OD MATRICES ESTIMATION FROM LINK FLOWS BY NEURAL NETWORKS AND PCA

Lorenzo Mussone[1] , Matteo Matteucci[2]

[1]*Department of Building Environment Science Technology, Politecnico di Milano, Italy*
mussone@polimi.it, via Bonardi, 9, 20123 Milano
[2] *Department of Electronic and Information, Politecnico di Milano, Italy*
matteucc@elet.polimi.it, via Ponzio,34/5, 20123 Milano

Abstract: The paper tackles OD matrix estimation starting from the measures of flow on road network links and proposes the application of soft-computing techniques. The application scenarios are two: a trial network and the real rural network of the Province of Naples both simulated by a micro-simulator dynamically assigning known OD matrices. A PCA (Principal Component Analysis) technique was also used to reduce the input space of variables in order to achieve better significance for input data and to study the possible eigengraphs of the road networks.

Keywords: OD estimation, Neural Networks, PCA, Link flow measures, Variance stabilization

1. INTRODUCTION

The analysis of urban networks was organically and significantly developed since the middle of 80s in order to solve problems related to road circulation. Many planning and control methodologies both with equilibrium and dynamic approaches are based on the use of an origin destination matrix (OD). Field survey necessary to build this matrix is expensive and cannot be repeated with a high frequency. For this reason, methodological and operating alternatives have been experimented since many years and they aim at building the OD matrix by using link flows that generally cost less to be measured.

In order to solve the problem of OD estimation many approaches have been developed. Some are based on entropy maximization that is on the maximization of trip distribution dispersion on all available paths; in some cases the built model refers to an OD matrix objective without referring indeed to estimation errors, or to statistic estimation indexes or to likelihood functions (Van Zuylen and Willumsen, 1980). This model was later extended to congested networks by formulating an optimization problem with variational disequality constraints leading to a bi-level programme. The bi-level approach presents some difficulties to find the optimal solution because of non-convex and non-differential formulation. Florian and Chen (1995) formulated a heuristic approach (of Gauss-Seidel type) capable of converging to optimal solution by limiting the objective to the correction of O/D matrix.

Other approaches are based on models that use the statistical properties of observed variables. For instance Maher (1983) proposed a Bayesian estimation by means of a normal multivariate distribution both for matrix distribution and for link flow; Cascetta (1984) used an estimation based on generalized least squares (GLS). An overview of statistical methods for estimating OD matrix can be found in Cascetta and Nguyen (1988); it regards generalized and constrained least squares and estimation of likelihood and Bayesian type. The most of these studies assumes a fixed percentage of link or path choice calculated by a deterministic user equilibrium assignment model. This can cause some inconsistency between flows and the OD matrix especially when the network is highly congested. In order to overtake this limitation, Cascetta (1989) proposed to interpret link and path flows like stochastic variables and, therefore, values obtained by a SUE assignment like the average values of these variables. On the same line there are other papers such as Yang et al. (1992), and in a successive improvement in (Yang et al., 2001); Lo et al. (1996) proposed a unified statistical approach for the estimation of OD matrix using simultaneously link flow data and information about link choice percentage. Gong (1998) uses a Hopfield neural network as a tool for solving an optimal problem,

formulated however as an entropy maximization problem, of the same type cited before.

The aim of this paper is to solve OD matrix estimation by soft computing techniques, specifically neural networks, starting from the knowledge of flow measures on road network links. It works out an application of multilayer feed-forward neural networks in order to estimate OD matrices by using the well-known approximation property typical of these models; without loosing generality, the existence of a continuous relationship between flow measured on links and OD matrix that produces them is assumed. Because of the learning mechanism of feed-forward neural networks, however, the contemporary knowledge of OD matrix and related link flows is required for the training set. This requirement is achieved thanks to the laboratory of the transport group of the University of Naples (Bifulco, 2004) that produced all necessary information usually not easy to collect on field with an adequate time detail and with the necessary completeness.

2. DATA PRE-PROCESSING

Signal cleaning and normalization are fundamental to get an easier training of neural networks in the following. The basic idea is to reduce the problem to stationary conditions or, if not possible, to stabilize the relationship between the signal mean and its variance. In this paper, we focus mainly on signal variance stabilization; we have no missing data or wrong measurements.

In literature there are a few definitions of stationarity; in a strong sense we can state complete statistical stationarity for the observed process, but this is impractical since this requires specifying infinite constraints on the moments of the distribution (Bittanti, 1986). A simpler approach reduce the problem of stationarity to second order statistics (i.e., the signal distribution can be completely described using the first two moments) by imposing a constant expected value $E[(v(t))] = m,\ \forall t$ and a covariance function $\gamma_v(\tau) = E[(v(t)-m)(v(t+\tau)-m)]$ independent from specific time indexes, but depending only on their difference $\tau = t_2 - t_1$. In the following we use this weaker definition of stationarity assuming that expected value and covariance are invariant with respect to time shifts.

The interest on (weak) stationary processes is related to the classical least squares minimization approach. Also learning neural network weights by back-propagation pertains to this framework since the error function minimized by training procedure, being this the classical gradient descent or another training algorithm, is the sum of squared errors between the (*K*-dimensional) target value $\vec{t}_n$ and the network output $\vec{y}_n$:

$$\sum_n^N \left\| \vec{t}_n - \vec{y}_n(\vec{w}) \right\|^2 = \sum_n^N \sum_k^K \left(t_n^k - y_n^k(\vec{w}) \right)^2 \qquad (1)$$

Learning by using this error function can be seen as the maximum likelihood estimation of neural network's parameters (i.e. the weights) under the hypothesis of $\vec{t}$ being a corrupted version of $\vec{y}(\vec{w},\vec{x})$ (i.e., the neural network output) $t_n^k = y_n^k(\vec{w},\vec{x}_n) + \varepsilon$ with Gaussian noise $\varepsilon \sim \mathrm{N}(0,\Sigma)$, thus being $\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_N$ an i.i.d. sample from a Gaussian process $\vec{t} \sim \mathrm{N}(\vec{y}(\vec{w},\vec{x}),\Sigma)$ with mean $\vec{y}(\vec{w},\vec{x})$. If we compute the maximum (log) likelihood estimation of this mean (i.e., the maximum likelihood estimation of neural network weights), it turns out to be equivalent to the classical least squares minimization.

When the process is non-stationary this learning approach does not hold any more (e.g., with non symmetric error/noise in measurements) and pre-processing is required to "normalize" the signal. Whenever the signal remains non-stationary after trend (~f(t)=kt) and seasonality (~f(t)=s(t+wT)) removal (k, w and T are constant), we should investigate the relationship between signal mean and variance; an important example of this relationship is given by a variance that is function of mean: usually we have a variance that increases when mean increases. In these cases, it is useful to apply variance stabilizing transformations (Montgomery e Peck, 1992 and Potts, 1999).

A second issue in learning with neural networks, and modeling in general, is the so-called *curse of dimensionality* (Scott, 1992). Learning in high dimensional spaces is hard and error prone so it is common practice to pre-process data to reduce input and/or output cardinality. Dealing with traffic networks the flow vector has the cardinality of the number of arcs in the network and this can be quite large with complex networks. Since available samples, as well as arcs flows, are highly correlated, we can use this property to reduce significantly input dimensionality by loosing only a small amount of information. Using PCA (Principal Component Analysis) (Lebart et al., 1977) we can reduce dimensionality by preserving information in the form of signal variance. The aim of PCA is to define an orthogonal space (eigenvectors) ordering its bases by the amount of explained signal variance; once this space has been defined we can project our data on the first components by loosing only a small amount of variance and thus loosing a minimal information. The number of components can be directly selected or can be defined as the number of components preserving a given amount of variance. In following experiments we always kept the 99% of variance achieving however a significant reduction for input dimensionality, due to high correlation in link flows.
By plotting the eigenvectors of this (reduced) space on the network graph obtaining what we call 'eigengraphs'; thus we can have a visualization of the links with a higher contribution to the principal

components. In the following plots, arcs with a higher contribution are thicker and their color is red if they have a positive contribution otherwise is blue. The visualization of eigenflows, i.e., the product of the flow matrix and the eigengraphs, shown in the next chapters, is the projection of traffic flows onto the eigengraph space; in the same way as before, arcs with a higher contribution to flow are thicker and their color is red if they have a positive contribution otherwise their color is blue.

3. APPLICATION SCENARIOS

3.1 The simulator

The scenarios referred to during experimentations are produced by a simulation environment for road networks developed by the transport section of the University of Naples. This simulator is able to reproduce dynamic vehicular flow starting from a generic transport demand. Setting up the experiments requires the definition of the transport system, demand and supply, by which every model and procedures must be calibrated. For this reason, a trial network with 6 nodes (among which there are 3 centroids) and 12 links was firstly used (Figure 1).

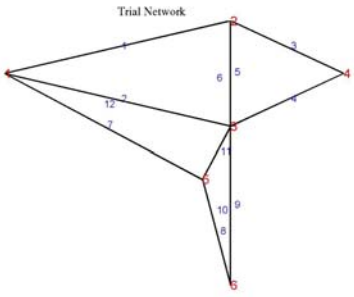


Fig. 1: The simplified network for first experiments.

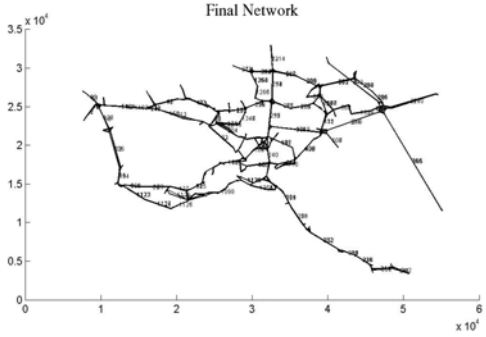


Fig. 2: Schematic network used for the rural road network of the Province of Naples.

Then, with the aim at making more likely the application scenery, a real land system has been surveyed both with its demographic characteristics and activity locations, and actual transport supply. The territorial context is the whole province of Naples from which the rural road network has been extracted (Figure 2). This network has 994 nodes (among which there are 45 centroids) and 1363 links (reduced to 1190 after discharging links with no flow).

3.2 Demand and vehicular flow on links

Demand is characterized by within-day and day-to-day dynamic and it is updated each 15 minutes. Flows on link are detected each five minutes.

For the trial network, demand is described by a 3x3 matrix and the nodes 1 and 6 are origin centroids; the nodes 1, 4 and 6 are destination centroids (for a total of four non-empty cells). Simulation days are 15.

For the network of Naples province (the final network) the scenery is more complex. Demand is described by a 48x48 with 1004 non-empty cells (out of 2304) and with a structure highly variable during the day. Besides the high variability of demand it must be underlined that non-empty cells sets in the 15 minute intervals are generally not overlapping. Cells with less than 10 vehicles per hour are cancelled for two main reasons: firstly to reduce data dispersion and secondly because it is very difficult that these low values have a statistical significance while it is likely they don't give a real information. The final matrix has therefore 726 non empty cells (about the 31% of the total) with a reduction of the number of non-empty cells of about 28%, but with reduction of demand of only about 8.8%. Simulation days are 15 also for this scenario but obviously the amount of data to be used is more than ten times higher (total demand is 41,488,230 veh/h).

For both networks the number of records is 1440 for OD demand and 4320 for link flows. For this reason each row of OD demand (15 minute interval) is related to the three rows of flow of the same 15 interval, leading to 4320 records both for OD demand and link flows.

By using the dataset under examination and observing the plot of seasonality, we can argue that process variance is low when the seasonality value is low and, conversely, it is high when the seasonality value is high. Assuming this, it is reasonable to look for the existence of an analytical relationship between seasonality (basically a mean) and variance. The existence of a relationship between these two variables (plotted in Figure 3) confirms that the process is not stationary. From this figure it can be easy argued that variance and average are related by a non-linear relationship, specifically a quadratic one. In this situation a possible technique to stabilize the variance is to apply a logarithmic transformation. The following results show that this transformation really allows us to improve performance of about some percentage points for the RMSE index with respect to results obtained without stabilization. However, it must be underlined that this result is not general though in transport application it is rather frequent.

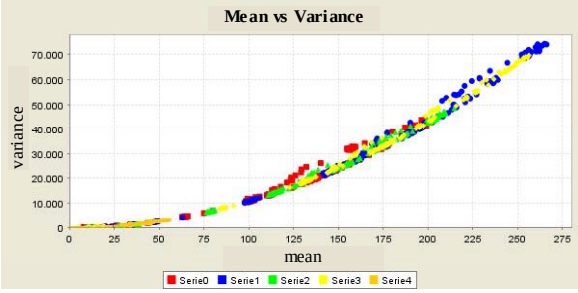


Fig. 3: Relationship between average value and variance in OD data.

4. EXPERIMENTAL VALIDATION

In the application of neural network to O/D estimation we have performed the pre-processing described in Section 2 to flow dataset then stabilizing the variance with a log transformation. The dimensionality reduction has been obtained by projecting the dataset on the eigengraphs. After reduction, the dataset has been divided into three subsets using uniform random sampling; the first subset has been used for training the network, the second one has been used for early stopping and the selection of network topology, while the third subset has been used to assess network generalization performance on data it has never seen. The generalization capability of the model has been evaluated using minimum/maximum/average error and error percentage on the O/D after post-processing the network output to reintroduce variance-mean relationship.

In the test network the total number of components is 12 (i.e., equal to the number of arcs) and the contribution to the variance of the first 5 of them is reported in Table 1.

The first 3 components (eigenvalues) explain the 97% of variance in the signal and the first 5 explain the 99% of signal variance. It is noticeable how the first eigenflow represents the 87% of total signal variance. The graphs in Figure 4 show the components of first 2 eigenvectors on the network graph; the 2 principal eigenflows are represented in Figure 5. From these plots turns out that most significant contribution is given by arcs 1, 2, 3, 4, and 9. In the final network the number of principal components is 1190 and the contribution to the signal variance of the first 10 is reported in Table 2.

Table 1: Variance explained by the first five eigenvalues (trial network).

Eigenvalue number	*Explained variance (%)*
1	87.0
2	5.4
3	4.3
4	1.7
5	0.9

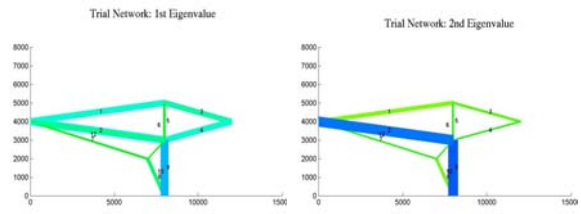


Fig. 4: Graph plots (eigengraphs) of the first two eigenvectors (trial network).

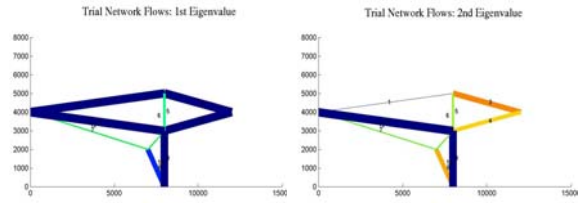


Fig. 5: Graph plots of the first two eigenflows (trial network)

Table 2: Variance explained by first ten eigenvalues (final network).

Eigenvalue number	*Explained variance (%)*	*Eigenvalue number*	*Explained variance (%)*
1	76.43	6	1.05
2	5.75	7	1.00
3	3.82	8	0.70
4	2.59	9	0.58
5	1.29	10	0.55

The first 6 eigenvalues sum up to 90% of explained variance, the first 92 eigenvalues explain the 99% and 494 eigenvalues on 1190 can explain the 99.9%. The four graphs in Figure 6 represent the components of the first 4 eigenvectors while graphs in Figure 7 the first 4 eigenflows. A complete analysis of these plots is quite complex and out of the scope of this paper; however, as a first glance it is possible to notice some sort of backbones in the central part of the network.

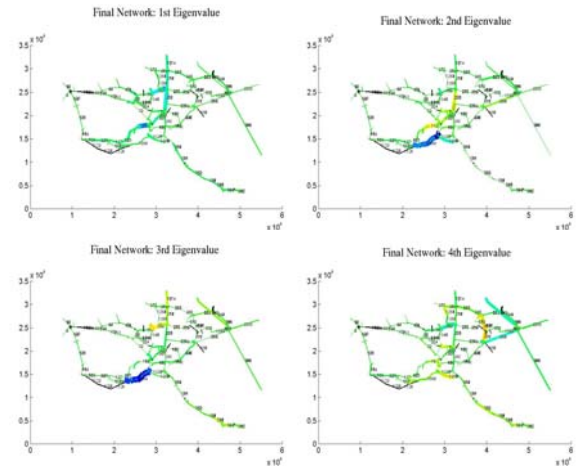


Fig. 6: Graph plots (eigengraphs) of the first four eigenvectors (final network).

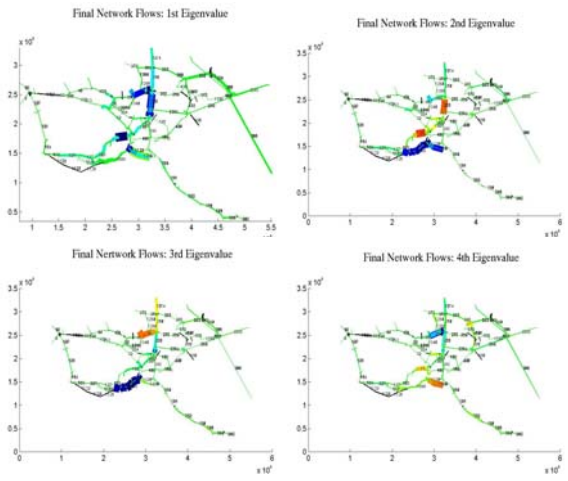


Fig. 7: Graph plots of the first four eigenflows (final network).

The neural network used in this work has a multi layer topology with only feed-forward connections (Figure 8); it is trained using as input the eigengraph projection of flows and the OD matrix that generated those flows as output. The hidden nodes have a hyperbolic tangent activation function while the output layer has a linear one. The topology selected by using cross-validation has 10 hidden neurons for the test network and 50 for the final network; the number of neurons in the input layer depends on the number of components used in reducing the input and the number of neurons in the output layer is equal to the number of ODs. Considering all the components the total number of parameters (i.e. weights) in the two models are respectively 160 for the trial network and 84,450 for the final network. By reducing the explained variance of input to the 99% of total, we reduce the number of parameters in the first model to 90 and in the second model to 40,900.

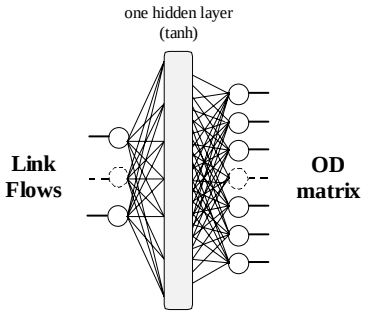


Fig. 8: MLP Neural Network structure for OD estimation.

5. RESULTS

5.1 Trial network

In Figure 9 and 10 results obtained by using the first five eigenvalues (99% of the total variance) for the four ODs are reported. Correlation between predicted and real data is always very high and it does not vary much using five eigenvalues instead of twelve. The correlation for the four ODs with the first five eigenvalues configuration is in the range 0.86-0.89.

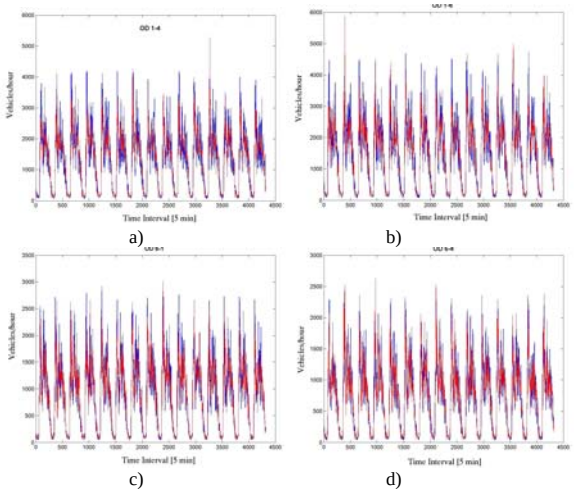


Fig. 9: Predicted and real curves for the four ODs of the trial network: 1-4(a), 1-6(b), 6-1(c), 6-4(d).

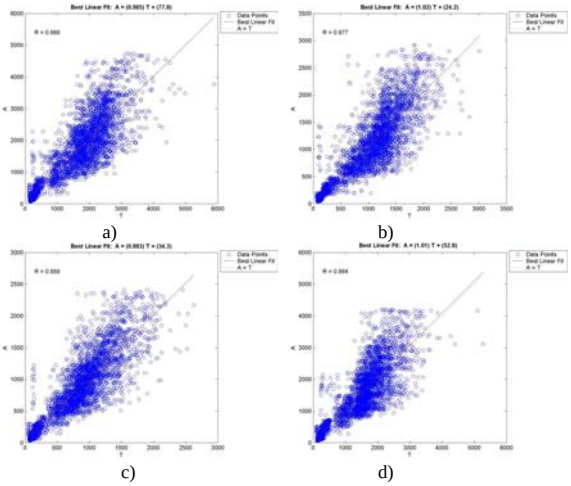


Fig. 10: Correlation between predicted and real OD data (trial network):1-4(a),1-6(b),6-1(c),6-4(d).

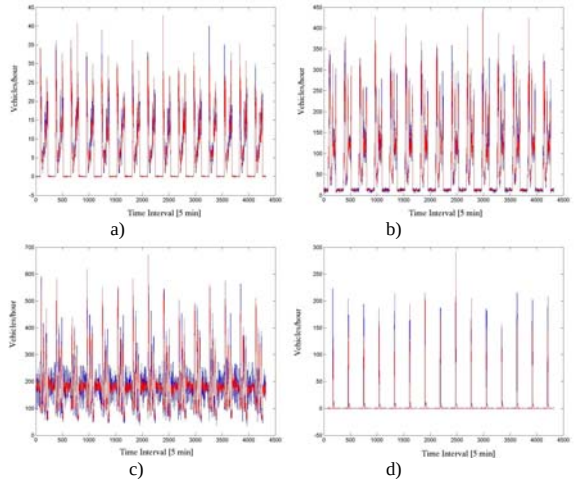
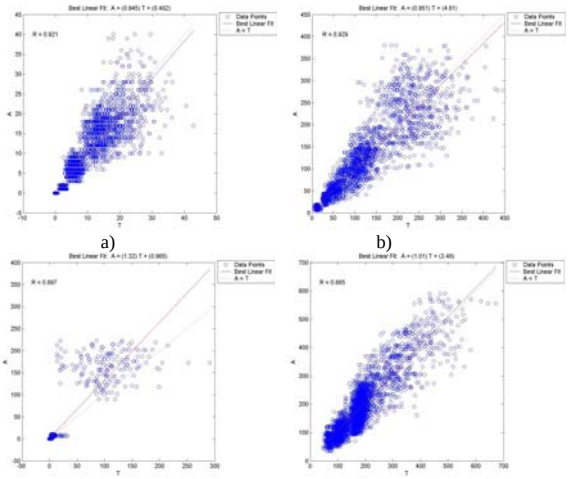


Fig. 11: Predicted and real curves: OD 11-20(a), 12-31(b), 13-36(c), 13-46(d) (final network).

5.2 Final Network

For this network some possible combinations of eigenvalues have been tested. In Figure 11 results for four significant ODs are reported using the first 92 (on 1190) eigenvalues explaining the 99% of the total variance. The average percentage error is 2.14.



a) b) c) d)

Fig. 12: Correlation between predicted and real OD data : OD 11-20(a), 12-31(b), 13-36(c), 13-46(d) (final network).

Correlation between predicted and real data is always very high and in the range 0.81-0.93 (Figure 12). It must be underlined that also in this scenario the neural network is capable of following well enough the signal peaks. This behavior is particularly good for OD 13-46 (item d) that has isolated peaks on a almost nil signal. Low values are generally predicted with a good accuracy. In the case of a signal with a lot of noise (item c) prediction can follow well enough peaks though it is not perfect when predicting fluctuations at low values. Also when the signal has a limited range (item a) prediction follows real data well enough without loosing in precision.

6. CONCLUSIONS

Estimation of OD matrices, based on the use of link flow, has resulted an affordable operation by applying a neural network model. Results obtained also for middle-large dimension networks are particularly stimulating and anyway more than sufficient to prove the potential usefulness.

Experiments have been carried out using flows of all network links. But, it must be stressed that this is not a limit of the approach. Future research will investigate how much performance is reduced by lowering the number of links used for estimation. In order to overtake the problem of statistical significance arising from a limited number of data, techniques of reduction of input space, such as PCA, and pre-processing of data, such as the variance stabilization, have been successfully applied. PCA application has a relevant role not only for achieving considerable model performance but also for evaluating the eigengraphs of the networks and the projection of flows on them. From this graphs it is reasonable to think they contain other information about a convenient reduction of the number of links in order to maintain OD estimation within a good level of performance. Signal stabilization, developed in details in another work, has also an important role and allows us to increase further model performance.

The perspectives of the research are therefore manifold. It must be underlined the strategic importance of the simulator of virtual reality that provided us a rather detailed description of the scenario to be analyzed and modelled. The simulator can be used as a preliminary tool to investigate the network structure and behaviour thanks to eigengraphs and eigenflows, to evaluate links more representative for describing the whole network, to analyze the demand and finally to make the necessary synthesis and stabilization.

REFERENCES

Bifulco G.N., 2004. Road transport systems in information society (in Italian). ISBN 88-7999-857-9, Aracne, Roma.

Bishop, C.M.,1995. Neural networks for pattern recognition. Oxford University Press, Oxford.

Bittanti, S., 1993. Prediction and filter theory (in Italian). Pitagora Editrice, Bologna, 4th ed.

Cascetta E., 1984. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimators. Transpn. Res. 18B, 289-299.

Cascetta E., 1989. A stochastic process approach to the analysis of temporal dynamics in transportation networks. Transpn. Res.23B,1-17.

Cascetta E., Nguyen S., 1988. A unified framework for estimating or updating origin/destination matrices from traffic counts. Transpn. Res. 22B, 437-455.

Florian M., Chen Y., 1995. A coordinate descent method for the bi-level O-D matrix adjustment problem. Int. Trans. Opl. Res. Vol.2, 165-179.

Going Z., 1998. Estimatine the urban OD matrix: a neural network approach. European Journal of operational research 106, 108-115.

Lebart L., Morineau A., Tabard N., 1997. Techniques de la description statistique: méthodes et logiciels pour l'analyse des grands tableaux. Dunod, Paris.

Lo H.P., Lam W.H.K., Zhang N., 1996. Estimation of an origin-destination matrix with random link choice proportions: a statistical approach. Transpn. Res. 30B, 309-324.

Maher M.J., 1983. Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. Transpn. Res.17B, 435-447.

Scott D.W., 1992. Multivariate Density Estimation, Wiley, NY.

Van Zuylen H.J., Willumsen L.G., 1980. The most likely trip matrix estimated from traffic counts. Transpn. Res. 14B, 281-293.

Yang H., Meng Q., Bell M.G.H., 2001. Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. Transportation Science 35, 107-123.

Yang H., Sasaki T., Iida Y., Asakura Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. Transpn. Res. 26B, 417-434.