# Traffic matrix estimation

Olga Grebenkova
*Optimization Class Project. MIPT*

## Introduction

Despite a large body of literature and methods devoted to the Traffic Matrix (TM) estimation problem, the inference of traffic flows volume from aggregated data still represents a major issue for network operators. Directly and frequently measuring a complete TM in a large-scale network is costly and difficult to perform due to routers limited capacities. Many approaches of TM estimating were proposed [1, 2, 3]. In this project we compare two methods — linear programming [1] and algorithm with Random Neural Network [4].

## Formulation of the problem

Let $c$ be the number of origin-destination (OD) pairs. Although conceptually traffic demands are represented in a matrix , with the amount of data transmitted from node $i$ to node $j$ as element $X_{ij}$ , it is more convenient to use a vector representation. Thus, we order the OD pairs and let $X_j$ be the amount of data transmitted by OD pair $j$. Let $(\mathbf{Y} = (y_1, \dots, y_r)^\top)$ be the vector of link counts, where $y_l$ gives the link count for link $l$ and $r$ denotes the number of links in the network. The vectors $X$ and $T$ are related through an $r$ by routing matrix $A$. $A$ is a $\{0, 1\}$ matrix with rows representing the links of the network and columns representing the OD pairs. The OD flows are thus related to the link counts according to the following linear relation:

$$\mathbf{Y} = \mathbf{AX}$$

## Linear Programming

Knowing that the link count $Y_l$ has to be the sum of all the traffic demands that use link $l$ , the LP model is defined as the optimization of an objective function:

$$\max_{X_a} \sum_{a=1}^{c} w_a X_a,$$

where $w_j$ is a weight for OD pair $j$.
The objective function is subject to link constraints:

$$\sum_{a=1}^{c} A_{s,a} X_a \le Y_s \quad s = 1, \dots, r$$

$$X_a \ge 0 \quad \forall a$$

$$\sum_{s=(i,j)} Y_s A_{s,k} - \sum_{s=(j,i)} A_{s,k} = \begin{cases} X_k, & j \ \ src \ \ of \ \ k \\ -X_k, & i \ \ dst \ \ of \ \ k \\ 0, otherwise \end{cases}$$

## Random Neural Network

**The random neural network (RNN)** is a mathematical representation of an interconnected network of neurons or cells which exchange spiking signals. Each cell state is represented by an integer whose value rises when the cell receives an excitatory spike and drops when it receives an inhibitory spike. The spikes can originate outside the network itself, or they can come from other cells in the networks. Cells whose internal excitatory state has a positive value are allowed to send out spikes of either kind to other cells in the network according to specific cell-dependent spiking rates. The model has a mathematical solution in steady-state which provides the joint probability distribution of the network in terms of the individual probabilities that each cell is excited and able to send out spikes.

The main idea of our method is to find a certain non-linear transfer-block $f_k(\cdot) : R^{n_k} \to R$ for each OD flow k, such that:

$$X_t(k) = f_k(\mathbf{Y_t}(\delta) = f_k(Y_t(\delta_k^1), Y_t(\delta_k^2), \dots, Y_t(\delta_k^{n_k}))$$

## Experiment

We use both methods on self generated data and the real data from the Anaheim network for 1992. During training our RNN the calibration of each block $f_k(\cdot)$ is performed by supervised learning, using a learning dataset composed of T input-output pairs $\{Y_t(\delta_k), x_t(k)\}$ , $t = 1, \dots, T$. The set of N neurons is divided into three subsets: a set of I input neurons, a set of H hidden neurons, and a set of O output neurons.

$n_k$ — 10 — 1

figures/Processoflearning-eps-converted-to.pdf

## Results

RNN is still in process.

figures/comparewithlinear-eps-converted-to.pdf

## Conclusion

As we can see RNN model works better than linear one. However, it is still not accurate.

## References

[1] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Computer Communication Review*, volume 32, pages 161–174, oct 2002.

[2] Claudia Tebaldi and Mike West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–573, jun 1998.

[3] Dingde Jiang and Guangmin Hu. Large-scale IP traffic matrix estimation based on backpropagation neural network. In *Proceedings - The 1st International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2008*, 2008.

[4] Pedro Casas and Sandrine Vaton. On the use of random neural networks for traffic matrix estimation in large-scale IP networks. In *IWCMC 2010 - Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, pages 326–330, 2010.