our method generally outperforms state of the art approaches in terms of accuracy of estimation for the same overhead. This approach may still be improved by including first principle results in the estimation model. In future work, we plan to verify the method on other network deployments and technologies, in particular in Software Defined Networks.

REFERENCES

[1] H. Song, F. Qin, P. Martinez-Julia, L. Ciavaglia, and A. Wang, "Network Telemetry Framework," Internet Engineering Task Force, Internet-Draft draft-ietf-opsawg-ntf-07, Feb. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-ntf-07

[2] P. Tune and M. Roughan, "Internet traffic matrices: A primer," *Recent Advances in Networking*, 2013.

[3] Cisco Systems, "Introduction to Cisco IOS ® NetFlow," *White Paper*, 2012.

[4] Y. Vardi, "Network Tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, pp. 365–377, 1996.

[5] M. Roughan, "Simplifying the synthesis of internet traffic matrices," *SIGCOMM Computer Communication Review*, vol. 35, no. 5, p. 93–96, Oct. 2005. [Online]. Available: https://doi.org/10.1145/1096536.1096551

[6] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," *SIGMETRICS Performance Evaluation Review*, vol. 31, no. 1, p. 206–217, Jun. 2003. [Online]. Available: https://doi.org/10.1145/885651.781053

[7] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, p. 61, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1012888.1005697

[8] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 109–130, 2001.

[9] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, p. 61–72, Jun. 2004. [Online]. Available: https://doi.org/10.1145/1012888.1005697

[10] L. Nie, D. Jiang, L. Guo, and S. Yu, "Traffic Matrix prediction and estimation based on deep learning in large-scale ip backbone networks," *Journal of Network and Computer Applications*, vol. 76, no. C, p. 16–22, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2016.10.006

[11] J. Yang, X. Huang, and S. Jiang, "An accurate approach for traffic matrix estimation in large-scale backbone networks," in *15th International Symposium on Parallel and Distributed Computing (ISPDC)*. IEEE, 2016, pp. 425–431.

[12] H. Zhou, L. Tan, Q. Zeng, and C. Wu, "Traffic matrix estimation: A neural network approach with extended input and expectation maximization iteration," *Journal of Network and Computer Applications*, vol. 60, pp. 220–232, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515002854

[13] H. Zhou, D. Zhang, and K. Xie, "Accurate traffic matrix completion based on multi-gaussian models," *Computer Communications*, vol. 102, no. C, p. 165–176, 2017. [Online]. Available: https://doi.org/10.1016/j.comcom.2016.11.011

[14] G. Liang, N. Taft, and B. Yu, "A fast lightweight approach to origin-destination ip traffic estimation using partial measurements," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2634–2648, 2006.

[15] L. Nie, "Traffic matrix estimation approach based on partial direct measurements in large-scale IP backbone networks," in *IEEE 5th International Conference on Electronics Information and Emergency Communication*, 2015, pp. 178–181.

[16] M. Polverini, A. Baiocchi, A. Cianfrani, A. Iacovazzi, and M. Listanti, "The power of SDN to improve the estimation of the ISP traffic matrix through the flow spread concept," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1904–1913, 2016.

[17] Y. Tian, W. Chen, and C.-T. Lea, "An sdn-based traffic matrix estimation framework," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1435–1445, 2018.

[18] S. Xu, M. Kodialam, T. V. Lakshman, and S. S. Panwar, "Learning based methods for traffic matrix estimation from link measurements," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 488–499, 2021.

[19] A. Kumar, S. Vidyapu, V. V. Saradhi, and V. Tamarapalli, "A multi-view subspace learning approach to internet traffic matrix estimation," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1282–1293, 2020.

[20] S. Eum, J. Murphy, and R. J. Harris, "A failure analysis of the Tomogravity and EM methods," in *TENCON 2005 - 2005 IEEE Region 10 Conference*. IEEE, 2005, pp. 1–6.

[21] M. M. Rahman, S. Saha, U. Chengan, and A. S. Alfa, "Ip traffic matrix estimation methods: Comparisons and improvements," in *2006 IEEE International Conference on Communications*, vol. 1. IEEE, 2006, pp. 90–96.

[22] J. Camacho, J. Picó, and A. Ferrer, "Bilinear modelling of batch processes. part i: theoretical discussion," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 22, no. 5, pp. 299–308, 2008.

[23] J. Camacho and A. Ferrer, "Cross-validation in pca models with the element-wise k-fold (ekf) algorithm: Practical aspects," *Chemometrics and Intelligent Laboratory Systems*, vol. 131, pp. 37–50, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743913002335

[24] A. Höskuldsson, "Pls regression methods," *Jurnal of Chemometrics*, vol. 2, pp. 211–228, 1988.

[25] B. Dayal and J. MacGregor, "Improved PLS algorithms," *Journal of Chemometrics*, vol. 11, pp. 73–85, 1997.

[26] H. Wold, *Multivariate Analysis*. Academic Press, New York, 1966, ch. Estimation of principal components and related models by iterative least squares.

[27] S. de Jong, "Simpls: An alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251 – 263, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016974399385002X

[28] "Abilene Dataset." [Online]. Available: http://www.cs.utexas.edu/ yzhang/research/AbileneTM/

[29] E. Zhao and L. Tan, "A pca based optimization approach for ip traffic matrix estimation," *Journal of Network and Computer Applications*, vol. 57, pp. 12–20, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515001691