Calibration of urban road network capacities

Guang Wei*[1], Joakim Ekström[2], and Gunnar Flötteröd[3]

[1,2,3]Department of Science and Technology (ITN), Linköping University, Sweden

[3]Swedish National Road and Transport Research Institute (VTI), Sweden

SHORT SUMMARY

In this paper, a new calibration method for road capacities in urban networks is presented. The method relies on partial least squares (PLS) regression, which combines calibration and dimensionality reduction capabilities. A sampling strategy is implemented to further improve the calibration efficiency and accuracy. Moreover, influences of different parameters on calibration results are investigated. This method is demonstrated to be feasible and efficient in an urban road network (Stockholm, Sweden).

Keywords: dimensionality reduction, partial least squares regression, road capacity calibration.

1. INTRODUCTION

Traffic simulation models are important for describing urban congestion and estimating travel times. The quality of the estimates provided by traffic simulation models depends critically on the way in which network capacity is represented. The present work considers network models in which road capacity is exogenously represented by attaching a capacity parameter to every homogeneous road segment, with this parameter specifying the maximum number vehicles that could pass this road segment in one time unit. Thus, calibration of the capacity parameters is essential in order for the model to represent an actual traffic network.

Research problem and potential issues

The calibration problem in our case is presented as following: Given a road network simulation model that is parameterized with one capacity value per homogeneous road segment, and given link flow measurements, the objective is to calibrate the capacity parameters on all links in the network.

The capacity calibration problem is solved through a minimization problem in which the squared distance between the observed output (flow) vector and the predicted one obtained from the estimated capacities needs to be minimized.

A frequently-encountered calibration method includes numerical estimation of the full Jacobian matrix (sensitivity analysis) (Cascetta, 1984), aiming at finding the local linear approximation between the input and output variables. The issue is that high dimensionality (number of links) in networks introduces great complicity. This classical calibration method, consisting of series of iterations in which estimation of local Jacobian matirx is computed, faces computational efficiency problems.

One of the most popular methods for improving efficiency of estimating the full Jacobian matrix is Simultaneous Perturbation Stochastic Approximation (SPSA). It simplifies multivariate optimization problems by approximating gradient with only a small number of measurements per iteration in which all variables are varying randomly in a proper way (Spall, 1992). Nowadays, some works investigate modifications and variations of SPSA method to achieve higher efficiency or better robustness, such as weighted-SPSA (Lu, Xu, Antoniou, & Ben-Akiva, 2015) (Antoniou, Azevedo, Lu, Pereira, & Ben-Akiva, 2015) and cluster-wise SPSA (Tympakianaki, Koutsopoulos, & Jenelius, 2015).

Contribution

The present work aims to overcome the challenge of estimating full Jacobian matrix by reducing problem dimensionality simultaneously with estimating a relatively small Jacobian matrix of link flows (measurements) with respect to capacities (decision variables). The purpose is similar to SPSA in terms of improving computational efficiency but from a different perspective. For this, the partial least squares (PLS) regression method is applied (Geladi & Kowalski, 1986). PLS regression has been broadly used in chemometrics, e.g. (Godoy, Vega, & Marchetti, 2014), but has to the best of our knowledge not yet been accessed in the area of traffic model calibration.

Compared to principal components analysis (PCA), which has been used in traffic calibration problems, PLS regression has following advantages:

1. Since loading vectors (approximately can be regarded as principal components) are considered pairwisely, only the diagonal elements in the regression matrix needs to be calculated. This will be illustrated in a more detailed way in latter sections.

2. In order to achieve the same level of approximation, less components are needed in partial least squares regression (PLSR) than in principal components regression (PCR). (Helland, 1988).

Since dimensionality reduction is introduced in PLS, it is potentially more suitable for the cases in which the number of output measurements is smaller than that of input variables needing to be estimated when compared with other calibration methods without dimensionality reduction.

2. METHODOLOGY

Figure 1 outlines the calibration approach in our work. The method is iterative, with each iteration comprising the following steps in Figure 1.

1. **Generation and evaluation of trial points.** Variations of a current capacity solution vector are generated. For each variation, a simulation run is conducted and the resulting simulated network flows are obtained. These trial points are used as training data to find local linear approximation in the following step.
2. **Joint linear approximation and dimensionality reduction.** The *Partial Least Squares (PLS)* algorithm is adopted to estimate a low-dimensional linear regression model approximating the high-dimensional simulator mapping of link capacities onto network flows to achieve computational efficiency improvement.
3. **Capacity update.** An improved capacity estimate is obtained by solving a quadratic optimization problem based on the most recently obtained linear simulator approximation.

The remainder of this section details these steps, with a particular focus on the PLS regression

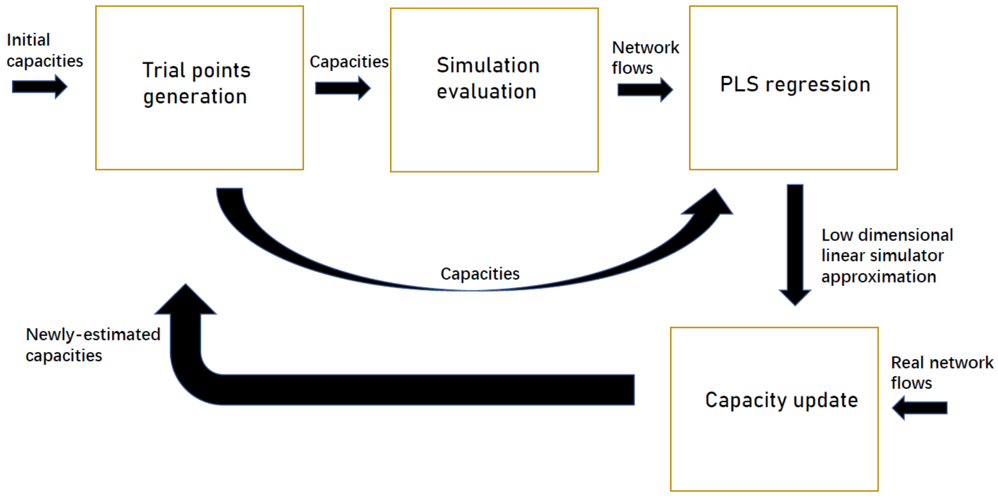

Figure 1: Flow chart of the proposed method

method. Table 1 lists the mainly used symbols.

Table 1: Main symbols

symbol	meaning
m	dimension of input variables (capacities) and output variables (flows), i.e. number of links
n	number of trial points used in PLS regression
δ	variation coefficient in Step **Generation and evaluation of trial points**
$\mathbf{X}$	mean-centered matrix of input trial points, size: $n \times m$
$\mathbf{Y}$	mean-centered matrix of output trial points, size: $n \times m$
$\mathbf{X}_i$	the mean-centered input matrix in i-th outer loop in PLS regression, size: $n \times m$
$\mathbf{Y}_i$	the mean-centered output matrix in i-th outer loop in PLS regression, size: $n \times m$
$\mathbf{p}_i$	i-th loading vector in input space, size: $m \times 1$
$\mathbf{t}_i$	i-th score vector in input space, size: $n \times 1$
$\mathbf{q}_i$	i-th loading vector in output space, size: $m \times 1$
$\mathbf{u}_i$	i-th score vector in output space, size: $n \times 1$
a	the number of loading vector pairs used, i.e. dimension after dimensionality reduction operation
b_i	the regression coefficient in i-th outer loop (of $\mathbf{u}_i$ on $\mathbf{t}_i$) in PLS regression

Generation and evaluation of trial points

A network with m links and a current best capacity vector estimate is given. A trial capacity vector is obtained by independently and uniformly varying each capacity value in a range of $\pm\delta$ times its current value. Overall n trial capacity vectors are generated. Each capacity vector is then evaluated in a road traffic simulator and corresponding network flows are obtained.

Omitting an iteration index, the resulting n (capacity, flow) vector tuples are denoted by $(\mathbf{x}_r, \mathbf{y}_r)$, with r the *replication index* within the given algorithm iteration. Both capacity and flow vectors are mean-centered in an immediate post-processing step.

Joint linear approximation and dimensionality reduction

Assume a set of mean-centered input/output (in the present application, capacity/flow) data tuples $(\mathbf{x}_r, \mathbf{y}_r)$, $r = 1 \ldots n$, to be given, with both input and output being m-dimensional vectors. We are interested in estimating a linear model relationship between independent (input) variables $\mathbf{x}$ and dependent (output) variables $\mathbf{y}$. We further assume that n, the number of observations used in PLS regression, is relatively small compared to m, the dimensionality of the model's in- and output space.

To arrive at an identifiable model, we reduce the dimensionality of both in- and output space from dimension m to dimension a. The low-dimensional representation of the input space is spanned by *loading vectors* $\mathbf{p}_i$, $i = 1 \ldots a$, and every input vector $\mathbf{x}_r$ is represented as a linear combination of these loading vectors:

$$\mathbf{x}_r = \sum_{i=1}^{a} t_{ri} \mathbf{p}_i + \mathbf{f}_r \tag{1}$$

where the *score* t_{ri} represents the contribution of the i-th loading vector to $\mathbf{x}_r$ and $\mathbf{f}_r$ absorbs the approximation error in input space. Symmetrically, the output space is spanned by loading vectors $\mathbf{q}_i$, $i = 1 \ldots a$:

$$\mathbf{y}_r = \sum_{i=1}^{a} u_{ri} \mathbf{q}_i + \mathbf{g}_r \tag{2}$$

with u_{ri} and $\mathbf{g}_r$ being specified symmetrically to t_{ri} and $\mathbf{f}_r$. Given the loading vectors, the input and output vectors are hence encoded by the *score vectors*

$$\begin{aligned} \mathbf{t}_i &= (t_{1i} \ldots t_{ni})^T & (3) \\ \mathbf{u}_i &= (u_{1i} \ldots u_{ni})^T & (4) \end{aligned}$$

with $i = 1 \ldots a$. Instead of estimating a regression model coupling $\mathbf{x}$ and $\mathbf{y}$, we estimate one regression model for each $i = 1 \ldots a$ by ordinary least squares. For the i-th model, its single regression coefficient b_i is given by

$$b_i = \arg\min_{b \in \mathbf{R}} \|\mathbf{u}_i - b\mathbf{t}_i\|^2 . \tag{5}$$

For given loading vectors, this model can be used for prediction by (i) representing an input vector $\mathbf{x}_{n+1}$ in terms of its input scores $t_{(n+1)i}$, $i = 1 \ldots a$, (ii) using the b_i regression coefficients of (5) to compute the corresponding output scores $u_{(n+1)i} = b_i t_{(n+1)i}$, $i = 1 \ldots a$, and (iii) approximating the output signal from (2) with a zero residual vector $\mathbf{g}$.

The PLS algorithm estimates simultaneously the loading vectors $\mathbf{p}_i$, $\mathbf{q}_i$, the corresponding score representation of a set of data tuples $(\mathbf{x}_r, \mathbf{y}_r)$, $r = 1 \ldots n$, and the low-dimensional regression models (5). To simplify notation, in- and ouput vectors are stacked in the following matrices:

$$\begin{aligned} \mathbf{X} &= (\mathbf{x}_1 \ldots \mathbf{x}_n)^T & (6) \\ \mathbf{Y} &= (\mathbf{y}_1 \ldots \mathbf{y}_n)^T . & (7) \end{aligned}$$

Based on this, the PLS regression algorithm of (Geladi & Kowalski, 1986) can be given in Algorithm 1.

From an alternative approach, the PLS regression reduces problem dimensionality by enforcing an OLS solution that is located in a low-dimensional subspace that is constructed along directions

Algorithm 1 PLS regression

Notation: "←" means a variable assignment from right to left.

```
1. Initialize:
    (a) $\mathbf{X}_1 \leftarrow \mathbf{X}$
    (b) $\mathbf{Y}_1 \leftarrow \mathbf{Y}$.
2. For $i = 1 \ldots a$:
    (a) Set $\mathbf{u}_i$ to an arbitrary column of $\mathbf{Y}_i$.
    (b) "X block":
        i. $\mathbf{w}_i \leftarrow \mathbf{X}_i\mathbf{u}_i / \|\mathbf{u}_i\|^2$
        ii. $\mathbf{w}_i \leftarrow \mathbf{w}_i / \|\mathbf{w}_i\|$
        iii. $\mathbf{t}_i \leftarrow \mathbf{X}_i\mathbf{w}_i$
    (c) "Y block":
        i. $\mathbf{q}_i \leftarrow \mathbf{Y}_i\mathbf{t}_i / \|\mathbf{t}_i\|^2$
        ii. $\mathbf{q}_i \leftarrow \mathbf{q}_i / \|\mathbf{q}_i\|$
        iii. $\mathbf{u}_i \leftarrow \mathbf{Y}_i\mathbf{q}_i$
    (d) Update of loadings and scores:
        i. $\mathbf{p}_i \leftarrow \mathbf{X}_i\mathbf{t}_i / \|\mathbf{t}_i\|^2$
        ii. $\mathbf{t}_i \leftarrow \mathbf{t}_i \|\mathbf{p}_i\|$
        iii. $\mathbf{p}_i \leftarrow \mathbf{p}_i \|\mathbf{p}_i\|$
    (e) Regression: $b_i = \mathbf{u}_i^T \mathbf{t}_i / \|\mathbf{t}_i\|^2$
    (f) Calculation of residuals:
        i. $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\mathbf{p}_i^T$
        ii. $\mathbf{Y}_{i+1} = \mathbf{Y}_i - b_i\mathbf{t}_i\mathbf{q}_i^T$
```

of large variability in the explanatory variables (Frank & Friedman, 1993). Since the previous m input variables are 'compressed' into a new variables through PLS regression. It no longer requires all of the m output variables should have a observed value in order to get a unique solution in the optimization problem (instead, the number of observed output variables just needs to be no less than a).

Capacity update

Denote the (again, mean-centered) real network flow observations as $\tilde{\mathbf{y}}' = (\tilde{y}_j)'$. The updated (and mean-centered) capacities $\hat{\mathbf{x}}'$ are then obtained in two steps. First, the following optimization problem is solved:

$$\min_{\mathbf{s}=(s_i)} \sum_{j=1}^{m} \left(\tilde{y}'_j - \sum_{i=1}^{a} b_i s_i q_{ij} \right)^2 \tag{8}$$

where q_{ij} is the jth element in the loading vector $\mathbf{q}_i$. The solution $\mathbf{s}$ of this optimization problem contain the scores on all the loading vectors $\mathbf{p}_i$, $i = 1 \dots a$ and $\mathbf{s}$ has a dimension of a. Next, the corresponding main-centered capacity vector $\hat{\mathbf{x}}'$ is constructed according to

$$\hat{\mathbf{x}}' = \sum_{i=1}^{a} s_i \mathbf{p}_i. \tag{9}$$

Then the estimated capacity estimation $\hat{\mathbf{x}}$ (without mean-centering) can be recovered. To avoid oscillations, the currently best capacity estimate is updated by computing a convex combination of the previous estimate and $\hat{\mathbf{x}}$, with the weight on $\hat{\mathbf{x}}$ being specified further below.

Weight Settings

In capacity update step, a weight α_k (where k is the current iteration number) on $\hat{\mathbf{x}}$ needs to be set in order to guarantee that the optimization point can be obtained and avoid oscillations. In generation of trial points, there also exists a weight β_k, which makes the range coefficient of variation δ change after each iteration (i.e. $\delta = \delta_0 \beta_k$, where δ_0 is the fixed initial variation coefficient). α_k and β_k needs to satisfy the following conditions (Spall, 1992):

$$\alpha_k > 0, \;\; \beta_k > 0, \;\; \forall\, k \tag{10}$$

$$\alpha_k \to 0, \;\; \beta_k \to 0, \;\; as\; k \to \infty \tag{11}$$

$$\sum_{k=1}^{\infty} \alpha_k = \infty \tag{12}$$

$$\sum_{k=1}^{\infty} \left(\frac{\alpha_k}{\beta_k}\right)^2 < \infty. \tag{13}$$

Based on these conditions, $\alpha_k = \frac{1}{k}$ and $\beta_k = (\frac{1}{k})^{\frac{1}{3}}$ are set in this work. They remain unchanged in all the following experiments. In other words, the final estimated capacities $\hat{\mathbf{x}}$ in kth iteration is set according to

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_p + \frac{1}{k}(\hat{\mathbf{x}} - \hat{\mathbf{x}}_p) \tag{14}$$

where $\hat{\mathbf{x}}_p$ is the final estimated capacities in the previous iteration and "←" means a variable assignment from right to left.

3. RESULTS AND DISCUSSION

Experiment 1: Default parameter setting with a sampling strategy

Based on the method mentioned in the previous section, a sampling strategy is implemented which aims at further improving the efficiency of calibration. We do not create a new set of simulator replications in every iteration but also recycle all replications from earlier iterations. In the first iteration, 101 trial points are generated in the simulator and used in the PLS regression. From the second iteration, only 11 replications are implemented and the newly generated trial points are added to the complete pool of trial points from all the previous iterations. Correspondingly, in the PLS regression from 2nd iteration, 101 trial points that are closest to the current capacity estimate are used.

Then we formulate a basic experiment (Experiment 1) with parameter settings shown in 2:

Table 2: Default parameter values

parameter	default value
initial variation coefficient δ_0	0.1
generated trial point number in 1st iteration N_1	101
generated trial point number in k-th iteration ($k \neq 1$) N_2	11
number of trial points used in PLS regression N_3	101
number of loading vector pairs a	20

In order to evaluate the performance of the method, a test network of Stockholm with 22547 links is used. A synthetic real capacity vector is generated by randomly varying the given network parameters. The MATSim network assignment package is used to compute network flows; for simplicity the following experiments consider a fixed demand exogenously given route choice.

A real capacity vector $\tilde{\mathbf{x}}$ is randomly (but in a reasonable range) generated and the corresponding network flows $\tilde{\mathbf{y}}$ are simulated.

The performance of different configurations of the proposed method is evaluated in terms of the square error between real flows $\tilde{\mathbf{y}}$ and the estimated ones obtained from mapping estimated capacities $\hat{\mathbf{x}}$ in the simulator:

$$e_{flow}^2 = ||\tilde{\mathbf{y}} - f(\hat{\mathbf{x}})||^2 \tag{15}$$

where function f represents the map from capacities to flows in MATSim simulator. It should be noted that in our simulation, **both $\tilde{\mathbf{y}}$ and the mapping function f lead to** 2% **of the real flows and estimated flows, respectively** in order to reduce the computation time. This setting does not influence feasibility of the whole method but it should be kept in mind when reading the result graphs.

An alternative approach is to evaluate the performance in terms of the square error between real (yet to the calibration method unknown) capacities $\tilde{\mathbf{x}}$ and their estimated counterpart $\hat{\mathbf{x}}$:

$$e_{capacity}^2 = ||\tilde{\mathbf{x}} - \hat{\mathbf{x}}||^2. \tag{16}$$

We can also investigate the result in a more detailed way: Relationship between the relative calibration error e_i for a given link i and the congestion level of it $cong_i$ can be plotted for all the links. Difference d_i between initial relative error $e_{ini,i}$ and e_i can also be calculated to see if calibration makes the estimated capacity value of link i move towards the right direction. The mathematical expression of theses parameters are shown below:

$$cong_i = \frac{\tilde{y}_i}{\tilde{x}_i} \tag{17}$$

$$e_i = \frac{|\tilde{x}_i - \hat{x}_i|}{\tilde{x}_i} \tag{18}$$

$$e_{int,i} = \frac{|\tilde{x}_i - x_{int,i}|}{\tilde{x}_i} \tag{19}$$

$$d_i = e_i - e_{ini,i} \tag{20}$$

where subscript i represent link index and $x_{int,i}$ is the initial guess of capacity value of link i.

One intuition is that calibration of those links which are relatively more congested should be more precise. A variable of congestion level $cong$ can be defined and the observed data is $cong_i$ for each link i. Similarly, variables e and d can be defined with observations e_i and d_i for all links, respectively. Then two singular linear regression models can be constructed to test the conjecture:

$$e = \hat{\gamma}_1 cong + \hat{\gamma}_0 + \varepsilon \tag{21}$$

$$d = \hat{\mu}_1 cong + \hat{\mu}_0 + \varepsilon' \tag{22}$$

where ε and ε' are error terms.

Figure 2 shows the capacity error $e^2_{capacity}$ and flow error e^2_{flow} versus iteration numbers, where the error at iteration number 0 represents the error between initial guess and real values of capacities and flows, respectively. From these graphs it can be seen that both errors goes down as iteration number increases, which indicates our method works. Given that this simulation requires about 8 hours of computing time on a PC with a RAM of 16GB and a clock speed of 2400 GHz, we observe that the method is computationally feasible.

In this work, a random seed is set in MATSim in determining the generation of input trial data and it has been found that the flow error is always decreasing for different random seeds but the capacity error does not always reduce. One reasonable explanation is that the randomness of trial points in the input space leads to variation in capacity estimation, but different estimations map to similar values in output space due the fact that the functional relationship between capacities and flows are "many to one".

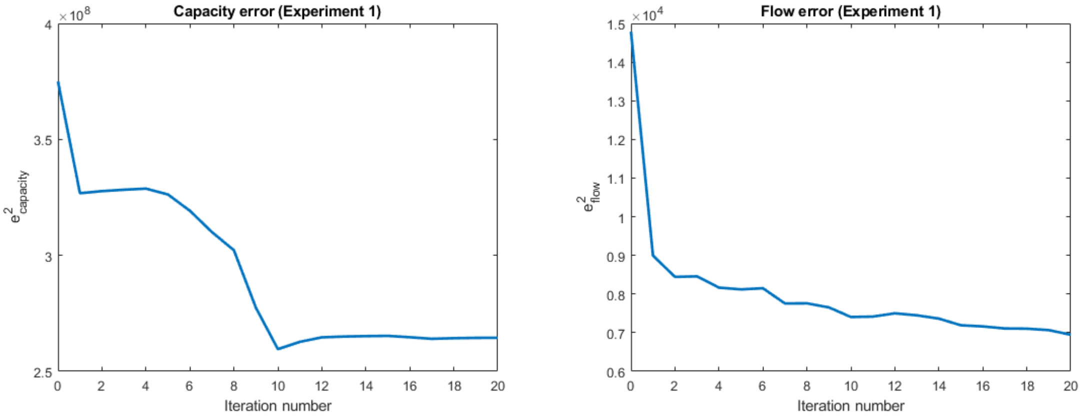

Figure 2: Capacity error $e^2_{capacity}$ and flow error e^2_{flow} versus iteration numbers in Experiment 1

Figure 3 shows the relative error e (vertical axis) and calibration difference d (color mapping) versus congestion level $cong$ (horizontal axis), respectively. The congestion level is above 0.02 occasionally due to the numerical approximation, which does not influence the performance of the whole method. The estimation of $\hat{\gamma}_1$ and $\hat{\mu}_1$ are shown in Table 3 and no conclusion can be drawn that more congested links have better calibration results.

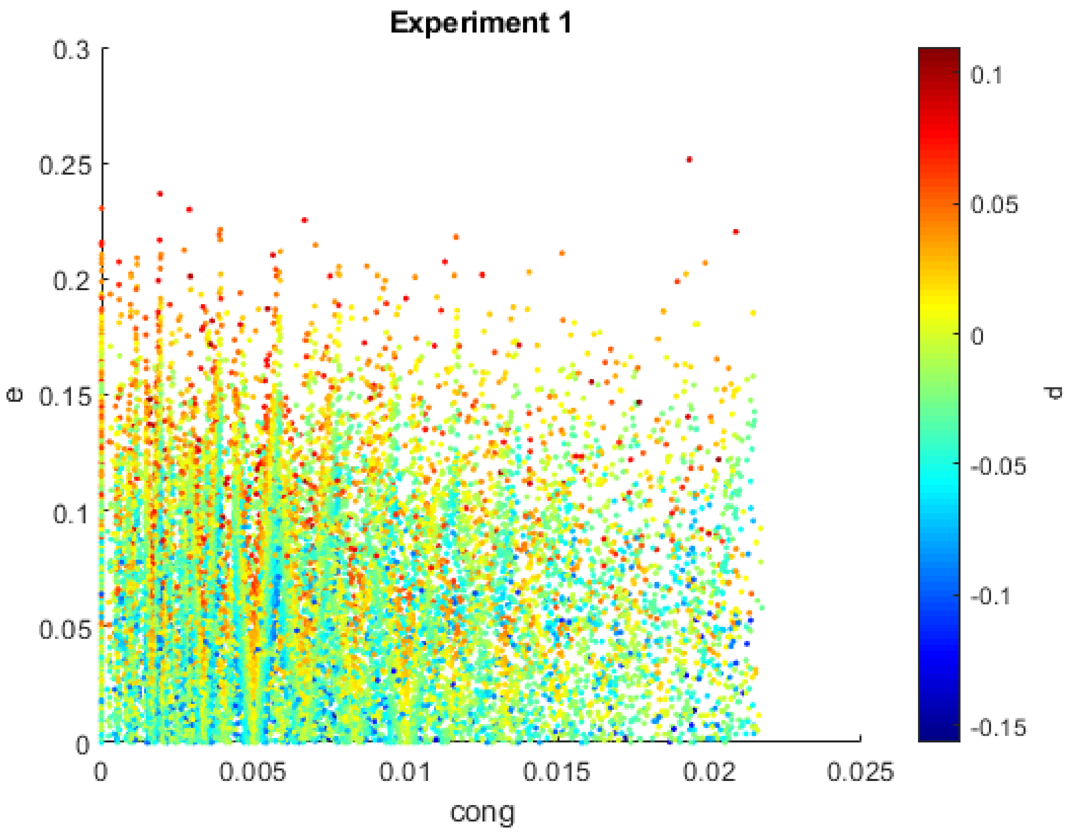

Figure 3: Relative error e and calibration difference d versus congestion level $cong$ in Experiment 1

Table 3: Estimation of $\hat{\gamma}_1$ and $\hat{\mu}_1$ in Experiment 1

Estimator	$\hat{\gamma}_1$	**Standard deviation of $\hat{\gamma}_1$**	$\hat{\mu}_1$	**Standard deviation of $\hat{\mu}_1$**
	0.037	0.054	-0.058	0.045

Experiment 2 and 3: Influence of different parameters

Based on Experiment 1, each of the following experiments aims at investigating the influence of one specific parameter:

1. **Experiment 2:** Influence of initial variation coefficient δ_0 in trial points generation.
2. **Experiment 3:** Influence of number of loading vector pairs a in PLS regression.

Figure 4 shows the flow error e^2_{flow} versus iteration numbers in Experiment 2. From the graph, $\delta_0 = 0.15$ has the best calibration result. It indicates that the ideal δ_0 value should be neither too large (approximation will be inaccurate) nor too small (noise influence will be great).

Figure 5 shows the flow error for different numbers of loading vector pairs (i.e. different a values) used in PLS regression. An increased number of loading vector pairs indicates better result. But

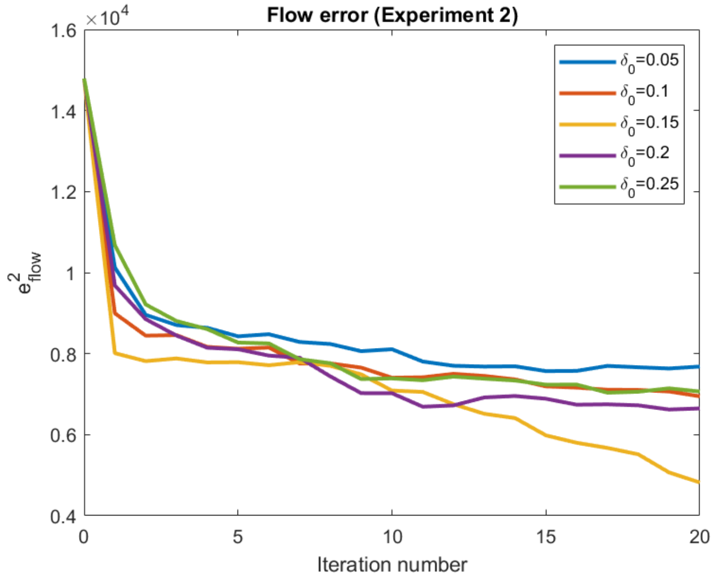

Figure 4: Flow error e^2_{flow} versus iteration numbers in Experiment 2

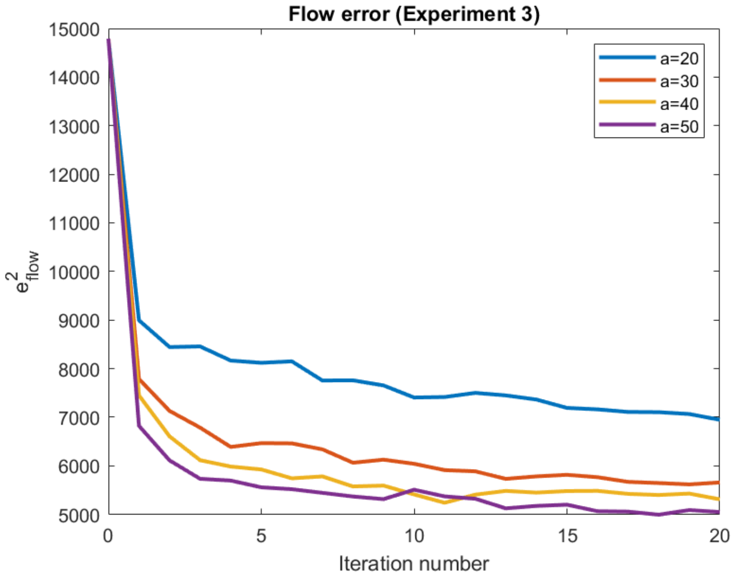

Figure 5: Flow error e^2_{flow} versus iteration numbers in Experiment 3

it should be noticed that the introduction of more loading vectors leads to increase of the running time.

For Experiment 2 and 3 the estimation of μ_1 is investigated for the best calibration case in each experiment ($\delta_0 = 0.15$ for Experiment 2 and $a = 50$ for experiment 3). Table 4 shows the estimation of μ_1 for these cases. It can be observed that $\hat{\mu}_1$ has a relatively smaller (more negative) estimated value and smaller deviation when compared with that from Experiment 1, which indicates that for these cases, more congested links have a better calibration result.

Table 4: Estimation of γ_1 and μ_1 in Experiment 2 ($\delta_0 = 0.15$) and 3 ($a = 50$)

Estimator	$\hat{\mu}_1$	**Standard deviation of $\hat{\mu}_1$**
Experiment 2 ($\delta_0 = 0.15$)	-0.087	0.048
Experiment 3 ($a = 50$)	-0.097	0.059

4. CONCLUSIONS

In this work, we present a method for urban network road capacity calibration in which PLS regression is introduced to achieve dimensionality reduction. By applying the method to a test urban network - Stockholm network, we find that this method is feasible and efficient. Moreover, it does not require other constraints in the optimization part (such as non-negativity).

It is a promising method since it can not only be used in capacity calibration but also other calibration problems (such as O-D calibration) which share a very similar structure. Even more generally, it has the potential to be applied on most of high-dimensional inverse problems.

The presented results are preliminary and specific to one considered road network. Important further investigations comprise the effect of incomplete network flow measurements, unfixed route choices and other networks.

ACKNOWLEDGMENT

This research has been funded by the Swedish Transport Administration (TRV 2018/134731).

CONTRIBUTION STATEMENT

Guang Wei: Writing, design and coding of experiments and conceptual contributions;
Joakim Ekström: Text editing and supervision;
Gunnar Flötteröd: Supervision

REFERENCES

Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., & Ben-Akiva, M. (2015). W–spsa in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Procedia*, *7*, 233-253. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352146515000812 (21st International Symposium on Transportation and Traffic Theory Kobe, Japan, 5-7 August, 2015) doi: https://doi.org/10.1016/j.trpro.2015.06.013

Cascetta, E. (1984). Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. *Transportation Research Part B: Methodological*, *18*(4), 289 - 299. Retrieved from http://www.sciencedirect.com/science/article/pii/0191261584900122 doi: https://doi.org/10.1016/0191-2615(84)90012-2

Frank, I. E., & Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, *35*(2), 109–135. Retrieved from http://www.jstor.org/

stable/1269656

Geladi, P., & Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, *185*, 1 - 17. Retrieved from http://www.sciencedirect.com/science/article/pii/0003267086800289 doi: https://doi.org/10.1016/0003-2670(86)80028-9

Godoy, J. L., Vega, J. R., & Marchetti, J. L. (2014). Relationships between pca and pls-regression. *Chemometrics and Intelligent Laboratory Systems*, *130*, 182-191. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169743913002189 doi: https://doi.org/10.1016/j.chemolab.2013.11.008

Helland, I. S. (1988). On the structure of partial least squares regression. *Communications in Statistics - Simulation and Computation*, *17*(2), 581-607. Retrieved from https://doi.org/10.1080/03610918808812681 doi: 10.1080/03610918808812681

Lu, L., Xu, Y., Antoniou, C., & Ben-Akiva, M. (2015). An enhanced spsa algorithm for the calibration of dynamic traffic assignment models. *Transportation Research Part C: Emerging Technologies*, *51*, 149 - 166. Retrieved from http://www.sciencedirect.com/science/article/pii/S0968090X14003295 doi: https://doi.org/10.1016/j.trc.2014.11.006

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*(3), 332-341. doi: 10.1109/9.119632

Tympakianaki, A., Koutsopoulos, H., & Jenelius, E. (2015, 02). c-spsa: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin-destination matrix estimation. *Transportation Research Part C Emerging Technologies*, *55*. doi: 10.1016/j.trc.2015.01.016