A Multiplicative Conjugate Gradient method for the O–D Adjustment Matrix

M. Victoria Chávez–Hernández
Autonomous Metropolitan University - Iztapalapa, Mathematics Department.
vicky.fis@gmail.com

L. Héctor Juárez–Valencia
Autonomous Metropolitan University - Iztapalapa, Mathematics Department.
hect@xanum.uam.mx

June 21, 2014

Abstract

Estimating an origin–destination matrix (O–D matrix) from previously known data has been very important in public transportation systems. Many different models have been used, some of them take into account marginal totals of the O–D matrix, and many others consider observed link volumes at strategic links on the network. In practice, it is easier to obtain link volumes than marginal totals. In the past a steepest descent model was applied successfully to large scale networks. This model is mathematically formulated as a convex minimization problem that includes the difference between the observed and the assigned volumes. In this paper we propose a model which considers the difference between the obsolete O–D matrix and the adjusted one, and we suggest to penalize the difference the volumes. Also we propose to use a conjugate gradient method with a multiplicative iteration formula, where the descent direction is found in a natural way to avoid evaluation of Hessians at iterations.

Keywords: O–D matrix adjustment, demand models, transit assignment, public transportation, conjugate gradient method, bilevel programming.

1 Introduction

In modern societies, public transport is becoming more relevant, especially in large cities where it is extremely important to have a good transportation planning in order to provide an efficient service. This requires studying and understanding the operation of the transportation system using the right tools. In particular, mathematical models for transit assignment are very useful to help understand how users travel from their different origins to their diverse destinations. To have a good transportation planning, results must be as realistic as possible, and for this purpose it is necessary to collect field data. These data may be obtained based on surveys and other complex and expensive studies, but unfortunately they are useful only for a short limited time due to growth in demand and change in infrastructure around big cities. To avoid making new comprehensive studies there
are some techniques that allow approximations to the most recent data using previously known data and only a small, but significant, amount of new data.

One of the most important elements on the transportation planning process is the O–D matrix, which represents the flow between each origin to each destination node of the transportation network. Generating or updating these data is not easy, since people travel to satisfy many diverse needs, and it clearly depends on many factors, as day and time. One way to count flow volumes may be using ticket counting machines at bus stations, or using sensors; but this information, if it is available, may not be entirely realistic. Another more efficient and less expensive way to estimate O-D matrices is from volume counts at some specific important links of the network. A large amount of research has been done in this direction, see for instance references [2], [10], [11], [12], [8], [3]. One of these methods is the generalized least squares, which consists on minimize both the square of the difference between the previously known and the unknown adjusted matrices, and the square of the difference between the observed and the assigned volumes.

In 1990, Spiess [11] proposed a multiplicative steepest descent method to find O–D matrix from a least squares model that minimizes the square of the difference of the observed and assigned volumes. This multiplicative algorithm keeps the structure of the “a priori” matrix and it can handle large scale networks. Least squares models has been studied and extended over the years; for instance, in [2] and [10] the objective function also incorporates the difference between the obsolete matrix and the unknown adjusted matrix. Schemes like this are part of the well known bilevel programs, [8], [3]. An important issue on these bilevel programs is finding a good iterative algorithm that solves efficiently the corresponding model for large networks. A more general overview of O–D matrix estimation can be found in reference [1]. In this work we introduce a multiplicative conjugated gradient method to find an O–D matrix from a generalized least–squares model, but where the difference of volumes (measured and assigned) is penalized. This work aims to explore the conjugate gradient algorithm to enhance the solution performance of the least squares model.

2 The O–D matrix adjustment problem

In order to state the mathematical formulation of the problem, the following notation is introduced. The sets of origin and destination nodes in the transit network are denoted by $P$ and $Q$, respectively; and, $A$ is the set of links where counts are available. A generic origin node is denoted by $p \in P$, a destination node is denoted by $q \in Q$ and the links are denoted by $a \in A$. The set of O–D pairs is denoted by $PQ$ and any of the O–D pairs will be denoted by $pq = (p, q)$, $p \in P$ and $q \in Q$. The “a priori” known demand matrix is denoted by $G = \{G_{pq}\}$ and the adjusted matrix is denoted by $g = \{g_{pq}\}$. Spiess’ model, [11], reads as follows:

$$\begin{align*}
\min_{\mathbf{g}} Z(\mathbf{g}) &= \frac{1}{2} \sum_{a \in A} (v(\mathbf{g})_a - V_a)^2 \\
v(\mathbf{g})_a &= assign(\mathbf{g})
\end{align*}$$

where $v(\mathbf{g})_a = assign(\mathbf{g})$, indicates the volumes resulting from an assignment of the demand matrix $\mathbf{g}$. This assignment procedure must correspond to a convex optimization problem and is understood as an equilibrium assignment to ensure the convexity of the model, see [4], [7] and [5]. This model can be extended, adding the distance between the “a priori” matrix and the adjusted matrix,
\( \frac{1}{2} \sum_{pq \in PQ} (g_{pq} - G_{pq})^2 \), see [2], [10], [8], [3]. Here, we depart from the following model:

\[
\min \ Z(g) = \frac{1}{2} \sum_{pq \in PQ} (g_{pq} - G_{pq})^2 \\
\text{Subject to: } v(g)_a \approx V_a, \text{ and } g_{pq} \geq 0 \\
v(g)_a = \text{assign}(g)
\]  

We can simplify the formulated problem by adding, to the objective function (3), a penalized sum of the squared differences between the observed and the assigned volumes, obtaining the following convex minimization problem:

\[
\min \ Z(g) = \frac{1}{2} \sum_{pq \in PQ} (g_{pq} - G_{pq})^2 + k \sum_{a \in A} (v(g)_a - V_a)^2, \quad k > 0
\]

\[
\text{Subject to: } v(g) = \text{assign}(g), \text{ and } g_{pq} \geq 0
\]  

where \( k \) is the penalty coefficient. The resulting effect of this formulation is that, the greater the value of \( k \) the better restriction (7) is satisfied, as it is demonstrated in the numerical results. It is possible that the measured link volumes are more reliable or accurate on some links. However, different penalty values \( k_a \) may be used for each \( a \in A \), depending on the relative confidence of the correspondent measured volumes \( V_a \). To keep the discussion as simple as possible, in this paper a constant value of \( k \) is considered for all links.

### 3 Conjugate Gradient Method

The method of steepest descent has been successfully used to solve the O–D matrix adjustment problem (see [10] and [11], for instance), however it is well known that in practice the method of steepest descent can be inefficient because of the zigzag phenomenon, which occurs mainly with ill-conditioned problems, requiring a large number of iterations to approach the optimum. In those cases, the Newton’s method is much more efficient; but it is very expensive, specially for large scale problems, since it needs the evaluation of Hessians and the solution of a linear algebraic equation at each iteration. An intermediate iterative descent method with a comparable computational cost to the method of steepest descent, and for which neither of those disadvantages occur, is the conjugate gradient method.

In the simplest case the conjugate gradient method can be formulated as follows:

\[
g_{pq}^{l+1} = \begin{cases} 
G_{pq} & \text{for } l = 0, \\
g_{pq}^l - \lambda^l d_{pq}^l & \text{for } l = 1, 2, ..., L,
\end{cases}
\]

where \( d_{pq}^l \) is a conjugate direction at iteration \( l \) and \( \lambda^l \) is the length step that minimizes de objective function along that direction. In an actual planning context, we expect the resulting matrix to resemble as closely as possible the initial matrix, since it contains important structural information on the origin–destination movements. Following this idea, Spiess proposed a multiplicative iteration formula where a change in demand is proportional to the demand in the initial matrix and, in
particular, zeros are preserved by the process, [11]. Applying this idea to the conjugate gradient method, the iterative formula can be expressed as:

$$g_{pq}^{l+1} = \begin{cases} G_{pq} & \text{for } l = 0, \\ g_{pq}^l (1 - \lambda^l d_{pq}^l) & \text{for } l = 1, 2, \ldots, L. \end{cases}$$

(9)

In the conjugate gradient algorithm the conjugate direction at each iteration is generated as a linear combination of the previous conjugate direction and the current gradient

$$d_{pq}^{l+1} = g_{pq}^l \left[ -\frac{\partial Z(g^{l+1})}{\partial g_{pq}} \right] + \beta^l d_{pq}^l$$

(10)

where $\beta^l$ is computed to ensure that the two directions in (10) are conjugate to each other. The gradient can be computed from (6) using the chain rule

$$\frac{\partial Z(g^{l+1})}{\partial g_{pq}} = (g_{pq}^{l+1} - G_{pq}) + k \sum_{a \in A} \left( v_a(g^{l+1}) - V_a \right) \frac{\partial v_a(g^{l+1})}{\partial g_{pq}}, \quad pq \in PQ.$$  

(11)

where we still need to compute the last derivative. To do this, the link volumes can be expressed as:

$$v_a(g^{l+1}) = \sum_{pq \in PQ} \sum_{s \in S_{pq}} \delta_{as} h_s, \quad a \in A \quad \delta_{as} := \begin{cases} 0 & \text{if } a \notin s \\ 1 & \text{if } a \in s \end{cases}$$

(12)

where $S_{pq}$ is the set of used paths in the network to go from $p \in P$ to $q \in Q$, and $h_s$ denotes the total flow along one path $s \in S_{pq}$. Equation (12) can be rewritten in terms of the path probabilities $\pi_s^{l+1} = h_s/g_{pq}^{l+1}, \ s \in S_{pq}, \ pq \in PQ$, obtaining

$$v_a(g^{l+1}) = \sum_{pq \in PQ} g_{pq}^{l+1} \sum_{s \in S_{pq}} \delta_{as} \pi_s^{l+1}, \quad a \in A.$$  

(13)

Assuming that $\pi_s^{l+1} \approx \pi_s^l$, we get

$$\frac{\partial v_a}{\partial g_{pq}} (g^{l+1}) = \sum_{s \in S_{pq}} \delta_{as} \pi_s^l, \quad a \in A, \ pq \in PQ.$$  

(14)

Therefore,

$$\frac{\partial Z(g^{l+1})}{\partial g_{pq}} = (g_{pq}^{l+1} - G_{pq}) + k \sum_{s \in S_{pq}} \pi_s^l \sum_{a \in A} \delta_{as} \left( v_a(g^{l+1}) - V_a \right), \quad pq \in PQ.$$  

(15)

Note. The assumption $\pi_s^{l+1} \approx \pi_s^l$ is very reasonable, specially when the sequence $\{g^l\}$ is close to the optimum. It not only simplifies the computation of the gradient $\nabla Z(g)$, but also gives a “linear behavior” to $v_a(g)$, since

$$v_a(g + \lambda d) \approx v_a(g) + \lambda v(d)$$
if $||d||$ is small, with $d = \{d_{pq}\}$. We still need to provide the value for the step length $\lambda^l$ in (9). The optimal step length $\lambda^l$ can be found as the minimum point of the one-dimensional sub problem

$$
\phi(\lambda) = Z(g + \lambda d) \approx \frac{1}{2} \sum_{pq \in PQ} (g_{pq} + \lambda d_{pq} - G_{pq})^2 + \frac{k}{2} \sum_{a \in A} (v_a(g) + \lambda v_a(d) - V_a)^2
$$

subject to: $\lambda d \leq 1$, $\forall pq \in PQ$, with $g_{pq} > 0$, (17)

which is given by

$$
\lambda^* \approx \frac{\sum_{pq \in PQ} d_{pq}(G_{pq} - g_{pq}) + k \sum_{a \in A} v_a(d) (V_a - v_a(g))}{\sum_{pq \in PQ} d^2_{pq} + k \sum_{a \in A} v_a(d)^2}.
$$

(18)

After obtaining $g^{l+1}$, the new conjugate direction is computed using formula (10), where the value of $\beta^l$ is calculated by the Hestenes–Stiefel formula (see ref. [9]):

$$
\beta^l = \sum_{pq \in PQ} d^l_{pq} \frac{\partial Z(g^{l+1})}{\partial g_{pq}} \left( \frac{\partial Z(g^{l+1})}{\partial g_{pq}} - \frac{\partial Z(g^l)}{\partial g_{pq}} \right)
$$

$$
- \frac{\sum_{pq \in PQ} d^l_{pq} \left( \frac{\partial Z(g^{l+1})}{\partial g_{pq}} - \frac{\partial Z(g^l)}{\partial g_{pq}} \right)}{\sum_{pq \in PQ} d^2_{pq} + k \sum_{a \in A} v_a(d)^2}.
$$

(19)

4 Numerical results

The multiplicative conjugate gradient algorithm was tested with a network from the city of Winnipeg, Canada, based on the standard EMME/4 Winnipeg Demonstration Database [6]. This is a relatively modest network of 154 zones, 906 nodes, 3005 directional links and 4347 transit segments, where we have 136 available counts, i.e. the segments in $\bar{A}$. All the numerical calculations shown in this section were done in a HP–Pavilion dm4 computer with an Intel(R) Core(TM) i5 processor and 3 GB RAM. The transit assignment steps were done with the software package EMME/4 trial version.

We constructed the following scenario: we did an assignment on the Winnipeg network with the exact O-D matrix $g$, which corresponds to the peak hour in the morning. Then, we extracted the obtained synthetic volumes $V_a$, $a \in A$. Next, we generated the “a priori” O-D matrix $G$, perturbing stochastically the exact O-D matrix by 30%. With this information ($G$ and $V_a$, $a \in A$), we can apply the multiplicative steepest descent method (SD) and the multiplicative conjugate gradient method (CG), and compare their performances, when recovering the original matrix $g$.

For the numerical experiments, we first chose the penalty parameter $k = 100$. Figure 1 shows the Winnipeg demand deviations (scatterplot of the exact demand versus the recovered one) obtained with both iterative methods, so that each red point has coordinates $(g_{pq}, G_{pq})$. In this figure $A$ and $B$ are the parameter values of the regression line, thus the adjustment is better for those points that are closer to it. Also, $R^2$ and $RMSE$ are the correlation coefficient and the root–mean–square error, respectively. We observe that both methods provide very similar results, but the method of steepest descent requires 18 more iterations than the conjugate gradient method, as we expected. Figure 2 shows the Winnipeg flow comparison (scatterplot of observed versus assigned volumes, $(V_a, v(g)_a)$; the results obtained with both methods are quite similar and accurate to enforce restriction (4).
It is still possible to reduce the number of conjugate gradient iterations when higher values of the penalty parameter are chosen. Using the values $k = 1000$ and $k = 10000$ the number of iterations is reduced to 54 and 53, respectively, indicating that $k = 10000$ is close to the optimal penalty parameter. So, the multiplicative conjugate gradient method applied to the penalized model is able to reduce the number of iterations from 86 to 53, i.e. about 40%. Figure 3 shows the results with $k = 1000$. 
5 Conclusions

In this work we consider an O–D matrix demand adjustment model, which is based on available volume data on some known links in the network. This model considers the difference between observed and assigned volumes as a constraint, and incorporates those differences to the objective function as penalized quadratic terms, as in (6)–(7). To solve the optimization problem we introduce a multiplicative conjugate gradient method. The performance of this algorithm is compared with the method of steepest descent of Spiess [11]. Both methods yield very similar solutions, but with the advantage that the conjugate gradient algorithm does much less iterations to get the same accuracy. Also, we have observed that the number of iterations can be further reduced by using large values of the penalty parameter in the model.

A potential drawback of our method is that the number of operations at each iteration is greater than in Spiess’ algorithm, due to the calculation of $\beta^l$ in (19). However, it may be possible to find a good preconditioner for the CG algorithm to reduce the number of iterations further. Also, there are additional issues that we need to investigate further, like the convergence properties of the iterative algorithm, the influence of the penalty parameter, and how the conjugate directions $d^l$ behave in the iterative process. An important pending task is to test the reliability of our approach for large scale problems, and how it compares with Spiess’ algorithm and recent developments, like bilevel programming techniques, but in the context of transit assignment. In particular, we want to apply this methodology to the transit network that represents the metropolitan area of Mexico City and surroundings.

Acknowledgements

We would like to thank the referees for their valuable comments and suggestions, which helped to improve substantially this document.

Figure 3: Scatter plots for the conjugate gradient method with segment counts and $k = 1000$. 

$A = -0.06, B = 0.95, R^2 = 0.95, RMSE = 1.55$

$A = -0.02, B = 1.00, R^2 = 1.00, RMSE = 0.38$
References


