OD 表の構成順序に着目した整数推定法

01405010 (株) 東芝 重電技術研究所 ＊中山 靖子 NAKAYAMA Yasuko
01701460 (株) 東芝 研究開発センター 米田 清 YONEDA Kiyoshi
01900750 (株) 東芝 研究開発センター 和田 維作 WADA Isaku
(株) 東芝 府中工場 久保 進 KUBO Susumu

1 はじめに

交通網や通信網において, 所定時間内に, どの地点からどの地点にどれだけの traffic があったかを調べたい. これをまとめた, 図 1 のような表を origin-destination table とか OD 表とよぶ. OD 表は網の基礎的な資料であり, これに基づいて設備計画や制御を行なう.

		到着点			合計
		1	…	C	
出発点	1	x_{11}	…	x_{1C}	$x_{1.}$
	⋮			⋮	⋮
	R	x_{R1}	…	x_{RC}	$x_{R.}$
合計		$x_{.1}$	…	$x_{.C}$	$x_{..}$

図 1: OD 表

各地点間の traffic 調査は高価である場合が多い. 一方, 一般に各地点における流入量と流出量は, 簡単に測定できる. そこで, 各出発点から網内への流入量と, 各到着点からの流出量から, 地点間別の traffic を推定したいという要求が古くからある.

多くの場合, OD 表の要素は整数であることがわかっている. たとえば, 交通の主体が離散である場合等である. このような問題に対しては, 実数推定より整数推定の方が, 結果の解釈が容易で分解能が高いので望ましい.

組合せ最適化を用いた OD 表の整数推定法 [3] は, (1) 制約条件が完全には満たされない, (2) 計算結果に再現性がない, (3) 計算時間が長い, といった問題点がある.

ここでは, 上の欠点を解消する新しい OD 表の整数推定法－構成順序推定法－を提案する. 第 2 節で OD 表推定問題の定式化を行ない, 第 3 節で数値例を紹介する. 第 4 節で構成順序推定法を説明し, 評価を行う. 最後に第 5 節でまとめと今後の課題を述べる.

2 OD 表の推定問題

図 1 の行和と列和から, 表の中身を推定したい. これは等式よりも変数の方が多い連立一次方程式になるので, 解は不定である. その中から 1 つだけ解を選ぶには, これくらい流量があるだろうという先験的な OD 表をあらかじめ作っておいて, それになるべく近い解を選ぶ.

より一般的には, 次のように書ける. 実数の (m,n) 次元の行列 A と非負の m 次元ベクトル q, それに n 次元ベクトル b が与えられる. 通常は $n \leq m$ である.

$$yA = b \tag{1}$$

を満し, かつ q に最も近い非負の m 次元ベクトル y を求める. 先験 OD 表の与え方や, 先験 OD 表と求める OD 表との遠さの表し方には, さまざまな方法がある.

Entropy 最大化原理 [2] によって遠さを Kullback-Leibler 情報量で計るならば, 目的関数は

$$\sum y_k \log(y_k/q_k) \longrightarrow \min \tag{2}$$

である.

この場合, 反復尺度法 (iterative scaling) [1] や一般の数理計画法で非負実数の解が求められる. 目的関数に情報量以外の凸関数, たとえば χ^2 を選んでも同様である.

3 数値例

実数推定と整数推定の例を示す. 表 1は入力データで, 表の中身が先験 OD 表, 表の最終行が列和, 最終列が行和を表している.

表 2は実数推定, 表 3は整数推定の計算結果, である.

表 1: 入力データ (先験 OD 表と行和・列和)

OD	1	2	3	4	計
1	1	0	1	1	2
2	1	0	1	1	1
3	0	0	1	1	2
計	1	0	3	1	5

表 2: 実数推定の結果

0.667	0.000	1.000	0.333
0.333	0.000	0.500	0.167
0.000	0.000	1.500	0.500

表 3: 整数推定の結果

1	0	1	0
0	0	1	0
0	0	1	1

4 構成順序推定法

等式制約 (1) の下で, 目的関数 (2) を最小化するような, 0 を含む自然数で構成されたベクトル y を求める.

y を q に近くなるように構成するという静的な視点に加えて, 動的な視点を導入する. すなわち, y は「変数 y_k を 1 つ選んでは 選んだ y_k に 1 を加える」という操作をくりかえして構成するものと考える. 選択するのは, その時点で最大の事後確率を持つ y_k である. これは y に代えて, y の構成順序を Bayes 推定することに相当する. 手順にまとめると, 次のとおり.

構成順序推定法の算法 0:

1. $r := \sum y_k$
2. $y := 0$ とする.
3. 以下を r 回くりかえす.
   (a) 問題 (1), (2) を解き, その実数解 z の最大要素を z_h とする.
   (b) $y_h := y_h + 1$ とする.
   (c) (1) において y_h を含む等式制約に関わる b_l について
   $b_l := b_l - 1$ とする.
   (d) $q_k := z_k$ ($k = 1, ..., m$ 但し $k \neq h$)
   $q_h := (z_h - 1)^+$ とする.
   ここに $x^+ := max(0, x)$.

上記の算法で, (d) において $0 \leq z_h - 1$ の場合, 更新された q は, 更新された b を含む制約式 (1) を既に満たしている. そのため, 次に (a) を実行して得られる実数解 z は, (d) で得られた q と等しくなる. したがって以下のように近道をしても, 同じ結果が得られる.

構成順序推定法の算法 1:

1. $s := \sum y_h$
2. 問題 (1), (2) を解く. その解 z を整数に打ち切って y とする.
3. $q_k := z_k - y_k$ $(k = 1, ..., m)$ とする.
4. $r := s - \sum y_h$
5. **算法 0** の手順 3 以降に同じ.

この算法によれば, 出発点と到着点がそれぞれ数十ある実用規模の問題を扱っても制約条件は完全に満たされ, 計算結果に再現性もある. また計算時間は実数推定法として反復尺度法を用いた場合, 20 × 20 の問題で総交通量が 2,442 のとき, SUN4 で約 5 時間半かかった. これは [3] の方法による場合の約 10 時間より短い.

5 おわりに

状態と観測が 0 を含む自然数のベクトルで表され, 状態の情報が線形変換で縮約されて観測される場合の状態推定法として, 構成順序推定法を提案した.

構成順序推定法は, 整数推定の問題を実数推定をくりかえし行うことに帰着する. その原理は entropy 最大化原理を, 解の空間でなく, 解を構成する到着順序全体の空間に適用することである.

ここで提案した方法は, 従来の組合せ最適化を用いた推定方法に比べて, 制約条件を完全に満たし, 結果に再現性があり, かつ計算時間が短いという点で優れている.

今後は, くりかえし使う実数推定を高速化し, 計算時間の短縮を図りたい.

参考文献

[1] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for loglinear models. *Annals of Mathematical Statistics*, Vol. 43, No. 5, pp. 1470–1480, 1972.

[2] H. Gzyl. *Maximum Entropy*. World Scientific, 1995.

[3] K. Yoneda. Integer estimation of origin-destination tables. *Trans. Inst. Electric Engineers of Japan*, Vol. 114-C, No. 4 (April), 1994.