□ 論 文 □

一般化重力模型 파라메터의 새로운 最優推定技法 開發

New Maximum Likelihood Estimation Algorithms for the Parameters of Generalized Gravity Model

尹 成 洵
(交通開發研究院 責任研究員)

目 次

국 문 요 약

본 논문에서는 최근에 소개된 일반화중력모형(Generalized Gravity Model)파라메터의 최우추정치(Maximum Likelihood Estimates)계산을 위한 새로운 알고리즘을 이론적으로 도출하였다.

개발된 알고리즘은 첫째 계산속도, 둘째 정밀도, 세째 모형변수(예컨데 통행시간, 통행비용 등)들 간에 공선성(multicollinearity)이 존재할 경우의 계산능력, 넷째 대규모 스케일의 기·종점자료(large O-D Matrices)에 적용시의 계산능력, 다섯째 모형변수의 갯수에 따른 계산능력의 평가기준에서 그 계산실적이 기존의 알고리즘과 비교 평가 되었다.

제안된 기법중에서 Modified Scoring기법은 계산속도 및 정밀도등 앞서 나열한 계산능력의 평가기준 중 모든 부문에서 매우 탁월한 계산실적을 보이는 것으로 판명되었다. 따라서 최선의 추정치를 보장하는 최우추정기법이 대규모 스케일의 교통계획 적용에도 큰 비용(시간)부담없이 손쉽게 적용될 수 있게 되었다. 제안된 새로운 알고리즘의 적용시 교통계획분야에 가져올 수 있는 기대효과는 다음과 같다. 첫째, 최우추정법이 대규모 O-D 통행표에 쉽게 적용될 수 있고 또한 PC등 소형 컴퓨터에서도 처리가 쉽다. 둘째, 모형설명변수의 자유로운 선택등 통계적실험(experimentation)을 가능케 한다. 세째, 중력모형이 내재되어 있는 결합모형(Combined Model)의 정산속도를 높인다. 네째, IVHS(Intelligent Vehicle and Highway System)와 같은 분야에서 온라인(On-line)모형정산을 가능케 할 수 있다.

1 The Generalized Gravity Model

Let N_{ij} be a flow (e.g., of people or vehicles) between an origin $i \in \mathcal{I}$ and a destination $j \in \mathcal{J}$ and let $c_{ij} = (c_{ij}^{(1)}, \ldots, c_{ij}^{(K)})'$ be a vector with components $c_{ij}^{(k)}$ which are different measures of separation (e.g., travel time, generalized cost, logarithm of travel time, etc.) between i and j. For all i and j, the gravity model may be written as

$$T_{ij} = E(N_{ij}) = A(i)\,B(j)\,F(c_{ij}). \quad (1)$$

We shall assume the N_{ij}'s have independent Poisson distributions [see Smith (1987) for the conditions which are necessary and sufficient for N_{ij}'s to be Poisson]. In the classical gravity models the $A(i)$'s and $B(j)$'s have been, a priori, set equal to functions of observable factors such as population, number of jobs and so on. However they led to inconsistencies in practice. In the *Generalized Gravity Model*, $A(i)$'s and $B(j)$'s are considered to be unknown parameters, the value of which are then to be estimated from observations of flows.

A form of $F(c_{ij})$ which subsumes as special cases the most commonly used expressions for $F(c_{ij})$, and appears to be general enough for most practical use, is the exponential form:

$$F(c_{ij}) = \exp[\theta' c_{ij}] \quad (2)$$

where $\theta = (\theta_1, \ldots, \theta_K)'$ consists of unknown parameters.

Methods based on Maximum Likelihood (ML) are deservedly among the most frequently used techniques for the estimation of gravity model parameters. ML estimates have pleasant asymptotic properties and are essentially unbiased for a very small sample of trips as frequently found in intra-urban O-D tables. It is well known that the ML estimate can be obtained by maximizing the log-likelihood function

$$\mathcal{L} = \sum_{ij}\{-A(i)B(j)\exp[\theta' c_{ij}] + N_{ij}(\log A(i) + \log B(j) + \theta' c_{ij}) - \log(N_{ij}!)\}, \quad (3)$$

which is equivalent to solving

$$T_{i*} = N_{i*} \quad \text{for } i \in I, \quad T_{*j} = N_{*j} \quad \text{for } j \in J \quad (4)$$

and

$$\sum c_{ij}^{(k)} T_{ij} = \sum_{ij} c_{ij}^{(k)} N_{ij} \qquad \text{for } k \in K, \quad (5)$$

where replacement of a subscript by a $*$ indicates that we have summed with respect to that subscript (e.g., $T_{i*} = \sum_j T_{ij}$, $T_{*j} = \sum_i T_{ij}$).

2 Derivation of the LDSF Procedure

Since the LDSF procedure is at the heart of the procedures proposed in this paper, we outline it below. The LDSF procedure is a linearized version of the well known DSF procedure, also called the row-column balancing algorithm, Furness iterations.

Let $\Delta O = (\Delta O_1, \ldots, \Delta O_I)'$ and $\Delta D = (\Delta D_1, \ldots, \Delta D_J)'$ to be small changes in the values of $O = (O_1, \ldots, O_I)'$ and $D = (D_1, \ldots, D_J)'$, and let ΔF_{ij} be a small change in F_{ij}. Then the LDSF procedure consists of the iterative steps

$$\Delta T_{ij}^{(2r-1)} = \Delta T_{ij}^{(2r-2)} + \frac{T_{ij}}{O_i}(\Delta O_i - \Delta T_{i*}^{(2r-2)}) \quad (6)$$

and

$$\Delta T_{ij}^{(2r)} = \Delta T_{ij}^{(2r-1)} + \frac{T_{ij}}{D_j}(\Delta D_j - \Delta T_{*j}^{(2r-1)}) \quad (7)$$

with initial values

$$\Delta T_{ij}^{(0)} = (T_{ij}/F_{ij})\Delta F_{ij}. \quad (8)$$

This procedure also converges under mild conditions. Let $\Delta T_{ij}^{(r)} \to \Delta T_{ij}$. If T_{ij}'s are of the form $A(i)B(j)F_{ij}$, then

$$T_{ij} + \Delta T_{ij} \approx [A(i) + \Delta A(i)][B(j) + \Delta B(j)][F_{ij} + \Delta F_{ij}] \quad (9)$$

for some $\Delta A(i)$s and $\Delta B(j)$s with

$$\sum_j \Delta T_{ij} = \Delta O_i$$

and

$$\sum_i \Delta T_{ij} = \Delta D_j.$$

An attractive feature of the LDSF procedure is that in most practical situations where T_{ij}'s are positive, convergence is very rapid — often one or two iterations are adequate. Let ΔF_{ij} be a small change in F_{ij} due to $\Delta\boldsymbol{\theta}$. Then

$$\Delta F_{ij} = \sum_{k=1}^{K} \frac{\partial F_{ij}}{\partial \theta_k} \Delta\theta_k = \sum_{k=1}^{K} c_{ij}^{(k)} F_{ij} \Delta\theta_k, \quad (10)$$

$$\Delta T_{ij}^{(0)} = \Delta F_{ij} T_{ij} / F_{ij} = T_{ij} \sum_{k=1}^{K} (c_{ij}^{(k)} \Delta\theta_k) \quad (11)$$

and if $\Delta\boldsymbol{O} = \Delta\boldsymbol{D} = \mathbf{o}$, the first two LDSF steps are (from (6) and (7))

$$\Delta T_{ij}^{(1)} = \Delta T_{ij}^{(0)} - \Delta T_{i*}^{(0)} \left(\frac{T_{ij}}{O_i}\right) \quad (12)$$

and

$$\Delta T_{ij}^{(2)} = \Delta T_{ij}^{(1)} - \Delta T_{*j}^{(1)} \left(\frac{T_{ij}}{D_j}\right). \quad (13)$$

Combining (12) and (13), we get

$$\begin{aligned}\Delta T_{ij}^{(2)} &= \Delta T_{ij}^{(0)} - \Delta T_{i*}^{(0)} \left(\frac{T_{ij}}{O_i}\right) \\ &\quad - [\Delta T_{*j}^{(0)} - \sum_i (\Delta T_{i*}^{(0)} \left(\frac{T_{ij}}{O_i}\right))] \left(\frac{T_{ij}}{D_j}\right) \\ &= \Delta T_{ij}^{(0)} - \Delta T_{i*}^{(0)} \left(\frac{T_{ij}}{O_i}\right) \\ &\quad - \Delta T_{*j}^{(0)} \left(\frac{T_{ij}}{D_j}\right) + \sum_i [T_{i*}^{(0)} \left(\frac{T_{ij}}{O_i}\right)] \left(\frac{T_{ij}}{D_j}\right), \end{aligned} \quad (14)$$

Using (11), it follows that

$$\Delta T_{ij} = \sum_{k=1}^{K} [S_{ij}^{(k)} \Delta\theta_k]. \quad (15)$$

where $O_i = T_{i*}$, $D_j = T_{*j}$, and

$$\begin{aligned} S_{ij}^{(k)} &= c_{ij}^{(k)} T_{ij} - \sum_j [c_{ij}^{(k)} T_{ij}] \left(\frac{T_{ij}}{O_i}\right) \\ &\quad - \sum_i [c_{ij}^{(k)} T_{ij}] \left(\frac{T_{ij}}{D_j}\right) \\ &\quad + \sum_i [\sum_j [c_{ij}^{(k)} T_{ij}] \left(\frac{T_{ij}}{O_i}\right)] \left(\frac{T_{ij}}{D_j}\right). \end{aligned} \quad (16)$$

The only unknowns in (15) are $\Delta\theta_k$'s. The $S_{ij}^{(k)}$'s are constants if T_{ij}'s are known and $O_i = T_{i*}$ and $D_j = T_{*j}$. Equation (15) will be used for developing the Modified Scoring Procedure and two of the Modified Gradient Search Procedures, presented in the following sections.

3 The Modified Scoring Procedure

One approach to obtaining ML estimates is by solving (4) and (5). For some reasonable initial choice of $\boldsymbol{\theta}$, we use the DSF procedure with $O_i = N_{i*}$ and $D_j = N_{*j}$. Then (4) would be solved. In order to solve (5) we could use a Newton-Raphson type procedure or, equivalently, a procedure akin to the method of scoring (Rao, 1973). That is, we could augment $\boldsymbol{\theta}$ to $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, compute the corresponding $(T_{ij} + \Delta T_{ij})$s and insert these into (5) to obtain

$$\sum_{ij} c_{ij}^{(k)}(T_{ij} + \Delta T_{ij}) = \sum_{ij} c_{ij}^{(k)} N_{ij}. \quad (17)$$

If these ΔT_{ij}s are computed using the LDSF procedure, (4) would remain approximately satisfied, while (17) would become

$$\sum_{ij} c_{ij}^{(k)}[\sum_k S_{ij}^{(k)} \Delta\theta_k] = \sum_{ij} c_{ij}^{(k)}(N_{ij} - T_{ij}). \quad (18)$$

A solution of (18) for $\Delta\boldsymbol{\theta}$ would result in $T_{ij} + \Delta T_{ij}$s which come closer to solving (17) and hence (5). The $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$s could become the $\boldsymbol{\theta}$s for the next iteration.

Notice that (18) gives a system of K linear equations in K unknowns ($\Delta\theta_k$'s) which can be solved by any of the standard solution methods (e.g., Gaussian elimination). Notice also that in practice K is usually quite small (e.g., 2, 3 or 4). The quantities on the right hand side of (18) which are $\partial\mathcal{L}/\partial\theta_k$'s, where $\mathcal{L}$ is the log-likelihood of the parameter vector $\boldsymbol{\theta}$, are defined as *efficient* scores for $\boldsymbol{\theta}$ (Rao, 1973). The maximum likelihood estimates are the values of $\boldsymbol{\theta}$ for which the efficient score vanishes. This gives the name 'method of scoring' to what is essentially a Newton-Raphson type algorithm.

We call the procedure just described as the Modified Scoring Procedure. Its steps may be summarized as follows:

Algorithm for Modified Scoring Procedure

1. Select an initial value $\boldsymbol{\theta}^{(0)}$.

2. Using the DSF procedure (with $O_i = N_{i*}$, and $D_j = N_{*j}$), obtain T_{ij}s and the coefficients $S_{ij}^{(k)}$'s (from (16)).

3. Solve (18) for $\Delta\theta_k$'s.

4. Revise the value of $\boldsymbol{\theta}^{(r)}$ as follows:

$$\boldsymbol{\theta}^{(r)} = \boldsymbol{\theta}^{(r-1)} + \Delta\boldsymbol{\theta}^{(r-1)} \quad (19)$$

5. Iterate steps 2, 3, and 4 until stable values of $\boldsymbol{\theta}$ are obtained (changes from iteration to iteration become negligible).

4 The Modified Gradient Search Procedures

Sen (1986) introduced a version of the gradient search procedure which we will refer to as the 'General Procedure'. This procedure, outlined below, was claimed by Sen to be the fastest available procedure which could handle large O-D matrices. We shall use it as a benchmark for comparisons and also modify it using the LDSF procedure. In each step of the General Procedure, $\boldsymbol{\theta}^{(r+1)}$ is obtained from the value $\boldsymbol{\theta}^{(r)}$ given by the previous step as follows:

$$\boldsymbol{\theta}^{(r+1)} = \boldsymbol{\theta}^{(r)} + \rho^{(r)}(\nu_1^{(r)}, \ldots, \nu_K^{(r)})' \quad (20)$$

where

$$\nu_k^{(r)} = \sum_{ij} c_{ij}^{(k)}[N_{ij} - T_{ij}(\boldsymbol{\theta}^{(r)})].$$

Then $T_{ij}(\boldsymbol{\theta}^{(r+1)})$s are obtained using the DSF procedure with $O_i = N_{i*}$ and $D_j = N_{*j}$. The value of $\rho^{(r)}$ is a solution to ρ in (21) below.

$$\sum_{k=1}^{K}\{C_k - \tau\sum_{ij} c_{ij}^{(k)} T_{ij}(\boldsymbol{\theta}^{(r)}) \exp[\sum_{l=1}^{K} \rho c_{ij}^{(l)} \nu_l^{(r)}]\}\nu_k^{(r)} = 0 \quad (21)$$

where

$$\tau^{-1} = \sum_{ij} T_{ij}(\boldsymbol{\theta}^{(r)}) \exp(\sum_{l=1}^{K} c_{ij}^{(l)} \rho \nu_l^{(r)})/T_{**}$$

$$\nu_k^{(r)} = \sum_{ij} c_{ij}^{(k)} N_{ij} - \sum_{ij} c_{ij}^{(k)} T_{ij}$$

and $C_k = \sum_{ij} c_{ij}^{(k)} N_{ij}$.

Since $\boldsymbol{\theta}$ is kept fixed at $\boldsymbol{\theta}^{(r)}$, $\mathcal{L}$ attains its maximum when $T_{i*}^{(r)} = N_{i*}$ and $T_{*j}^{(r)} = N_{*j}$. Consequently the DSF procedure which solves the ML equations (4) and (5) raises the value of $\mathcal{L}$. However, notice that only the last K components of the gradient $(\mathcal{L}, \zeta^{(r)})$ of the parameter vector were considered in the procedure. This was because $\partial\mathcal{L}/\partial A(i) = 0$ and $\partial\mathcal{L}/\partial B(j) = 0$ on account of the DSF procedure. However, it is obvious that the two conditions

$$\sum_j T_{ij}(\boldsymbol{\theta}^{(r)}) = N_{i*} \tag{22}$$

and

$$\sum_i T_{ij}(\boldsymbol{\theta}^{(r)}) = N_{*j} \tag{23}$$

need not exactly hold as ρ changes. After a suitable ρ is found and then the DSF procedure is applied (in the following iterations), a large part of the improvement due to the changing of $\boldsymbol{\theta}$ could be negated. This problem can be alleviated with the use of the LDSF procedure.

4.1 Procedures Ia and Ib

We conjecture, based on the discussion just made, that the incorporation of the LDSF procedure would improve the General Procedure. In order to do this, the log-likelihood function $\mathcal{L}$ is expressed in terms of the $(T_{ij} + \Delta T_{ij})$s where the ΔT_{ij}s are considered as additive *corrections* whose values are obtained using the LDSF procedure, i.e.,

$$\mathcal{L} = \sum_{ij}[-(T_{ij} + \Delta T_{ij}) + N_{ij}\log(T_{ij} + \Delta T_{ij}) - \log(N_{ij}!)]. \tag{24}$$

Notice that

$$\Delta\boldsymbol{\theta}^{(r)} = \rho^{(r)}\nabla(\mathcal{L}, \boldsymbol{\theta}^{(r)}) = \rho^{(r)}\boldsymbol{\nu}^{(r)} \tag{25}$$

The values of the $T_{ij}(\boldsymbol{\theta}^{(r)})$'s are the result of an application of the DSF procedure with $\boldsymbol{\theta} = \boldsymbol{\theta}^{(r)}, O_i = N_{i*}$, and $D_j = N_{*j}$. Equation (25) gives the direction of change for $\boldsymbol{\theta}$ along the gradient. The distance moved is given by ρ. In the direction of the gradient, $\mathcal{L}$ is a function of ρ alone. The use of the LDSF procedure in obtaining ΔT_{ij}s assume us that

$$T_{ij}(\boldsymbol{\theta}^{(r+1)}) \approx T_{ij}(\boldsymbol{\theta}^{(r)} + \Delta\boldsymbol{\theta}^{(r)}).$$

One method of obtaining $\rho^{(r)}$ is to set it equal to that value of ρ for which $d\mathcal{L}/d\rho = 0$. Now

$$\begin{aligned}\delta(\rho) = \frac{d\mathcal{L}}{d\rho} &\sum_i \frac{\partial\mathcal{L}}{\partial A(i)} \cdot \frac{dA(i)}{d\rho} + \\ &\sum_j \frac{\partial\mathcal{L}}{\partial B(j)} \cdot \frac{dB(j)}{d\rho} + \sum_k \frac{\partial\mathcal{L}}{\partial\theta_k} \cdot \frac{d\theta_k}{d\rho} \\ &= \sum_{ij} \frac{\partial\mathcal{L}}{\partial\Delta T_{ij}} \cdot \frac{d\Delta T_{ij}}{d\rho}\end{aligned} \tag{26}$$

where (from (24))

$$\frac{\partial\mathcal{L}}{\partial\Delta T_{ij}} = -1 + N_{ij}(T_{ij} + \Delta T_{ij})^{-1} \tag{27}$$

Since $\Delta\theta_k = \rho\nu_k$,

$$\begin{aligned}\frac{d\Delta T_{ij}}{d\rho} &= \sum_{k=1}^{K}[\nu_k\{c_{ij}^{(k)}T_{ij} - \sum_j[c_{ij}^{(k)}T_{ij}] \cdot \left(\frac{T_{ij}}{O_i}\right) \\ &\quad - \sum_i[c_{ij}^{(k)}T_{ij}] \cdot \left(\frac{T_{ij}}{D_j}\right) \\ &\quad + \sum_i[\sum_j[c_{ij}^{(k)}T_{ij}]\left(\frac{T_{ij}}{O_i}\right)]\left(\frac{T_{ij}}{D_j}\right)\}] \\ &= \sum_{k=1}^{K}[\nu_k \cdot S_{ij}^{(k)}] = \sum_{k=1}^{K} Q_{ij}^{(k)} \text{ (say)}\end{aligned} \tag{28}$$

where $S_{ij}^{(k)}$'s are given by (16). Therefore the function $\delta(\rho)$ can be written as

$$\delta(\rho) = \sum_{ij}\{[N_{ij}(T_{ij} + \rho\sum_{k=1}^{K} Q_{ij}^{(k)})^{-1} - 1] \cdot (\sum_{k=1}^{K} Q_{ij}^{(k)})\}. \tag{29}$$

Let ρ^* be a solution to $\delta(\rho) = 0$. Newton-Raphson iterations can be used for this purpose. This ρ^* then is the step size for this modified gradient search procedure, **Procedure Ia**.

The equation $\delta(\rho) = 0$ can be linearized using a Taylor series approximation and the resultant linear equation in one unknown can be solved. Such an approximation for (29) is

$$\sum_{ij}\left([N_{ij}(1+\rho\sum_{k=1}^{K} Q_{ij}^{(k)}/T_{ij})^{-1} \cdot T_{ij}^{-1} - 1] \cdot (\sum_{k=1}^{K} Q_{ij}^{(k)})\right)$$
$$= \sum_{ij}\left([N_{ij}(1-\rho\sum_{k=1}^{K} Q_{ij}^{(k)}/T_{ij}) \cdot T_{ij}^{-1} - 1] \cdot (\sum_{k=1}^{K} Q_{ij}^{(k)})\right) \tag{30}$$

Notice that this approximation (taking Taylor series and retaining only linear term) would be allowable since $\rho\sum_{k=1}^{K} Q_{ij}^{(k)} = \Delta T_{ij}$ in

$$(1+\rho\sum_{k=1}^{K} Q_{ij}^{(k)}/T_{ij})^{-1}$$

would be much smaller than T_{ij}. Now setting (30) to 0, we have

$$\rho = \frac{\sum_{ij}\left(\frac{N_{ij}}{T_{ij}} \cdot \sum_{k=1}^{K} Q_{ij}^{(k)}\right) - \sum_{ij}\left(\sum_{k=1}^{K} Q_{ij}^{(k)}\right)}{\sum_{ij}\left(\frac{N_{ij}}{T_{ij}^2} \cdot (\sum_{k=1}^{K} Q_{ij}^{(k)})^2\right)}. \tag{31}$$

The resultant procedure is designated as **Procedure Ib**. Notice that while a linear approximation will lead to greater computational efficiency, this approximation may increase the total number of iterations needed to converge.

Algorithm for Procedures Ia and Ib

1. Select an initial value $\boldsymbol{\theta}^{(0)}$.

2. Compute $F(c_{ij}'\boldsymbol{\theta}^{(r)})$.

3. Apply the DSF procedure to obtain the $T_{ij}(\boldsymbol{\theta}^{(r)})$'s.

4. Set $\nu_k^{(r)} = \sum_{ij} c_{ij}^{(k)}[N_{ij} - T_{ij}(\boldsymbol{\theta}^{(r)})]$.

5. Compute the terms in $\delta(\rho)$. The values of $Q_{ij}^{(k)}$ and hence $S_{ij}^{(k)}$ are obtained by the LDSF procedure.

6. Solve the equation $\delta(\rho) = 0$ for ρ either by

   Procedure Ia:
   using an iterative Newton-Raphson algorithm,

   $$\rho^{(n)} = \rho^{(n-1)} - \frac{\delta(\rho^{(n-1)})}{\delta'(\rho^{(n-1)})} \tag{32}$$

   where n denotes the n-th iteration in the Newton-Raphson method and,

   $$\delta'(\rho) = d\delta(\rho)/d\rho = -\sum_{ij}\left(\frac{N_{ij}}{(T_{ij}/\sum_{k=1}^{K} Q_{ij}^{(k)} + \rho)^2}\right) \tag{33}$$

   Procedure Ib: directly solving the linear equation for ρ given by (31).

   Either way, call the solution ρ^*

7. Update the parameter estimates:

   $$\hat{\boldsymbol{\theta}}^{(r+1)} = \hat{\boldsymbol{\theta}}^{(r)} + \rho\nabla(\mathcal{L}, \hat{\boldsymbol{\theta}}^{(r)}). \tag{34}$$

8. Iterate steps 2 through 7 until $\nu^{(r)}$ is small enough.

5 Performance of New Algorithms

The three new procedures described in the last two sections were compared with each other and with two existing procedures. The existing procedures were

1. the GLIM procedure which has been widely used for computing maximum likelihood estimates, and

2. the procedure given by Sen (1986) and which has been outlined in Section 4.

Computer programs in FORTRAN were written for all procedures. Four data sets were used for the evaluation of performance.

1. One data set is based on 25247 trips allocated over 42 origins and destinations in the village of Skokie, Illinois, USA. The $c_{ij}^{(1)}$ was set equal to the Skokie travel times. Distances were also computed, but since these would be highly correlated with travel times, these were permuted over the various origin-destination pairs to essentially remove multicollinearity. This permuted distance is $c_{ij}^{(2)}$.

2. Same as above but with the distances left unpermuted. This data set was used to check for effects of multicollinearity.

3. In order to evaluate the performance of the algorithms on a larger data set, the data set described in Sööt and Sen (1991) was used. This data set includes a work trip O-D matrix (N_{ij}) with almost two thousand origin and destination zones for the Chicago Metropolitan Area. Four $c_{ij}^{(k)}$'s were used in the model. Three were related to travel times by various modes. The fourth measure was an occupational compatibility index.

4. Another data set consisted of flows of patients between 250 zip code areas and 94 hospitals in the Chicago area. Data on travel times, distances and a payer/hospital match index (reflecting the fact that different hospitals deal with indigents differently) constituted the $c_{ij}^{(k)}$s.

In the sequel these data sets will be denoted by numbers, e.g., data sets #1 and #2 will also be called Skokie data set and data set #3 will be called the SEED data set. Data set #4 is the Hospital data set. In the next few sections we report on:

1. Comparison of the performance of the procedures in terms of computing time while maintaining specified precision levels.

2. Effects of *multicollinearity* in the separation measures.

3. Effects of initial values.

4. Effect of the number of separation measures (e.g. $K = 1, K = 2, K = 4$).

It should be pointed out that in all the trials we ran, we did not have any convergence problems (except for the GLIM procedure which ran out of memory even for moderate sized problems). In all cases, successive iterations yielded parameter values that moved fairly smoothly towards their limits. The successive values of the Modified Scoring Procedure and the GLIM Procedure appeared to be smoother than the others. No underflows or overflows in the computer occurred during execution of any of the alternative procedures, and given the smoothness of convergence, no overflows or underflows should be expected in normal use. This is in sharp contrast to some competing procedures such as the usual Scoring Procedure (Batty, 1976 and Sen, 1986).

5.1 Speed Comparison

In order to set up identical conditions to evaluate the relative speed of the procedures, we set the initial values of all parameters to zero. In order to make the convergence criterion similar we first ran the Modified Scoring Procedure setting the requirement that the efficient scores (the right side of (18)) be less than 10^{-10}. The parameter values obtained in this way [call these values $\hat{\boldsymbol{\theta}}_{\text{base}}$] were taken to be the 'correct values.' Whenever, in the course of iterations, parameter values $\hat{\boldsymbol{\theta}}_{\text{alt}}$ entered a disk of TOL radius centered on this point, i.e.,

$$\left|\frac{\hat{\theta}_{\text{alt}} - \hat{\theta}_{\text{base}}}{\hat{\theta}_{\text{base}}}\right| \leq \text{TOL},$$

the corresponding procedure is deemed to have converged and the time taken to achieve convergence is used for comparison. The different levels of TOL (the tolerance) varied from 10^{-1} to 10^{-5}. The results, using data set #1, are shown in Table 1. The CPU times are in seconds and Speed-up is given in parenthesis. Speed-up is defined as:

$$\frac{\text{CPU time of the General Procedure}}{\text{CPU time of the alternative procedure}}.$$

It is easily seen from the Table 1 that the Modified Scoring Procedure is the fastest of the procedures at all tolerance levels. The speed-up of the Modified Scoring Procedure grows rapidly from a factor of 7 to over 600 as the tolerance level gets smaller.

We now address the question of what effect the number, K, of separation measures has on performance. For this we continued to use Data set #1 but for the $K = 4$ case, we set $c_{ij}^{(3)}$ and $c_{ij}^{(4)}$ as logarithms of travel time and permuted distance ($c_{ij}^{(1)}$ and $c_{ij}^{(2)}$ remained travel time and permuted distance). For the case where only one measure of separation is considered, that is, $K = 1$, $c_{ij}^{(1)}$ was set as simply travel time. The results, using essentially the same method as that described above, are shown in Table 2. The increase of computer times, with increasing values of K, for the modified scoring procedure is reasonably small and its relative standing vis a vis competing procedures actually improves.

A major problem with algorithms for ML estimation of gravity model parameters is that frequently there are a very large number of them. Notice that there are I values of $A(i)$, J values of $B(j)$ and K values of θ_k, and while K is usually not large, I and J often are. Therefore, it is reasonable to ask how the procedures would handle larger data sets. A similar approach to the one described above for the Skokie data set (data set #1) was applied to the Hospital data set (data set #4). The results are shown in Tables 3.

It can be seen from Table 3 that in the General Procedure, Procedure Ia, and Procedure Ib, because the DSF procedure is imbedded in each iteration, execution time dramatically increases as the size of the application problem get larger. Notice that although the DSF procedure is also imbedded in the Modified Scoring procedure, it converged so fast that the increase in computing time due to the increase of application size is not as bad.

Since the Modified Scoring Procedure appears to be the best in terms of computational performance, we applied it to the SEED data (data set #3). Notice that this data set, with about two thousand origins and destinations and four separation measures, is by all standards large, and indeed of a size such that maximum likelihood methods typically would not be considered. The results, included in Table 4, show that the Modified Scoring Procedure does very well and handles very low tolerances without substantial increases in computational time.

5.2 Effect on Multicollinearity and Initial Values

For the General Procedure the results are not quite as pleasant when the $c_{ij}^{(k)}$'s are highly correlated, i.e. if real distances (unpermuted) are used as $c_{ij}^{(2)}$. This is illustrated in Table 5, which shows the rate of convergence for the Skokie data with $c_{ij}^{(1)}$ being travel time and $c_{ij}^{(2)}$ being actual distance (data set #2). In order to further investigate this issue a 'seriously' multicollinear data set was created by generating *pseudo random numbers* from a uniform (0,1) distribution and then adding these random numbers to Skokie travel time matrix $c_{ij}^{(1)}$ to get $c_{ij}^{(2)}$. Given that the mean of the $c_{ij}^{(1)}$'s is 12.8 (minutes), one should expect that this would lead to very serious multicollinearity.

The results from using this data set are shown in Table 5.

The slowness of convergence of the gradient search type algorithms near the optimal value, particularly in the presence of multicollinearity, is well-known and discussed in Sen and Matuszewski(1991). Nevertheless the results show that the two modified gradient search procedures, Procedure Ia and Ib, more or less overcomes this shortcoming. It is easily seen from Table 5 that the convergence rate of the Modified Scoring Procedure is not seriously affected by the presence of multicollinearity among the independent variables (separation measures) — not even by the presence of as 'serious' a level multicollinearity as one is likely to permit.

Another comparative merit of the Modified Scoring Procedure is that no matter what initial values one chooses convergence is very fast so that one need not choose the initial values too carefully.

6 Conclusion

The Modified Scoring Procedure appears to be the most computationally efficient procedure, for obtaining maximum likelihood estimates of generalized gravity model parameters. Moreover,

1. its convergence rate does not appear to be affected by multicollinearity among the separation measures,
2. a careful choice of the initial values is not necessary,
3. as the application scale (size of problem) increases, the relative speed-up over other alternative procedure grows,
4. as the tolerance is decreased (i.e., required precision is enhanced), its relative speed-up grows,
5. it performs very efficiently when $K = 2, 3$ and 4 whereas convergence rates of alternative procedures frequently slow down as K increases.

Even in absolute (i.e., non comparative) terms the Modified Scoring Procedure has excellent properties. Indeed, because of it, maximum likelihood, which clearly gives the best estimates, can be used routinely, even in very large applications. With regard to future research, I expect that the Modified Scoring Procedure will be extended to Logit Models and to Combined Models as the maximum likelihood estimation procedure of them by taking advantage of its generality.

Table 1: Comparison of Execution Times

Tolerance level	General proceure	Procedure Ia	Procedure Ib	Modified Scoring	GLIM procedure
$\leq \|10^{-1}\|$	.23 (1.0)	.22 (1.1)	.10 (2.3)	.03 (7.7)	3.84 (.1)
$\leq \|10^{-2}\|$	.46 (1.0)	.33 (1.4)	.17 (2.7)	.04 (11.5)	3.84 (.1)
$\leq \|10^{-3}\|$	.62 (1.0)	.44 (1.4)	.25 (2.5)	.05 (12.4)	3.84 (.2)
$\leq \|10^{-4}\|$	32.70 (1.0)*	.55 (59.5)	.30 (109.0)	.05 (654.0)	3.84 (8.5)
$\leq \|10^{-5}\|$	32.70 (1.0)*	.62 (52.7)	.32 (102.2)	.06 (545.0)	3.84 (8.5)
Average time per iteration	.03	.02	.01	.02	.77

(0,0) were used as the initial values except GLIM procedure.

CPU times are in seconds.

Speed-up factors are given in parenthesis.

*indicates procedure fails to meet stopping criteria at maximum of 1000 iterations.

Table 2: Performance for Various values of K

value of K	Procedure				
	General proceure	Procedure Ia	Procedure Ib	Modified Scoring	GLIM procedure
$K=1$	.26 (1.0)	.05 (5.2)	9.43 (n.a)*	.03 (8.7)	3.31 (.1)
$K=2$	.62 (1.0)	.44 (1.4)	.25 (2.5)	.05 (12.4)	3.84 (.2)
$K=4$	40.79 (1.0)*	23.89 (n.a)*	10.37 (n.a)*	.10 (407.9)	3.92 (10.4)

10^{-3} tolerance level was used for this table.

Table 3: Performance for Various Application Scales

Data Sets	Procedure				
	General proceure	Procedure Ia	Procedure Ib	Modified Scoring	GLIM procedure
Skokie	.62 (1.0)	.44 (1.4)	.25 (2.5)	.05 (12.4)	3.84 (.2)
Hospital	1058.94 (1.0)*	556.50 (n.a)*	393.76 (n.a)*	1.43 (740.5)	n.a

10^{-3} tolerance level was used for this table.

Table 4: Performance of Modified Scoring Procedure

Application Scale	Convergence Limit[a]			Average time per iteration
	$\leq 10^{-3}$	$\leq 10^{-5}$	$\leq 10^{-10}$	
Skokie	.09 (7)	.11 (9)	.16 (13)	.01
Hospital	4.54 (12)	5.30 (14)	7.78 (21)	.38
SEED	56.34 (14)	64.44 (16)	94.58 (24)	4.00

Number of iterations is given in parenthesis.

[a](in terms of absolute values of efficient scores)

Table 5: Performances on Multicollinearity and Initial Values

Multi collinearity	Initial values	Procedures				
		General proceure	Procedure Ia	Procedure Ib	Modified Scoring	GLIM procedure
permuted distance[a]	(0,0)	.62 (1.0)	.44 (1.4)	.25 (2.5)	.05 (12.4)	3.84 (.2)
	LS[d]	30.61 (n.a)*	.52 (1.2)	.35 (1.8)	.04 (15.5)	
real distance[b]	(0,0)	33.08 (1.0)*	.29 (114.1)	.25 (132.3)	.04 (827.0)	4.01 (8.3)
	LS	30.74 (1.1)*	.41 (80.7)	.28 (118.1)	.03 (1102.7)	
simulated distance[c]	(0,0)	30.80 (1.0)*	3.20 (9.6)	12.33 (n.a)*	.05 (616.0)	3.87 (8.0)
	LS	30.64 (1.0)*	1.35 (22.8)	12.30 (n.a)*	.05 (616.0)	

[a](absence of multicollinearity)
[b](presence of 'high' multicollinearity)
[c](presence of 'serious' multicollinearity)
[d](least square estimate values)
10^{-3} tolerance level was used for this table.

References

[1] Batty, Michael (1976) *Urban Modelling — Algorithms, Calibration, Predictions*, Cambridge, England: Cambridge University Press.

[2] Boyce, D.E. and Ashish Sen (1990) *Estimation Procedures for Transportation Network Equilibrium Models*, in K. Peschel, ed., Infrastructure and Space-Economy, New York: Springer-Verlag.

[3] Dahlquist, Germund and Ake Bjorck (1974) *Numerical Methods*, Englewood Cliffs, N.J.: Prentice-Hall.

[4] Furness, K. P. (1965) *Time Function Iteration*, Traffic Engineering and Control 7, pp.458–460.

[5] McCullagh, P. and J.A. Nelder (1989) *Generalized Linear Models*, London: Chapman and Hall.

[6] Rao, C. Radhakrishna (1973) *Linear Statistical Inference and Its Application*, New York: Wiley.

[7] Sen, Ashish (1986) *Maximum Likelihood Estimation of Gravity Model Parameters*, Journal of Regional Science, Vol.26, No.3, pp. 461–474.

[8] Sen, Ashish and Zbigniew Matuszewski (1991) *Properties of Maximum Likelihood Estimation of Gravity Model Parameters*, Journal of Regional Science, Vol.31, No.4, pp. 469–486.

[9] Sen, Ashish and T.E. Smith (forthcoming) *Gravity Models of Spatial Interaction Behavior.*

[10] Sheffi, Yosef (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Method*, Englewood Cliffs, New Jersey: Prentice-Hall.

[11] Smith, Tony E. (1987) *Poisson Gravity Models of Spatial Flows,* Journal of Regional Science, Vol 27, pp. 315–340.

[12] Sööt, Siim and Ashish Sen (1991) *A Spatial Employment and Economic Development Model,* Papers in Regional Science: The Journal of the RSAI 70, 2, pp. 149–166.

[13] Weber, James and Ashish Sen (1985) *Sensitivity of Gravity Model Parameters,* Journal of Regional Science, Vol. 25, pp. 317–336.

[14] Yun, Seongsoon (1992) *Maximum Likelihood Estimation Procedures for Generalized Gravity Model Parameters,* Urban Transportation Center, University of Illinois at Chicago, unpublished Ph.D. Dissertation.