

Urban Transportation Networks

Yossi Sheffi, Prentice Hall, 1985

Chapter 11 "Stochastic Network
Loading Models"

&

交通ネットワークの均衡分析

9.1~9.3 (P167~181)

確率的利用者均衡モデルの解法

2015/5/22 (金)

理論勉強会2015#6

B4 庄司惟

はじめに

→ Part 1

Part 2

Part 3

Part 4

Part 5

流れ

青本9.1 Dialのアルゴリズム

(経路を限定するロジット型確率配分)

青本9.2 経路を限定しないロジット型確率配分への拡張

青本9.3 プロビット型確率配分への拡張

(ところどころsheffi11章の内容を盛り込みます。)

はじめに

→ Part 1

Part 2

Part 3

Part 4

Part 5

交通量計算→**経路列挙**を伴う が、
可能経路→多すぎ、無理。

また、経路交通量よりも**リンク交通量**がわかれば
よい場合が多いから、可能ならば避けたい。

→リンクとノードを変数にとって議論。

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

Step0(準備)

起点 r から他の全てのノードへの最小交通費用 $c(i)$ を計算する。

$$c(i) \leftarrow Cmin[r \rightarrow i]$$

全リンクについて、リンク尤度を計算する。

$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & \text{if } c(i) \leq c(j) \\ 0 & \text{if otherwise} \end{cases}$$

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

Step1(前進処理)

起点 r から $c(i)$ の値の昇順(r から近い順)にノードを考える。各ノード i から流出するリンクのリンクウェイト $W[i \rightarrow j]$ を次式により計算する。

$$W[i \rightarrow j] = \begin{cases} L[i \rightarrow j] & \text{for } i = r \\ L[i \rightarrow j] \sum_{m \in I_i} W[m \rightarrow i] & \text{otherwise} \end{cases}$$

※ I_j はノード i に流入するリンクの始点集合

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

Step2(後退処理)

$C(j)$ の値の降順 (rから遠い順)にノードを考える。各ノードjに流入するリンクの交通量 x_{ij} を次式により計算する。

$$x_{ij} = (q_{rj} + \sum_{m \in O_j} x_{jm}) \frac{W[i \rightarrow j]}{\sum_{m \in I_j} W[m \rightarrow j]}$$

※ q_{rj} は始点からjに向かおうとするトリップ数 (一交通量)

※ I_j はノードiに流入するリンクの始点集合

※ O_j はノードiから流出するリンクの終点集合

Dialのアルゴリズム

Part 1

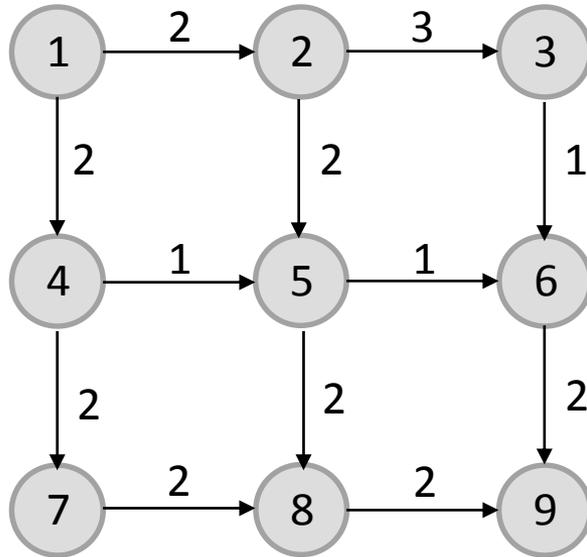
→ Part 2

Part 3

Part 4

Part 5

想定するネットワーク



※9個のノードに対してノード番号を、
12本のリンクに対してリンクコストを
それぞれ与える。

Dialのアルゴリズム

Part 1

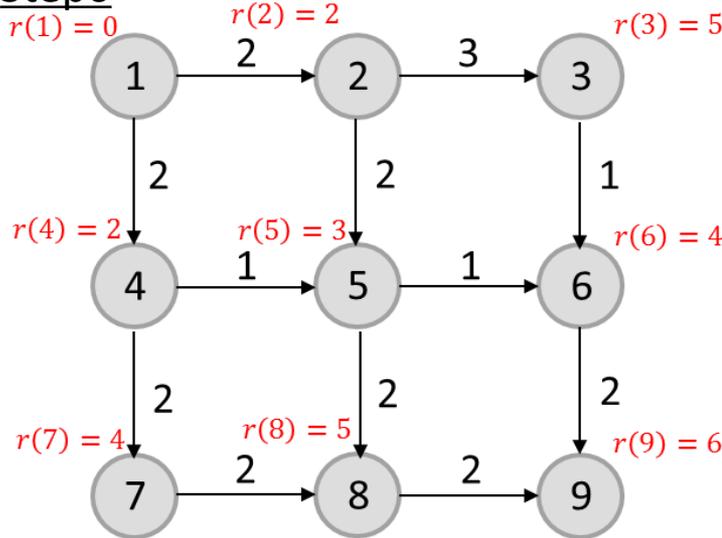
→ Part 2

Part 3

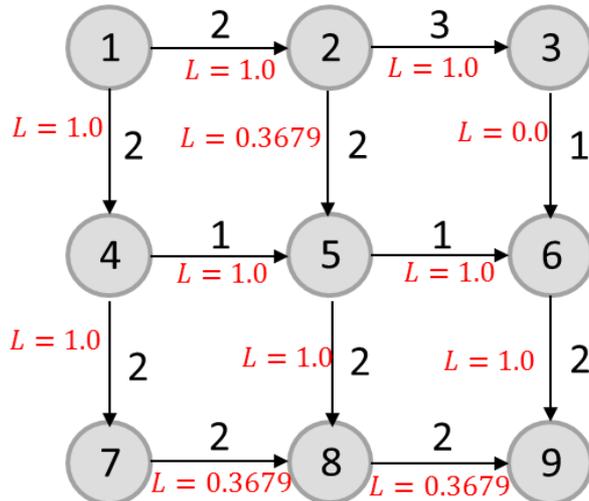
Part 4

Part 5

Step0



最小交通費用の計算
= 前回のDijkstra法で求める。



リンク尤度の計算($\theta=1$ とする)

$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & \text{if } c(i) < c(j) \\ 0 & \text{if otherwise} \end{cases}$$

例

$$L[4 \rightarrow 5] = \exp(3 - 2 - 1) = \exp(0) = 1$$

$$L[7 \rightarrow 8] = \exp(5 - 4 - 2) = \exp(-1) = 0.3679$$

Dialのアルゴリズム

Part 1

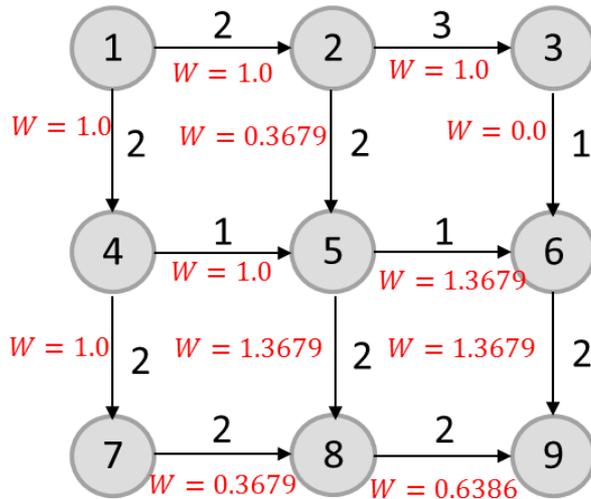
→ Part 2

Part 3

Part 4

Part 5

Step1



リンクウェイトの計算

$$W[i \rightarrow j] = \begin{cases} L[i \rightarrow j] & \text{for } i = r \\ L[i \rightarrow j] \sum_{m \in I_i} W[m \rightarrow i] & \text{otherwise} \end{cases}$$

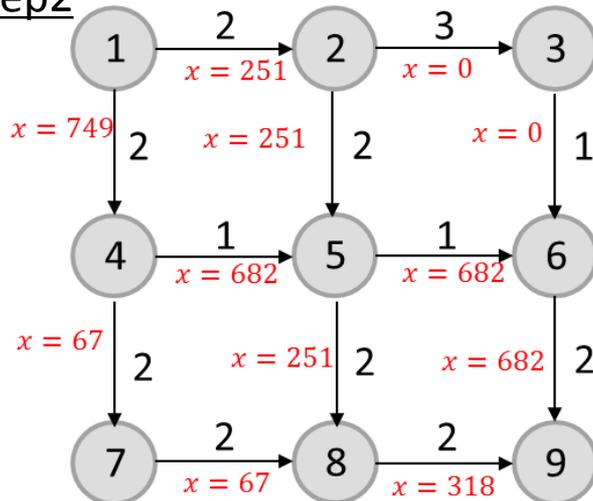
例

$$W[5 \rightarrow 6] = 1.0 (1.0 + 0.3679) = 1.367$$

$$W[3 \rightarrow 6] = 0.0(1.0) = 0.0$$

$$W[8 \rightarrow 9] = 0.3679(0.3679 + 1.3679) = 0.6386$$

Step2



リンク交通量の計算 ($q_{rj}=1000$)

$$x_{ij} = (q_{rj} + \sum_{m \in O_j} x_{jm}) \frac{W[i \rightarrow j]}{\sum_{m \in I_j} W[m \rightarrow j]}$$

例

$$x[8 \rightarrow 9] = (1000 + 0)0.6386 / (0.6386 + 1.3679) = 318$$

$$x[5 \rightarrow 8] = (0 + 318)1.3679 / (1.3679 + 0.3679) = 251$$

$$x[1 \rightarrow 4] = (0 + 67 + 682)1.0 / 1.0 = 749$$

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<Sheffiの記述>

- ・今紹介したのが“single-pass”アルゴリズム

$c(i) \leq c(j)$ で経路を限定している。

=一回の最小経路探索問題を解くことになる。

- ・ Sheffiは、“double-pass”の方から導入している

「始点から離れる、かつ、終点に近づく」という条件で経路を限定している。

=二回の最小経路探索問題を解くことになる。

- ・ “There is no a priori behavioral reason or any empirical evidence that would favor the original(double-pass) algorithm over the single-pass version.” とも言っています。

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<SheffiさんのDialアルゴリズム>

Step0(準備)

起点 r から他の全てのノードへの最小交通費用 $c(i)$ を計算する。

$$c(i) \leftarrow Cmin[r \rightarrow i]$$

終点 s に対し、ほかの全てのノード i からの最小交通費用 $s(i)$ を計算する。

$$s(i) \leftarrow Cmin[i \rightarrow s]$$

全リンクについて、リンク尤度を計算する。

$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & \text{if } c(i) < c(j) \text{ and } s(i) > s(j) \\ 0 & \text{if otherwise} \end{cases}$$

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

- ・ リンク尤度の置き方

$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & \text{if } c(i) < c(j) \\ 0 & \text{if otherwise} \end{cases}$$

→何故このような形になったのか？

(答) **ロジットモデルとの等価性**を持たせるため

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<Dialのアルゴリズムとロジットモデルの等価性> (式的な理解)

ODペアrsがk番目経路($r \rightarrow A \rightarrow B \dots Y \rightarrow Z \rightarrow s$)
を選択する確率 P_k は、

$$P_k = \frac{W[Z \rightarrow s]}{\sum_m W[m \rightarrow s]} \frac{W[Y \rightarrow Z]}{\sum_m W[m \rightarrow Z]} \\ \dots \frac{W[A \rightarrow B]}{W[A \rightarrow B]} \frac{W[r \rightarrow A]}{W[r \rightarrow A]} \\ \dots \frac{W[m \rightarrow B]}{\sum_m W[m \rightarrow B]} \frac{W[m \rightarrow A]}{\sum_m W[m \rightarrow A]}$$

リンクウエイトの定義

$$W[i \rightarrow j] = L[i \rightarrow j] \sum_{m \in I_i} W[m \rightarrow i]$$

を用いると、

$$P_k = L[Z \rightarrow s] L[Y \rightarrow Z] \\ \dots L[A \rightarrow B] L[r \rightarrow A] / \sum_m W[m \rightarrow s]$$

さらにリンク尤度の定義

$L[i \rightarrow j] = \exp[\theta\{c(j) - c(i) - t_{ij}\}]$
を用いると、

$$P_k = \exp[\theta\{c(s) - c(Z) - t_{Zs}\}] \\ \times \exp[\theta\{c(Z) - c(Y) - t_{YZ}\}] \\ \dots \exp[\theta\{c(B) - c(A) - t_{AB}\}] \\ \times \exp[\theta\{c(A) - c(r) - t_{rA}\}] \\ / \sum_m W[m \rightarrow s] \\ = \exp[\theta\{c(s) \\ - (t_{rA} + t_{AB} + \dots + t_{YZ})\}]$$

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<Dialのアルゴリズムとロジットモデルの等価性>

経路列挙することなしに、

あたかも**経路をロジット型確率で選択**しているかのように表現できている！！

(すごい！？)

Dialのアルゴリズム

Part 1

→ Part 2

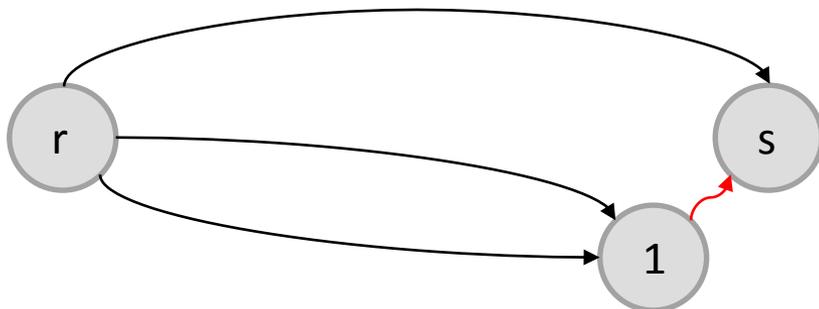
Part 3

Part 4

Part 5

<Sheffiの記述>

- ・ Dialがロジットベースであるがゆえに生じる不適切な配分例



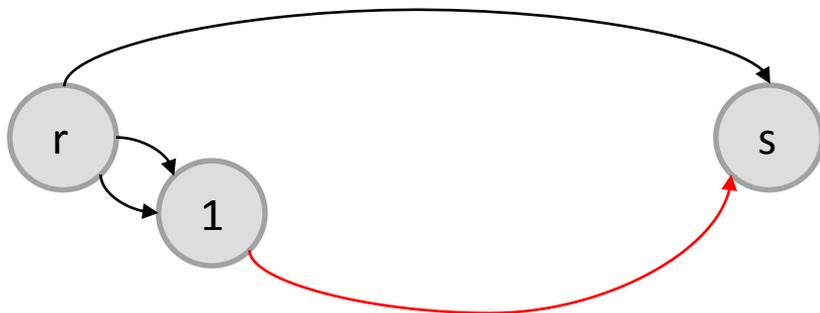
← ほぼ3経路と認識される

↑

どちらの場合でも、計算上
3つの経路はそれぞれ1/3ずつ
交通量を分担することになる

↓

← ほぼ2経路と認識される



Overlapping routesに過剰に配分してしまう！

Dialのアルゴリズム

Part 1

→ Part 2

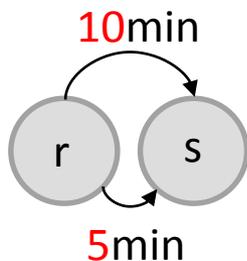
Part 3

Part 4

Part 5

<Sheffiの記述>

・コストの「差」に基づいて確率が決定するという仮定による不適切な配分例



←直感的には、
ほぼ全ての利用者が5minの方を利用する



↑直感的には、
結構な数の利用者が125minの方も利用する

が、ロジット型配分では両者とも同じ配分になってしまう！！！！

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<Sheffiの記述>

- ・ これらの二つの問題点は、
ロジット型における「誤差項」を見直すことで解消できる。

Overlapping → **誤差項相関**が考慮されていないこと
「差」 → **誤差項の分散**が同一定数であること

つまり、**プロビット型**を考えればよいことになる！
(スライドPart4でやります。)

Dialのアルゴリズム

Part 1

→ Part 2

Part 3

Part 4

Part 5

<Sheffiの記述>

- ・ Dial法では、前段階として経路を“reasonable”か“unreasonable”に分類している。
- ・ “reasonable”経路にのみOD交通量を振り分ける。
- ・ “unreasonable”経路は、実際には選択肢にすら含まれない。
- ・ 最短経路探索では「一つの視点から全てのノードへの最短経路」が求まるが、その性質を生かしてきれていない。

Dialのアルゴリズム

Part 1

→ Part 2

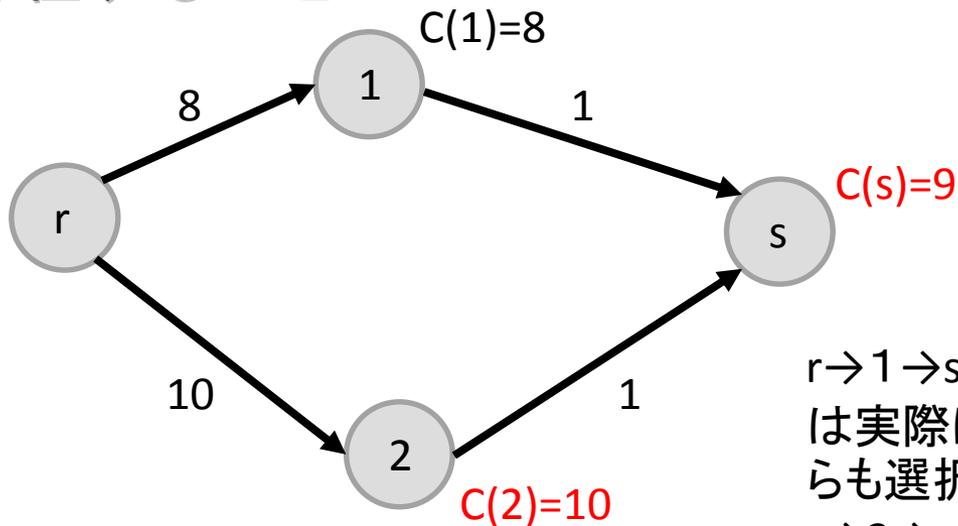
Part 3

Part 4

Part 5

<Dialのアルゴリズムの問題点>

- ・ 経路の限定が行われてしまっていること
- ・ それゆえに生じる非現実的なフローパターンが存在すること



$r \rightarrow 1 \rightarrow s$ と $r \rightarrow 2 \rightarrow s$ の二つの経路は実際にはコストが近く、どちらも選択されうるが、Dial法では $r \rightarrow 2 \rightarrow s$ の方はリンク尤度の時点で0と判断され、交通量が割り振られない。

ロジット型確率配分 (経路限定無し)

Part 1

Part 2

→ Part 3

Part 4

Part 5

- Markov連鎖配分 (Markov Chain Assignment) (佐佐木) を応用する。

Part3の概要

- ①MCAの概要説明
- ②ロジット型確率配分とMCAの関係性
- ③実際の計算、実装のために

①MCA（マルコフ連鎖配分）

Part 1

Part 2

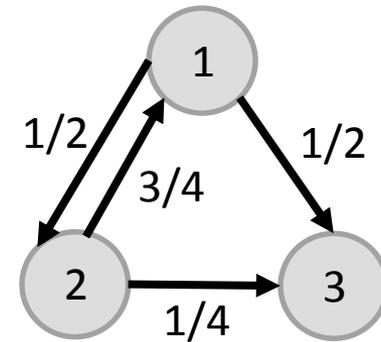
→ Part 3

Part 4

Part 5

- ・ **マルコフ性** = その過程の将来状態の条件付き確率分布が、現在状態のみに依存し、過去のいかなる状態にも依存しない特性（wikipediaより）
- ・ ノードから各ノードへの移動確率を表す **遷移確率行列**

$$\begin{array}{c} \begin{array}{ccc} & 1 & 2 & 3 \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 0 & 1/2 & 1/2 \\ 3/4 & 0 & 1/4 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \end{array}$$



①MCA（マルコフ連鎖配分）

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ 一般に遷移確率行列は、終点の数をa個、全ノード数をn個として

$$P = \begin{pmatrix} a & n-a \\ I & \mathbf{0} \\ R & Q \end{pmatrix}$$

I : 終点の移動を表す単位行列

R : 終点以外から終点に移る

Q : 終点以外から終点以外に移る

- ・ 実際のネットワークで効いてくるのは行列Q。
起点の数をg個として

$$Q = \begin{pmatrix} g & n-a-g \\ \mathbf{0} & Q_1 \\ \mathbf{0} & Q_2 \end{pmatrix}$$

いかなるノードも始点には戻らない

Q_1 : 始点から始点以外に移る

Q_2 : 始点以外から始点以外に移る

①MCA（マルコフ連鎖配分）

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・最終的に得たいのはリンク交通量。
- ・直感的には Q^n 計算したらなんかできそう。
- ・ Q^n の (i,j) 要素が何を表すのか！？

$$Q^2 = \left(\begin{array}{c} \boxed{\text{i行目}} \end{array} \right) \left(\begin{array}{c} \boxed{\text{j列目}} \end{array} \right) = \left(\begin{array}{c} (i,j) \end{array} \right)$$

つまり、 $i \rightarrow (\text{任意のノード}) \rightarrow j$ となる確率を表す。

$$Q^3 = \left(\begin{array}{c} \boxed{\text{i行目}} \end{array} \right) \left(\begin{array}{c} \boxed{\phantom{\text{ }}} \\ \boxed{\phantom{\text{ }}} \\ \boxed{\phantom{\text{ }}} \end{array} \right) \left(\begin{array}{c} \phantom{\text{ }} \\ \phantom{\text{ }} \\ \phantom{\text{ }} \end{array} \right)$$
$$= \left(\begin{array}{c} \boxed{\text{i行目}} \end{array} \right) \left(\begin{array}{c} \boxed{\text{j列目}} \end{array} \right) = \left(\begin{array}{c} (i,j) \end{array} \right)$$

つまり、 $i \rightarrow (\text{任意のノード}) \rightarrow (\text{任意のノード}) \rightarrow j$ となる確率を表す。

①MCA（マルコフ連鎖配分）

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ Q^n の (i,j) 要素は、最初 i にいた人（車）が、 n 回後に j にいる確率を表す。
- ・ リンク交通量は、通過する人が現在何回目の遷移状態なのかは問わない。（=全部足せばよい）
- ・ 以下の行列の無限級数の (i,j) 要素は、「 i から j に向かうある一台の車が各ノードを通る回数の期待値」を表すといえる。

$$I + Q^1 + Q^2 + \dots + Q^n + \dots = (I - Q)^{-1}$$

$$Q = \begin{pmatrix} 0 & Q_1 \\ 0 & Q_2 \end{pmatrix} \text{より、} (I - Q)^{-1} = \begin{pmatrix} I & Q_1(I - Q_2)^{-1} \\ 0 & (I - Q_2)^{-1} \end{pmatrix}$$

①MCA（マルコフ連鎖配分）

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ $(I - Q)^{-1}$ からわかるのは「ノード選択確率」 $P(i)$
- ・ ほしいのは「リンク選択確率」 P_{ij}

$$P_{ij} = P(i) \cdot p(j|i)$$

- ・ リンク選択率 = ノード選択率 × 遷移確率
 - ・ $p(j|i)$ はノード*i*から*j*への遷移確率であるが、これは遷移確率行列から既に与えられている。
- = 上の式は既に完成！！

→ じゃあもう終わり？

②ロジット型確率配分とMCAの関係性

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ MCAでは遷移確率行列を観測交通量などから算出することになっている。

- ・ つまり、選択にかかる**行動論的な視点が一切ない**。

- ・ 「行列ですべてのパターンを考慮する」というところは変えずに、ロジットモデル的な性質を与えてみよう。

(c_k^{rs} はODペアrsのk番目経路の計測可能コスト)

つまり、

$$P_k^{rs} = \prod_{ij} p(j|i)^{\delta_{ij,k}^{rs,n}} = \frac{\exp[-\theta c_k^{rs}]}{\sum_{k=1}^{\infty} \exp[-\theta c_k^{rs}]}$$

を満たすように、遷移確率行列 $p(j|i)$ を与える。

$\delta_{ij,k}^{rs,n}$: ODペアrs間のk番目経路がリンクijをn回通過するならn,そうでないなら0をとる！！

②ロジット型確率配分とMCAの関係性

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ 結論から言うと . . .

$$p(j|i) = \exp[-\theta t_{ij}] \frac{V_{js}}{V_{is}}$$

$$V_{is} = \sum_{k=1}^{\infty} \exp[-\theta c_k^{is}]$$

(証明)

$$\begin{aligned} P_k^{rs} &= \prod_{ij} p(j|i)^{\delta_{ij,k}^{rs,n}} = \prod_{ij} \exp[-\theta t_{ij}]^{\delta_{ij,k}^{rs,n}} \prod_{ij} \frac{V_{js}}{V_{is}}^{\delta_{ij,k}^{rs,n}} \\ &= \exp[-\theta c_k^{rs}] \prod_{ij} \frac{V_{js}}{V_{is}}^{\delta_{ij,k}^{rs,n}} \end{aligned}$$

②ロジット型確率配分とMCAの関係性

Part 1

Part 2

→ Part 3

Part 4

Part 5

(証明続き)

$$\prod_{ij} \frac{V_{js}}{V_{is}}^{\delta_{ij,k}^{rs,n}} = \exp \left[\sum_{ij} (\ln V_{js} - \ln V_{is}) \delta_{ij,k}^{rs,n} \right]$$
$$= \exp[(\ln V_{ss} - \ln V_{rs})] = \frac{1}{V_{rs}} = \frac{1}{\sum_{k=1}^{\infty} \exp[-\theta c_k^{rs}]}$$

よって

$$p_k^{rs} = \frac{\exp[-\theta c_k^{rs}]}{\sum_{k=1}^{\infty} \exp[-\theta c_k^{rs}]} \quad \blacksquare$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ 行列 V はどのように計算すればよいか？

$V_{is} = \sum_{k=1}^{\infty} \exp[-\theta c_k^{is}]$ を見ればわかるが、サイクル含む無限経路を考慮しているため、経路列挙では評価できない。

- ・ 先ほどと同様の行列演算アプローチが有効！

$$I + Q^1 + Q^2 + \dots + Q^n + \dots = (I - Q)^{-1}$$

- ・ こんな感じのやつ。無限級数が V に等しくなれば、簡単。では今回はどういった Q を定義すればよいだろうか？

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ またまた結論から言うと、次のような行列 W を置きます。

$$w_{ij} = \begin{cases} \exp[-\theta t_{ij}] & \text{リンク}ij\text{が存在} \\ 0 & \text{その他} \end{cases}$$

- ・ この行列のべき乗 W^n の要素 (i,j) は、先の Q^n と同様に考えると、

$$w_{ij}^n = \sum_{k \in K_n^{ij}} \exp[-\theta c_{k,n}^{ij}]$$

- ・ n 本のリンクを通過してノード i とノード j を結ぶ経路の集合 K_n^{ij} に対して、経路コスト $c_{k,n}^{ij}$ を足しあわせたものを表す。

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ V を求める上で、何本のリンクで結ばれているかという情報は関係ない。(全て足せばよい)

$$V = W^1 + W^2 + \dots + W^n + \dots = (I - W)^{-1} - I$$

- ・ これは計算可能！

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

- ・ 結果として逆行列計算が二つある。

$$(I - Q_2)^{-1} \& (I - W)^{-1}$$

- ・ これは計算量的に**高負荷**。
- ・ 以下のような工夫で、逆行列計算を一回にできる。
- ・ 多起点・1終点のODペアに分割する。（以下の操作を、終点の数だけ繰り返すことになる。）

このとき、

$$W = \begin{matrix} & \begin{matrix} 1 & g & n-g-1 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & W_1 \\ W_{s2} & 0 & W_2 \end{pmatrix} \end{matrix}$$

※ g :起点の数

のように表現することが可能！

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

$$W = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & W_1 \\ W_{s2} & 0 & W_2 \end{pmatrix}$$

で表されるならば、

$$[I - W]^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ V_{s1} & I & V_1 \\ V_{s2} & 0 & V_2 \end{pmatrix}$$

と、表すことができる。

ただし、

$$\begin{aligned} V_{s1} &= W_1 [I - W_2]^{-1} W_{s2} \\ V_{s2} &= [I - W_2]^{-1} W_{s2} \\ V_1 &= W_1 [I - W_2]^{-1} \\ V_2 &= [I - W_2]^{-1} \end{aligned}$$

※時間的に、これを示すことはしません。
が、暇なときに手を動かしてみてください。

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

遷移確率行列 $Q = \begin{pmatrix} 0 & Q_1 \\ 0 & Q_2 \end{pmatrix}$ は、 $p(j|i)$ を表す。

$q_{ij} = p(j|i) = \exp[-\theta t_{ij}] \frac{V_{js}}{V_{is}}$ 、 $w_{ij} = \exp[-\theta t_{ij}]$ より、行列 Q と W と V には何らかの関係が成り立ってそう。

成分表記では $q_{ij} = w_{ij} \frac{V_{js}}{V_{is}}$ であるから、これを行列表記に直すと、

i が起点、 j が起点以外するとき $Q_1 = \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}}$

i も j も起点以外するとき $Q_2 = \overline{V_{s2}}^{-1} W_2 \overline{V_{s2}}$

ただし、

$$W = \begin{pmatrix} 1 & g & n-g-1 \\ 0 & 0 & 0 \\ 0 & 0 & W_1 \\ W_{s2} & 0 & W_2 \end{pmatrix} \quad V = [I - W]^{-1} - I = \begin{pmatrix} 1 & 0 & 0 \\ V_{s1} & 0 & V_1 \\ V_{s2} & 0 & V_2 - I \end{pmatrix}$$

$\overline{V_{s1}}$ は V_{s1} を対角要素にした $g \times g$ 対角行列

$\overline{V_{s2}}$ は V_{s2} を対角要素にもつ $n-g-1 \times n-g-1$ 対角行列

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

(結論)

$Q_1 = \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}}$ 、 $Q_2 = \overline{V_{s2}}^{-1} W_2 \overline{V_{s2}}$ をもとに、求めるべきノード選択率行列 $P(i) = (I - Q)^{-1} = \begin{pmatrix} I & Q_1(I - Q_2)^{-1} \\ 0 & (I - Q_2)^{-1} \end{pmatrix}$ の逆行列計算 $(I - Q_2)^{-1}$ を省くことができる。

(証明)

$$Q_1(I - Q_2)^{-1} = \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \underline{(I - \overline{V_{s2}}^{-1} W_2 \overline{V_{s2}})^{-1}}$$

逆行列の補助定理

$$(A + BDC)^{-1}$$

$$= A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}$$

$$= \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \left\{ I + \overline{V_{s2}}^{-1} \left(W_2^{-1} - \overline{V_{s2}} \cdot \overline{V_{s2}}^{-1} \right)^{-1} \overline{V_{s2}} \right\}$$

$$= \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \left\{ I + \overline{V_{s2}}^{-1} \underline{(-I + IW_2^{-1}I)^{-1}} \overline{V_{s2}} \right\}$$

$$= \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \left\{ I + \overline{V_{s2}}^{-1} \{-I - (W_2 - I)^{-1}\} \overline{V_{s2}} \right\}$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

ここで、 $V_1 = W_1[I - W_2]^{-1}$ を用いると、

$$\begin{aligned} Q_1(I - Q_2)^{-1} &= \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \left\{ I + \overline{V_{s2}}^{-1} \{-I - (W_2 - I)^{-1}\} \overline{V_{s2}} \right\} \\ &= \overline{V_{s1}}^{-1} W_1 \overline{V_{s2}} \left[-\overline{V_{s2}}^{-1} (W_2 - I)^{-1} \overline{V_{s2}} \right] \\ &= \overline{V_{s1}}^{-1} W_1 [I - W_2]^{-1} \overline{V_{s2}} \\ &= \overline{V_{s1}}^{-1} V_1 \overline{V_{s2}} \end{aligned}$$

つまり、 $(I - Q_2)^{-1}$ において逆行列計算をする必要はない！

交通量が始点からのみ発生するとすれば、この $Q_1(I - Q_2)^{-1}$ を計算すれば事足りるからである。

とくにトリップが始点 r で発生し終点 s に向かう形で発生したときの、ノード i の選択確率 $P(i)^{rs}$ は、

$$P(i)^{rs} = V_{ri} V_{is} / V_{rs}$$

よってリンク選択確率は、

$$P_{ij}^{rs} = p(j|i) P(i)^{rs} = w_{ij} \frac{V_{js}}{V_{is}} \times \frac{V_{ri} V_{is}}{V_{rs}} = \frac{V_{ri} w_{ij} V_{js}}{V_{rs}}$$

③実際の計算、実装のために

- ・しかし、逆行列計算一回にしても、実際はかなり高負荷
- ・SOR法 (Successive Over Relaxation method) を用いる。

SOR法 (逐次加速緩和法)

- ・ n 元連立一次方程式を反復法で解く手法の一つであり、ガウス＝ザイデル法に加速パラメータを導入してその修正量を拡大することで、更なる加速を図った手法である。

(wikipediaより引用)

Part 1

Part 2

→ Part 3

Part 4

Part 5

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

SOR法(に入る前に・・・)

・今求めたいのは「リンク交通量」

・ $P_{ij}^{rs} = \frac{V_{ri}W_{ij}V_{js}}{V_{rs}}$ より、行列 V の中でも必要な要素は V_{ri} と V_{js} と V_{rs} のみ。(全ての要素を計算する必要はない。)

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

SOR法(に入る前に・・・)

～Markov連鎖を用いた確率的配分の流れ(まとめ)～

Step1

行列 W の値を設定 $w_{ij} = \exp(-\theta t_{ij})$

Step2

起点 r に対して、 $V_r = [V_{r1}, \dots, V_{rn}]$ を以下の線形連立方程式より算出。

$$[I - W]^T V_r^T = W_r^T$$

終点 s に対して、 $V_s = [V_{1s}, \dots, V_{ns}]^T$ を以下の線形連立方程式より算出。

$$[I - W]V_s = W_s$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

SOR法(に入る前に・・・)

～Markov連鎖を用いた確率的配分の流れ(まとめ)～

Step3

V_3 は以下のように簡単に求まる。

$$V_3 = V_1 W_2 \text{ or } V_3 = W_1 V_2$$

Step4

ODペア別のリンク交通量 y_{ij}^{rs} は以下で計算される。

$$y_{ij}^{rs} = q_{rs} \frac{V_{ri} w_{ij} V_{js}}{V_{rs}}$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

SOR法

連立方程式

$$\sum_{j=1}^n a_{ij}x_j = b_i (i = 1, \dots, n)$$

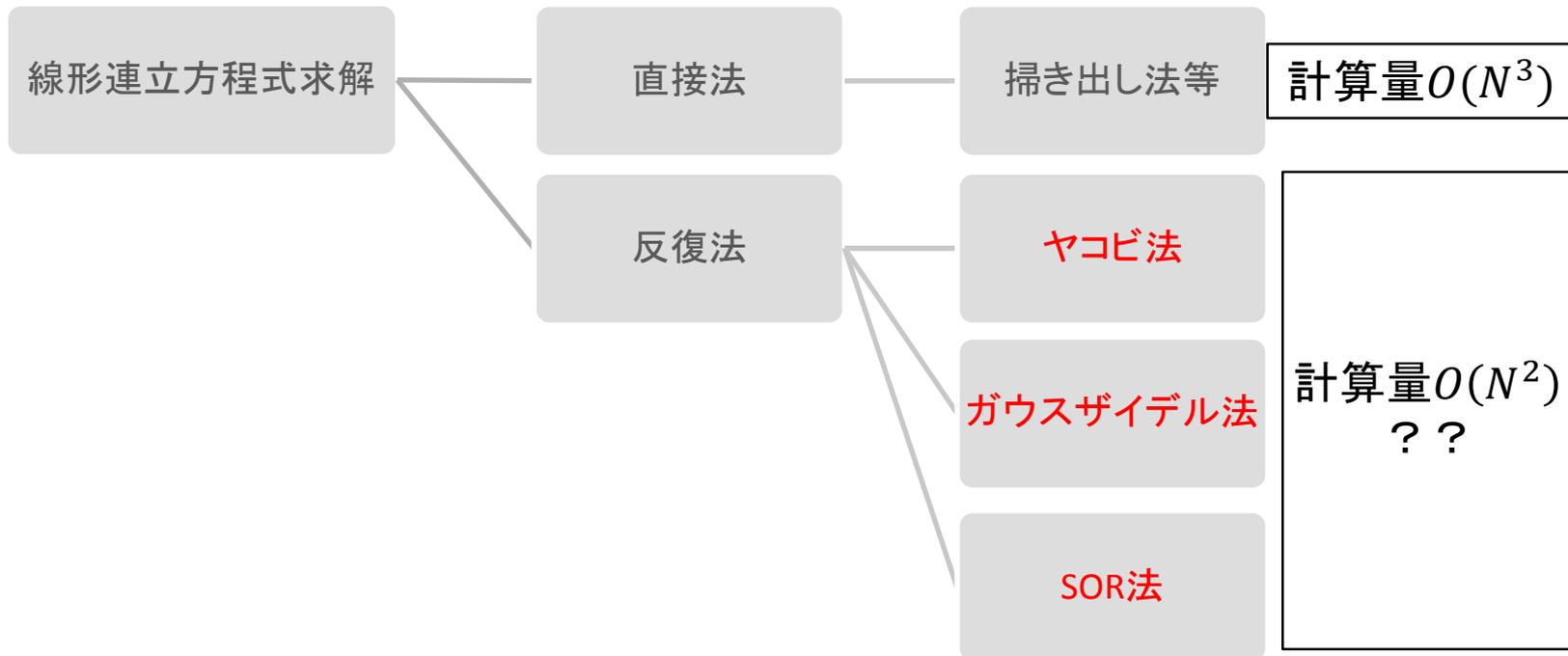
を解くとき、適当な初期値 $x(0)$ に対して以下の式で解の改訂を繰り返す。

$$x_i(t+1)$$

$$= (1 - \omega)x_i(t) + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j < i} a_{ij}x_j(t+1) - \sum_{j > i} a_{ij}x_j(t) \right]$$

一分でわかる線形連立方程式 求解のためのアルゴリズム

$\sum_{j=1}^n a_{ij}x_j = b_i (i = 1, \dots, n)$ を解くために...



直接法だと、(係数行列のタイプにもよるが、)計算量 $O(N^3)$ が確実にかかるし、計算が終わるまで解がわからない。

反復法では、途中で計算を止めても、多少は真の解にちかいものが得られる。また、工夫次第で計算量を抑えることも可能である。

一分でわかる線形連立方程式 求解のためのアルゴリズム

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

・ヤコビ法

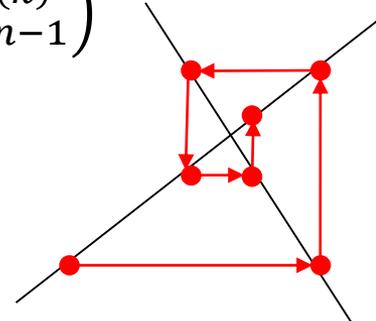
K回目の反復における解の推定値 $x^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix}$ に対して

次ステップの推定値

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)} \right)$$

\vdots

$$x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)} \right)$$



一分でわかる線形連立方程式 求解のためのアルゴリズム

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

・ガウスザイデル法

次ステップの推定値

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \cdots - a_{1n}x_n^{(k)} \right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} \cdots - a_{2n}x_n^{(k)} \right)$$

$$x_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} \cdots - a_{3n}x_n^{(k)} \right)$$

⋮

$$x_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(k+1)} \cdots - a_{n,n-1}x_{n-1}^{(k+1)} \right)$$

各段階で得られる最新のデータを代入している！

一分でわかる線形連立方程式 求解のためのアルゴリズム

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

・SOR法

各段階に、同一のパラメータ ω をかませる。

$$\tilde{x}_1^{(k+1)} = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \cdots - a_{1n}x_n^{(k)} \right)$$

$$x_1^{(k+1)} = x_1^{(k)} + \omega(\tilde{x}_1^{(k+1)} - x_1^{(k)})$$

$$\tilde{x}_2^{(k+1)} = \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} \cdots - a_{2n}x_n^{(k)} \right)$$

$$x_2^{(k+1)} = x_2^{(k)} + \omega(\tilde{x}_2^{(k+1)} - x_2^{(k)})$$

$$\tilde{x}_3^{(k+1)} = \frac{1}{a_{33}} \left(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} \cdots - a_{3n}x_n^{(k)} \right)$$

⋮

$$\tilde{x}_n^{(k+1)} = \frac{1}{a_{nn}} \left(b_n - a_{n1}x_1^{(k+1)} \cdots - a_{n,n-1}x_{n-1}^{(k+1)} \right)$$

$$x_n^{(k+1)} = x_n^{(k)} + \omega(\tilde{x}_n^{(k+1)} - x_n^{(k)})$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

Step2

起点 r に対して、 $V_r = [V_{r1}, \dots, V_{rn}]$ を以下の線形連立方程式より算出。

$$[I - W]^T V_r^T = W_r^T$$

ここにSOR法を適用する。起点 r に対して、

$V_{rj}(k + 1)$

$$= \begin{cases} (1 - \omega)V_{rj}(k) - \omega \left[1 - \sum_{ij \in In(j)} W_{ij} \bar{V}_{ri} \right] & \text{if } j \in E(r) \\ (1 - \omega)V_{rj}(k) + \omega \left[\sum_{ij \in In(j)} W_{ij} \bar{V}_{ri} \right] & \text{if } j \notin E(r) \end{cases}$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

Step2

終点 s に対して、 $V_s = [V_{1s}, \dots, V_{ns}]^T$ を以下の線形連立方程式より算出。

$$[I - W]V_s = W_s$$

終点 s に対してもまったく同様に、

$$V_{is}(k+1) = \begin{cases} (1 - \omega)V_{is}(k) - \omega \left[1 - \sum_{ij \in \text{Out}(i)} W_{ij} \bar{V}_{js} \right] & \text{if } i \in E(s) \\ (1 - \omega)V_{is}(k) + \omega \left[\sum_{ij \in \text{Out}(i)} W_{ij} \bar{V}_{js} \right] & \text{if } i \notin E(s) \end{cases}$$

③実際の計算、実装のために

Part 1

Part 2

→ Part 3

Part 4

Part 5

MCA利用した配分のまとめ
(利点)

- ・Dial法での経路制限問題は解消。

(問題点)

- ・非現実的に過大なcycle flowの発生可能性

- ・行列 W の収束条件が満たされず解が発散する可能性

(特に、1サイクルのコストが比較的小さいサイクルが存在する場合に顕著！)

- ・ロジット型の欠点であるIIA特性の増幅可能性

→この配分法は、「万能」とであると解釈するのではなく、「Dialのアルゴリズムで問題が発生しうる場合の救護策」として用いるべき。

→この配分法の課題はつまるところcycle flowの適切な削除である。

プロビット型確率配分

Part 1

Part 2

Part 3

→ Part 4

Part 5

本来のプロビットモデルとは、

「誤差項の多重正規分布を決定するための共分散行列の全ての要素をパラメータとするモデル」

→現実的には推定できない？

→実際は共分散構造に何らかの仮定を設定し、パラメータ個数を減らす。

今回は、

①パラメータを**唯一**つだけ与えた場合を考え、

②**Monte Carlo Simulation**法で計算する

という感じで学習します。

プロビット型確率配分

①単一パラメータのとき

Part 1

Part 2

Part 3

→ Part 4

Part 5

- ・ 単一パラメータを仮定

$$\widetilde{t}_{ij} = t_{ij} + \widetilde{\varepsilon}_{ij}$$

\widetilde{t}_{ij} : 利用者が認知するリンクコスト

t_{ij} : 測定可能なリンクコスト

$\widetilde{\varepsilon}_{ij}$: 誤差項は、各リンクごとに独立に、以下のような正規分布に従うと仮定する。（ β は全リンクを通して共通）

$$\widetilde{\varepsilon}_{ij} \sim N(0, \beta t_{ij})$$

→ 分散が t_{ij} に比例しているのはなぜ？（Sheffi11）

→ **リンクごとに独立な誤差項を仮定したとき、経路ごとの認知コスト**はどんな感じになるか？

プロビット型確率配分

①単一パラメータのとき

Part 1

Part 2

Part 3

→ Part 4

Part 5

<Sheffiの一言>

「認知旅行時間の不確実性は、旅行時間よりもむしろ
リンクの長さに関係している。」



提案① $\widetilde{\varepsilon}_{ij} \sim N(0, \beta l_{ij})$

ただし、 l_{ij} :リンクijの**長さ**

提案② $\widetilde{\varepsilon}_{ij} \sim N(0, \beta t^{(0)}_{ij})$

ただし、 $t^{(0)}$:**フリーフロー時**の測定可能リンクコスト

リンクコストのフロー依存性は、認知旅行時間の不確実性にかませるべきではない(???)

プロビット型確率配分

①単一パラメータのとき

<Sheffiの一言>

「リンクコストが正規分布なのは現実にそぐわない」
なぜなら、「**リンクコストは非負**」だから。
非負の分布であるガンマ分布等を用いるべき？

しかし、

- ・ 誤差項の大きさは確定項に比べ小さい
- ・ 計算は、ガンマ分布が正規分布でよく近似できるところでしか行わない。

から、このまま正規分布を用いても不適切ではない！

プロビット型確率配分

①単一パラメータのとき

Part 1

Part 2

Part 3

→ Part 4

Part 5

- ・ 経路コストとリンクコストの関係は

$$c_k^{rs} = \sum_{ij} t_{ij} \delta_{ij,k}^{rs}$$

$$\tilde{c}_k^{rs} = \sum_{ij} \tilde{t}_{ij} \delta_{ij,k}^{rs}$$

- ・ 認知経路コスト \tilde{c}_k^{rs} の平均、分散共分散を求める

$$\begin{aligned} \text{平均は、} E(\tilde{c}_k^{rs}) &= E\left(\sum_{ij} \tilde{t}_{ij} \delta_{ij,k}^{rs}\right) = E\left(\sum_{ij} (t_{ij} + \tilde{\varepsilon}_{ij}) \delta_{ij,k}^{rs}\right) \\ &= c_k^{rs} \end{aligned}$$

$$\text{分散は、} \text{Var}(\tilde{c}_k^{rs}) = \sum_{ij} \text{Var}\{(t_{ij} + \tilde{\varepsilon}_{ij}) \delta_{ij,k}^{rs}\}$$

$$= \sum_{ij} \delta_{ij,k}^{rs,2} \text{Var}(\tilde{\varepsilon}_{ij}) = \sum_{ij} \beta t_{ij} \delta_{ij,k}^{rs} = \beta c_k^{rs}$$

プロビット型確率配分

①単一パラメータのとき

Part 1

Part 2

Part 3

→ Part 4

Part 5

$$\begin{aligned} \text{共分散は、} & Cov(\tilde{c}_{k1}^{rs}, \tilde{c}_{k2}^{rs}) = E\{(\tilde{c}_{k1}^{rs} - c_{k1}^{rs})(\tilde{c}_{k2}^{rs} - c_{k2}^{rs})\} \\ & = E\left\{\left(\sum_{ij} \tilde{\varepsilon}_{ij} \delta_{ij,k1}^{rs}\right)\left(\sum_{ij} \tilde{\varepsilon}_{ij} \delta_{ij,k2}^{rs}\right)\right\} \\ & = E\left\{\sum_{ij} \tilde{\varepsilon}_{ij}^2 \delta_{ij,k1}^{rs} \delta_{ij,k2}^{rs}\right\} = \sum_{ij} \delta_{ij,k1}^{rs} \delta_{ij,k2}^{rs} E(\tilde{\varepsilon}_{ij}^2) \\ & = \sum_{ij} \delta_{ij,k1}^{rs} \delta_{ij,k2}^{rs} E((\tilde{\varepsilon}_{ij} - E(\tilde{\varepsilon}_{ij}))^2) = \sum_{ij} \delta_{ij,k1}^{rs} \delta_{ij,k2}^{rs} \beta t_{ij} \\ & = \beta c_{k1,k2}^{rs} \end{aligned}$$

($c_{k1,k2}^{rs}$: 経路k1と経路k2が共に通るリンクコストの和)

→ 経路間の重なりが大きいほど、認知コスト間の相関が高くなる！

プロビット型確率配分

①単一パラメータのとき

Part 1

Part 2

Part 3

→ Part 4

Part 5

結局、利用者の認知経路コストは、

平均ベクトル $\begin{pmatrix} c_{k1}^{rs} \\ \vdots \\ c_{kn}^{rs} \end{pmatrix}$

分散共分散行列 $\begin{pmatrix} \beta c_{k1}^{rs} & \beta c_{k1,k2}^{rs} & \dots & \beta c_{k1,kn-1}^{rs} & \beta c_{k1,kn}^{rs} \\ \beta c_{k2,k1}^{rs} & \beta c_{k2}^{rs} & \vdots & \vdots & \vdots \\ \beta c_{k3,k1}^{rs} & \beta c_{k3,k2}^{rs} & \ddots & \beta c_{kn-2,kn-1}^{rs} & \beta c_{kn-2,kn}^{rs} \\ \vdots & \vdots & \vdots & \beta c_{kn-1}^{rs} & \beta c_{kn-1,kn}^{rs} \\ \beta c_{kn,k1}^{rs} & \beta c_{kn,k2}^{rs} & \dots & \beta c_{kn,kn-1}^{rs} & \beta c_{kn}^{rs} \end{pmatrix}$

の多重正規分布に従うことになる。が、
ロジットモデルのようにクローズドフォームでないので、
近似的求解手法をとることになる。

プロビット型確率配分

②Monte Carlo Simulation法

Part 1

Part 2

Part 3

→ Part 4

Part 5

- ・ モナコ公国にある賭博のまち「モンテカルロ」由来
- ・ **乱数**を用いたシミュレーション法

Wikipedia(<http://ja.wikipedia.org/wiki/%E3%83%A2%E3%83%B3%E3%83%86%E3%82%AB%E3%83%AB%E3%83%AD>)より転載



プロビット型確率配分

②Monte Carlo Simulation法

Part 1

Part 2

Part 3

→ Part 4

Part 5

モンテカルロ法による
プロビット型確率配分のアルゴリズム

Step0(初期化)

サンプリング繰り返しカウンタ $m=1$

Step1(サンプリング)

各リンク(i,j)について、 $t_{ij}^{(m)}$ を $\widetilde{t}_{ij} \sim N(t_{ij}, \beta t_{ij})$ からサンプリング

Step2

$\{t_{ij}^{(m)}\}$ に基づいて、各ODペア間の最短経路を探索し、その最短経路にOD交通量 q_{rs} を配分する。その結果のリンクフローパターンを $\{y_{ij}^{(m)}\}$

プロビット型確率配分

②Monte Carlo Simulation法

Part 1

Part 2

Part 3

→ Part 4

Part 5

モンテカルロ法による
プロビット型確率配分のアルゴリズム

Step3(平均化操作)

$$x_{ij}^{(m)} = \left(1 - \frac{1}{m}\right) x_{ij}^{(m-1)} + \frac{1}{m} y_{ij}^{(m)}$$

Step4(終了判定)

各リンク(i,j)において、 $x_{ij}^{(m)}$ の標準偏差を計算

$$\sigma_{ij}^{(m)} = \sqrt{\frac{1}{m(m-1)} \sum_{n=1}^m \left(y_{ij}^{(n)} - x_{ij}^{(m)}\right)^2}$$

$$\text{If } \max_{ij} \left\{ \frac{\sigma_{ij}^{(m)}}{x_{ij}^{(m)}} \right\} \leq \varepsilon \text{ then stop.}$$

otherwise $m = m + 1$ and go to step 1.

まとめ

Part 1

Part 2

Part 3

Part 4

→ Part 5

- Dialアルゴリズム (Part2)
- MCA理論を用いた経路非制限アプローチ(Part3)
- ロジット型とプロビット型の比較(Part4)

参考文献

Part 1

Part 2

Part 3

Part 4

→ Part 5

■吸収マルコフ過程による交通量配分理論（佐々木綱）

ネットに落ちてます。

■確率過程の基礎（R.デュレット,丸善出版）

研究室に常備してあります。

■線形連立方程式求解については以下を参照

(http://www-antenna.ee.titech.ac.jp/~hira/hobby/edu/em/mom/linear_system/lin_eqs.pdf)