MDCEV モデルにおける 尤度の計算効率についての考察 ―GPU による並列計算の導入―

福井 昭吾

概要

MDCEV モデルは離散―連続モデルの一つで、複数の商品や選択肢に対する消費行動を説明する有用な分析手法である。MDCEV モデルでは尤度の導出に大量の計算を必要とするため、一般にその推定には多大な時間がかかる。本研究では、MDCEV モデルの尤度の計算に GPU による並列計算を導入することで計算効率の向上を試みた。その結果、GPU を使って尤度を計算することで、CPU による並列化を使った計算と比較して約 1/10 から約 1/7 程度、並列化を行わない計算と比較すると約 1/30 から約 1/17 程度、計算時間を短縮した。また、比較的安価なシステムであっても GPU を使うことで計算時間を短くできることを示した。

Keywords MDCEV モデル、CUDA, GPU, 並列計算

1 はじめに

MDCEV(multiple discrete-continuous extreme value) モデルは離散―連続モデルの一つで Bhat (2005) により提示された。福田・力石 (2013) によれば、離散―連続モデルとは「離散的な選択行動と連続量に関する選択行動とが部分的に共通な要因によって関連付けられている状況を記述するための行動モデル」である。例えば、一つ以上の商品についてそれを購入するかどうか、購入するならばどのくらい購入するかという消費行動について、離散―連続モデルを用いて説明することが可能となる。MDCEV モデルは、複数個の選択対象を仮定した離散―連続モデルの一種である。

MDCEV モデルは、経済理論を直接の基礎とするモデルである。Bhat (2008) によれば、MDCEV モデルの考え方は Wales and Woodland (1983) の示した多変数の消費者需要モデルに対する Karush-Kuhn-Tucker(KKT) 法によるアプローチを基礎とする。MDCEV モデルでは、効用関数と予算制約をモデル化し Karush-Kuhn-Tucker(KKT) 条件から尤度関数を導出することで、最尤法により効用関数のパラメータを推定する。この効用関数には、各選択肢の価格だけでなく、各選択肢の性質や観測対象の属性を含めることができる。そのため、価格以外の様々な要因が消費量に強く影響しうる商品・サービスについて、MDCEV モデルはその消費行動を詳しく分析するための有効な分析手法といえる。また、MDCEV モデルでは各商品間の代替関係を考慮してそれらの消費量を同時に決定する。各商品の消費量を被説明変数とするようなモデルの場合、各商品の代替を含めて消費行動を考察することは極めて難しい。したがって、MDCEV モデルは経済理論の下で商品・選択肢の消費行動を詳細に考察することが可能であるといえる。

現在、交通・環境・観光などの分野で MDCEV モデルによる分析が行われている。例えば、Fang (2008) は自動車の保有や走行距離に対して住宅密度が与える影響を捉えることを目的に、National Household Travel Survey における 2001 年のカリフォルニア州のデータをもとに MDCEV モデルの推定とシミュレーションを行っている。また Kuriyama et al. (2017) は、従来の MDCEV モデルを予算・短期休暇・長期休暇の三つの制約条件からなるモデルに拡張し、日本の国立公園に対するレクリエーション需要について分析している。

MDCEV モデルは消費行動を捉えるうえで有用な分析手法である。しかし、MDCEV モデルを使った実際の分析では尤度の導出に大量の計算が必要となるためにモデル推定に長い時間がかかることが多い。特に、変数変換に伴うヤコビアンの導出、および、同時密度を求める際の数値積分が尤度の計算における大きなボトルネックとなる。また、サンプルサイズの増大や、制約条件の追加などのモデル拡張によりその計算量はさらに増加する。本研究では、MDCEV モデルにおける尤度関数の計算効率の向上を目的として、GPU による並列計算を導入することでその計算時間がどの程度短縮できるかを示す。

2 MDCEV モデルにおける尤度の導出

MDCEV モデルは、個人の効用関数と制約条件（予算制約や時間制約）を設定し KKT 条件を通じて最適消費量の同時分布を導く。その同時密度関数が MDCEV

モデルにおける尤度関数となる。以下では、MDCEV モデルの尤度関数をプログラムにどう落とし込むか明確に示すため、効用関数と制約条件の形状を限定して尤度関数を導出する手順を示す。より一般的な仮定のもとでの尤度関数の導出については Bhat (2005) を参照されたい。

個人の効用関数と制約条件を設定しよう。以下で用いるモデルは、Kuriyama et al. (2017) の 3 制約モデルを土台とし、これを時間と予算の 2 制約に減らし、旅行先を日帰り旅行と宿泊旅行の 2 種類に分けるように変えたものである。ある個人の効用関数 $U(\boldsymbol{x};\boldsymbol{\theta})$ を

$$
\begin{aligned}
U(\boldsymbol{x};\boldsymbol{\theta}) = {} & \frac{\exp(\epsilon_{z1})}{\alpha_{z1}} z_1^{\alpha_{z1}} + \exp(\epsilon_{z2}) \ln z_2 \\
& + \sum_{k=1}^{K} \gamma_D \psi_{xk}^D \ln\left(\frac{x_k^D}{\gamma_D} + 1\right) + \sum_{k=1}^{K} \gamma_S \psi_{xk}^S \ln\left(\frac{x_k^S}{\gamma_S} + 1\right) \qquad (1) \\
\psi_{xk}^D = {} & \exp\left(\boldsymbol{\delta}_D' \boldsymbol{q}_k + \epsilon_{xk}^D\right) \\
\psi_{xk}^S = {} & \exp\left(\boldsymbol{\delta}_S' \boldsymbol{q}_k + \epsilon_{xk}^S\right)
\end{aligned}
$$

とし、制約条件

$$
\begin{aligned}
z_1 + \sum_{k=1}^{K} p_k^D x_k^D + \sum_{k=1}^{K} p_k^S x_k^S = E \quad \text{（予算制約）} \\
z_2 + \sum_{k=1}^{K} t_k^D x_k^D + \sum_{k=1}^{K} t_k^S x_k^S = T \quad \text{（時間制約）}
\end{aligned}
$$

のもとで、各個人が効用最大化を行うと考える。ただし、K は観光地の数、$\boldsymbol{x} = (x_1^D \ \cdots \ x_K^D \ x_1^S \ \cdots \ x_K^S)'$ はある個人が各観光地へ旅行した回数で、x_k^D はある個人が観光地 k へ日帰り旅行をした回数、x_k^S はある個人が観光地 k へ宿泊旅行をした回数、z_1 は予算制約におけるニュメレール財の消費量、z_2 は時間制約におけるニュメレール財の消費量、$\boldsymbol{Q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K)$ は個人属性および観光地属性をまとめたもの、p_k^D は観光地までの日帰り旅行の旅行費用、p_k^S は観光地までの宿泊旅行の旅行費用、t_k^D は観光地までの日帰り旅行の旅行時間、t_k^S は観光地までの宿泊旅行の旅行時間、E は所得、T は余暇時間である。なお、$\boldsymbol{q}_k$ は M 個の属性が含まれると仮定し、そのパラメータ $\boldsymbol{\delta}_D$ と $\boldsymbol{\delta}_S$ もまた M 個の要素から構成され、$\boldsymbol{\delta}_D = (\delta_1^D \ \cdots \ \delta_M^D)', \boldsymbol{\delta}_S = (\delta_1^S \ \cdots \ \delta_M^S)'$ となる。さらに、ϵ_{z1} および ϵ_{z2} は z_1 および z_2 が効用に影響する際の撹乱要因であり、それぞ

れ位置パラメータを 0, スケールパラメータを σ_{z1} および σ_{z2} とするガンベル分布に従うとする。(x_k^D, x_k^S) についても、それが効用に影響する際の撹乱要因 $\boldsymbol{\epsilon}_x = (\epsilon_{x1}^D \ \cdots \ \epsilon_{xK}^D \ \epsilon_{x1}^S \ \cdots \ \epsilon_{xK}^S)'$ が存在する。これらはすべて位置パラメータを 0, スケールパラメータを σ_x とするガンベル分布に従い、異なる k の間で互いに独立であると仮定する。$\boldsymbol{\theta} = (\sigma_{z1} \ \sigma_{z2} \ \sigma_x \ \alpha_{z1} \ \gamma_D \ \gamma_S \ \boldsymbol{\delta}_D' \ \boldsymbol{\delta}_S')'$ はモデルパラメータであり、$\sigma_{z1}, \sigma_{z2}, \sigma_x > 0, \ \alpha_{z1} \leq 1, \ \gamma_D, \gamma_S > 0$ という制約を満たすとする。

以上のモデルにおいて、効用最大化のための第 1 階 KKT 条件は以下のようになる。

$$\begin{cases} \epsilon_{xk}^D < W_k^D & x_k^D \text{の最適解} x_k^{*D} \text{が 0 に等しいとき（端点解）} \quad (2) \\ \epsilon_{xk}^D = W_k^D & x_k^D \text{の最適解} x_k^{*D} \text{が 0 より大きいとき（内点解）} \quad (3) \\ \epsilon_{xk}^S < W_k^S & x_k^S \text{の最適解} x_k^{*S} \text{が 0 に等しいとき（端点解）} \quad (4) \\ \epsilon_{xk}^S = W_k^S & x_k^S \text{の最適解} x_k^{*S} \text{が 0 より大きいとき（内点解）} \quad (5) \end{cases}$$

ここで、W_k^D は

$$\frac{\partial U}{\partial x_k^D} = p_k^D \frac{\partial U}{\partial z_1} + t_k^D \frac{\partial U}{\partial z_2}$$

としたときの ϵ_{xk}^D の解、W_k^S は

$$\frac{\partial U}{\partial x_k^S} = p_k^S \frac{\partial U}{\partial z_1} + t_k^S \frac{\partial U}{\partial z_2}$$

としたときの ϵ_{xk}^S の解である。式 (1) の効用関数の場合、

$$W_k^D = \ln \left\{ p_k^D \exp(\epsilon_{z1}) z_1^{\alpha_{z1}-1} + t_k^D \exp(\epsilon_{z2}) \frac{1}{z_2} \right\} + \ln \left(\frac{x_k^D}{\gamma_D} + 1 \right) - \boldsymbol{\delta}_D' \boldsymbol{q}_k \quad (6)$$

$$W_k^S = \ln \left\{ p_k^S \exp(\epsilon_{z1}) z_1^{\alpha_{z1}-1} + t_k^S \exp(\epsilon_{z2}) \frac{1}{z_2} \right\} + \ln \left(\frac{x_k^S}{\gamma_S} + 1 \right) - \boldsymbol{\delta}_S' \boldsymbol{q}_k \quad (7)$$

となる。

式 (1) の効用関数とその制約条件からなる非線形計画で、KKT 条件は $\boldsymbol{x}^* = (x_1^{*D} \ \cdots \ x_K^{*D} \ x_1^{*S} \cdots \ x_K^{*S})'$ が $\boldsymbol{x}$ の最適解であるための必要十分条件である。すべての制約条件が線形制約であることから、KKT 条件は $\boldsymbol{x}^*$ が最適解であるための必要条件である。一方、$\boldsymbol{x} > \boldsymbol{0}$ の範囲で効用関数が凹関数であり、かつ、制約条件は $\boldsymbol{x}$ の線形関数であることから、KKT 条件は $\boldsymbol{x}^*$ が最適解であるための十分条

件でもある。したがって、$\boldsymbol{x}^*$ は大域最大点となる。さらに、効用関数は $\boldsymbol{x}$ について狭義の凹関数であるため、大域最大点となる $\boldsymbol{x}^*$ がただ一つ存在する[*1]。

また、最適解 $\boldsymbol{x}^*$ は確率変数 $\epsilon_{z1}, \epsilon_{z2}, \boldsymbol{\epsilon}_x$ の関数であることから、$\boldsymbol{x}^*$ もまた確率変数である。

MDCEV モデルの推定は、実際の旅行回数が最適解であると仮定し、同時密度 $f(\boldsymbol{x}^*;\boldsymbol{\theta})$ に基づく尤度関数 $L(\boldsymbol{\theta}|\boldsymbol{x}^*)$ を最大にするパラメータ $\boldsymbol{\theta}$ を推定値とする。以下では $\boldsymbol{x}^*$ の同時密度を導出する。

$\boldsymbol{x}^*$ の同時密度の導出に先立ち、確率変数 $\epsilon_{z1}, \epsilon_{z2}$ を所与とする $(\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+})$ の条件付き同時密度を求める。$\boldsymbol{x}^{*D0}$ は日帰り旅行回数のうち端点解からなるベクトル、$\boldsymbol{x}^{*D+}$ は日帰り旅行回数のうち内点解からなるベクトル、$\boldsymbol{x}^{*S0}$ は宿泊旅行回数のうち端点解からなるベクトル、$\boldsymbol{x}^{*S+}$ は宿泊旅行回数のうち内点解からなるベクトルである。また、式 (2) の ϵ^D_{xk} からなるベクトルを $\boldsymbol{\epsilon}^{D0}_x$, 式 (3) の ϵ^D_{xk} からなるベクトルを $\boldsymbol{\epsilon}^{D+}_x$, 式 (4) の ϵ^S_{xk} からなるベクトルを $\boldsymbol{\epsilon}^{S0}_x$, 式 (5) の ϵ^S_{xk} からなるベクトルを $\boldsymbol{\epsilon}^{S+}_x$ とする。このとき、$(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ に何らかの値を与えると、KKT 条件の式 (3) と式 (5) は $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ についての連立方程式となり[*2]、これを解くことで、$(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ の値を得ることができる[*3]。また、ある $\boldsymbol{\epsilon}_x$ に対して KKT 条件を満たす $\boldsymbol{x}^*$ はただ一つの大域最大点となるが、いま、$\boldsymbol{x}^*$ のうち $\boldsymbol{x}^{*D0}$ と $\boldsymbol{x}^{*S0}$ は $\boldsymbol{0}$ であるため、ある $(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ に対して式 (3) と式 (5) からなる連立方程式を満たす $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ はただ一つのみ存在することになる。したがって、式 (3) と式 (5) は $(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ と $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ の全単射となる。さらに、$(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ の値を与えれば $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ が定まることで z_1 および z_2 の値が決定し、式 (2) および式 (4) における W^D_k と W^S_k の値が定まる。これら W^D_k からなるベクトルを $\boldsymbol{W}^{D0}$,W^S_k からなるベクトルを $\boldsymbol{W}^{S0}$ としよう。

以上より、$(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ を定めれば $(\boldsymbol{W}^{D0\prime}\ \boldsymbol{W}^{S0\prime})$ が決まるため、確率変数

[*1] KKT 条件が最適解の必要十分条件であることの証明については、例えばチャン (1996) を参照せよ。

[*2] いま、$\boldsymbol{x}^*$ のうち端点解 $\boldsymbol{x}^{*D0}$ および $\boldsymbol{x}^{*S0}$ は $\boldsymbol{0}$ であり、これらは z_1 および z_2 に影響しない。したがって、式 (3) と式 (5) は内点解 $\boldsymbol{x}^{*D+}$ および $\boldsymbol{x}^{*S+}$ のみからなる連立方程式となる。

[*3] ここでは、任意の $(\boldsymbol{\epsilon}^{D+\prime}_x\ \boldsymbol{\epsilon}^{S+\prime}_x)$ に対して、式 (3) および式 (5) からなる連立方程式における $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ の解が必ず得られると仮定する。

$\epsilon_{z1}, \epsilon_{z2}$ を所与とした $\boldsymbol{x}^*$ の条件付き同時確率について

$$
\begin{aligned}
&P(\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}, \boldsymbol{x}^{*D+} \in \boldsymbol{A}_D, \boldsymbol{x}^{*S+} \in \boldsymbol{A}_S | \epsilon_{z1}, \epsilon_{z2}) \\
&= P(\boldsymbol{\epsilon}_x^{D0} < \boldsymbol{W}^{D0}, \boldsymbol{\epsilon}_x^{S0} < \boldsymbol{W}^{S0}, (\boldsymbol{\epsilon}_x^{D+\prime}\ \boldsymbol{\epsilon}_x^{S+\prime}) \in \boldsymbol{B} | \epsilon_{z1}, \epsilon_{z2}) \\
&= \int\int_{\boldsymbol{B}} P(\boldsymbol{\epsilon}_x^{D0} < \boldsymbol{W}^{D0}, \boldsymbol{\epsilon}_x^{S0} < \boldsymbol{W}^{S0} | \boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+}, \epsilon_{z1}, \epsilon_{z2}) \\
&\quad \times g(\boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+} | \epsilon_{z1}, \epsilon_{z2}) d\boldsymbol{\epsilon}_x^{D+} d\boldsymbol{\epsilon}_x^{S+}
\end{aligned} \tag{8}
$$

が成り立つ。ただし、g は $\epsilon_{z1}, \epsilon_{z2}$ が与えられたときの $\boldsymbol{\epsilon}_x^{D+}$ と $\boldsymbol{\epsilon}_x^{S+}$ の同時密度関数である。また、$\boldsymbol{B}$ は、$\boldsymbol{x}^{*D+}$ と $\boldsymbol{x}^{*S+}$ の範囲 $\boldsymbol{A}_D$ と $\boldsymbol{A}_S$ を、KKT 条件の式 (3) および (5) により $(\boldsymbol{\epsilon}_x^{D+\prime}\ \boldsymbol{\epsilon}_x^{S+\prime})$ の範囲に変換したものである。$\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}$, かつ、$\boldsymbol{x}^{*D+}$ と $\boldsymbol{x}^{*S+}$ が何らかの値をとる時の $\boldsymbol{x}^*$ の同時密度を $f(\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+} | \epsilon_{z1}, \epsilon_{z2})$ と置くと、式 (8) は以下のように変形できる。

$$
\begin{aligned}
&\int_{A_S}\int_{A_D} f(\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+} | \epsilon_{z1}, \epsilon_{z2}) d\boldsymbol{x}^{*D+} d\boldsymbol{x}^{*S+} \\
&= \int\int_{\boldsymbol{B}} P(\boldsymbol{\epsilon}_x^{D0} < \boldsymbol{W}^{D0}, \boldsymbol{\epsilon}_x^{S0} < \boldsymbol{W}^{S0} | \boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+}, \epsilon_{z1}, \epsilon_{z2}) \\
&\quad \times g(\boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+} | \epsilon_{z1}, \epsilon_{z2}) d\boldsymbol{\epsilon}_x^{D+} d\boldsymbol{\epsilon}_x^{S+}
\end{aligned} \tag{9}
$$

いま、$(\boldsymbol{\epsilon}_x^{D+\prime}\ \boldsymbol{\epsilon}_x^{S+\prime})$ と $(\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})$ は式 (3) と (5) より 1 対 1 に対応するため、式 (9) から以下の変数変換が導かれる。

$$
\begin{aligned}
&f(\boldsymbol{x}^{*D0} = \boldsymbol{0}, \boldsymbol{x}^{*S0} = \boldsymbol{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+} | \epsilon_{z1}, \epsilon_{z2}) \\
&= |\boldsymbol{J}|\, P(\boldsymbol{\epsilon}_x^{D0} < \boldsymbol{W}^{D0}, \boldsymbol{\epsilon}_x^{S0} < \boldsymbol{W}^{S0} | \boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+}, \epsilon_{z1}, \epsilon_{z2}) \\
&\quad \times g(\boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+} | \epsilon_{z1}, \epsilon_{z2})
\end{aligned} \tag{10}
$$

ここで、$|\boldsymbol{J}|$ は変数変換に伴うヤコビアンであり、$\boldsymbol{\epsilon}_x^+ = (\boldsymbol{\epsilon}_x^{D+\prime}\ \boldsymbol{\epsilon}_x^{S+\prime})'$, $\boldsymbol{x}^{*+} = (\boldsymbol{x}^{*D+\prime}\ \boldsymbol{x}^{*S+\prime})'$ とするとき $\boldsymbol{J} = \partial\boldsymbol{\epsilon}_x^+ / \partial\boldsymbol{x}^{*+\prime}$ である。今回のモデルの場合、$\boldsymbol{J}$ の

(i,j) 要素 $J_{i,j}$ は以下のように求められる。

$$J_{i,j} = \frac{\partial \epsilon_{xi}^{+}}{\partial x_j^{*+}} = \begin{cases} (A_{ij} + B_{ij}) \quad i \neq j \\ (A_{ij} + B_{ij})/C_i + 1/(x_i^{*+} + \gamma_D) \\ \qquad i = j \text{ かつ } x_i\text{が日帰り旅行の回数の場合} \\ (A_{ij} + B_{ij})/C_i + 1/(x_i^{*+} + \gamma_S) \\ \qquad i = j \text{ かつ } x_i\text{が宿泊旅行の回数の場合} \end{cases} \tag{11}$$

$$A_{ij} = p_i p_j \exp(\epsilon_{z1})(1-\alpha_{z1})(z_1^*)^{\alpha_{z1}-2}$$
$$B_{ij} = t_i t_j \exp(\epsilon_{z2})(z_2^*)^{-2}$$
$$C_i = p_i \exp(\epsilon_{z1})(z_1^*)^{\alpha_{z1}-1} + t_i \exp(\epsilon_{z2})(1/z_2^*)$$

ただし、ϵ_{xi}^{+} は $\boldsymbol{\epsilon}_x^{+}$ の第 i 要素、x_i^{*+} は $\boldsymbol{x}^{*+}$ の第 i 要素、z_1^* および z_2^* は $\boldsymbol{x}^*$ の値から予算制約および時間制約に基づいて計算した z_1 および z_2 である。また、p_i は $\boldsymbol{x}^{*+}$ の i 番目の要素に対応する旅行費用、t_i は $\boldsymbol{x}^{*+}$ の i 番目の要素に対応する旅行時間であり、p_j および t_j についても同様である。

式 (10) は、ϵ_{z1} および ϵ_{z2} が与えられた場合の $\boldsymbol{x}^*$ の条件付き同時密度関数である。したがって、これを ϵ_{z1} と ϵ_{z2} について積分することで $\boldsymbol{x}^*$ の同時密度関数が得られ、

$$\begin{aligned} &f(\boldsymbol{x}^{*D0} = \mathbf{0}, \boldsymbol{x}^{*S0} = \mathbf{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+}) \\ &= \int\int |\boldsymbol{J}|\, P(\boldsymbol{\epsilon}_x^{D0} < \boldsymbol{W}^{D0}, \boldsymbol{\epsilon}_x^{S0} < \boldsymbol{W}^{S0} | \boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+}, \epsilon_{z1}, \epsilon_{z2}) \\ &\quad \times g(\boldsymbol{\epsilon}_x^{D+}, \boldsymbol{\epsilon}_x^{S+} | \epsilon_{z1}, \epsilon_{z2}) h_1(\epsilon_{z1}) h_2(\epsilon_{z2}) d\epsilon_{z1} d\epsilon_{z2} \end{aligned} \tag{12}$$

となる。ただし、h_1 は ϵ_{z1} の密度関数、h_2 は ϵ_{z2} の密度関数である[*4]。実際にはこの積分を解析的に求めることは難しいため、モンテカルロ積分やガウス求積法などの数値積分の方法を用いて計算する。

実際の同時密度の計算は以下のように行う。はじめに、$(\epsilon_{z1}, \epsilon_{z2})$ をどう与えるかを決める。式 (12) の積分を計算する手法に応じて $(\epsilon_{z1}, \epsilon_{z2})$ の設定の仕方が変わる。以下では、数値積分の一つであるガウス—ルジャンドル法を使うことにしよう。ガウス—ルジャンドル法はガウス求積法の一種で、$\int_{-1}^{1} f(x)dx$ の近似値を $f(x)$ の加重和として近似する方法である。近似する際の積分点（$f(x)$ を計算する

[*4] もし、ϵ_{z1} と ϵ_{z2} が互いに独立でないと仮定するならば、式 (12) の $h_1(\epsilon_{z1})h_2(\epsilon_{z2})$ の部分は ϵ_{z1} と ϵ_{z2} の同時密度 $h(\epsilon_{z1}, \epsilon_{z2})$ となる。

x の位置）とウエイトは計算式に基づいて事前に設定する。例えば積分点数を 3 とする場合、積分点 a_l は $-\sqrt{3/5}, 0, \sqrt{3/5}$ の 3 点、それらに対応するウエイト w_l は $5/9, 8/9, 5/9$ となり、加重和 $\sum_l w_l f(a_l)$ を $\int_{-1}^{1} f(x)dx$ の近似値とする。積分の範囲を $[a, b]$ とする定積分をガウス―ルジャンドル法で計算する場合は

$$\frac{b-a}{2}\sum_l w_l f\left(\frac{b-a}{2}a_l + \frac{a+b}{2}\right)$$

により近似する。また、$x \in [x_1, x_2], y \in [y_1, y_2]$ の範囲で $f(x, y)$ の二重積分を計算する場合は、x についての積分点 a_l およびウエイト w_l と、y についての積分点 a_m およびウエイト w_m をそれぞれ求めて

$$\frac{x_2-x_1}{2}\frac{y_2-y_1}{2} \times \sum_l\sum_m w_l w_m f\left(\frac{x_2-x_1}{2}a_l + \frac{x_1+x_2}{2}, \frac{y_2-y_1}{2}a_m + \frac{y_1+y_2}{2}\right) \quad (13)$$

をその近似値とする。ガウス―ルジャンドル法により式 (12) の二重積分を計算する場合、数値積分の範囲と積分点数を定めれば積分点となる $(\epsilon_{z1}, \epsilon_{z2})$ の値が決まる。

次に、与えられた積分点ごとに条件付き密度を計算する。最適解 $\boldsymbol{x}^*$ の値を与え、確率変数 $(\epsilon_{z1}, \epsilon_{z2})$ の値としてガウス―ルジャンドル法の積分点を与えることで、KKT 条件の W_k^D と W_k^S、および、ヤコビ行列 $\boldsymbol{J}$ が計算できる。端点解の条件式における W_k^D および W_k^S からなるベクトルをそれぞれ $\boldsymbol{W}^{D0}, \boldsymbol{W}^{S0}$ とし、内点解の条件式における W_k^D および W_k^S からなるベクトルをそれぞれ $\boldsymbol{W}^{D+}, \boldsymbol{W}^{S+}$ とする。さらに、$\boldsymbol{W}^{D0}, \boldsymbol{W}^{S0}, \boldsymbol{W}^{D+}, \boldsymbol{W}^{S+}$ の i 番目の要素を、それぞれ $W_i^{D0}, W_i^{S0}, W_i^{D+}, W_i^{S+}$ とする。今回のモデルでは、ϵ_{xk} はすべての k について位置パラメータを 0, スケールパラメータを σ_x とするガンベル分布に従い、異なる k の間で互いに独立であると仮定している。したがって、$\boldsymbol{W}^{D+}$ の要素数を

M_D, $\boldsymbol{W}^{S+}$ の要素数を M_S とするとき、上記の条件付き密度は

$$
\begin{aligned}
&f(\boldsymbol{x}^{*D0}=\mathbf{0},\boldsymbol{x}^{*S0}=\mathbf{0},\boldsymbol{x}^{*D+},\boldsymbol{x}^{*S+}|\epsilon_{z1},\epsilon_{z2})\\
&=|\boldsymbol{J}|\prod_{i=1}^{K-M_D}P(\epsilon_i^{D0}<W_i^{D0}|\boldsymbol{\epsilon}_x^{D+},\boldsymbol{\epsilon}_x^{S+},\epsilon_{z1},\epsilon_{z2})\\
&\quad\times\prod_{i=1}^{K-M_S}P(\epsilon_i^{S0}<W_i^{S0}|\boldsymbol{\epsilon}_x^{D+},\boldsymbol{\epsilon}_x^{S+},\epsilon_{z1},\epsilon_{z2})\\
&\quad\times\prod_{i=1}^{M_D}f(W_i^{D+}|\epsilon_{z1},\epsilon_{z2})\prod_{i=1}^{M_S}f(W_i^{S+}|\epsilon_{z1},\epsilon_{z2})\\
&=|\boldsymbol{J}|\prod_{i=1}^{K-M_D}\exp\left\{-\exp\left(-\frac{W_i^{D0}}{\sigma_x}\right)\right\}\prod_{i=1}^{K-M_S}\exp\left\{-\exp\left(-\frac{W_i^{S0}}{\sigma_x}\right)\right\}\\
&\quad\times\prod_{i=1}^{M_D}\frac{1}{\sigma_x}\exp\left(-\frac{W_i^{D+}}{\sigma_x}\right)\exp\left\{-\exp\left(-\frac{W_i^{D+}}{\sigma_x}\right)\right\}\\
&\quad\times\prod_{i=1}^{M_S}\frac{1}{\sigma_x}\exp\left(-\frac{W_i^{S+}}{\sigma_x}\right)\exp\left\{-\exp\left(-\frac{W_i^{S+}}{\sigma_x}\right)\right\}
\end{aligned}
\tag{14}
$$

と計算できる。これをすべての積分点について計算する。

最後に、式 (12) より $\boldsymbol{x}^*$ の同時密度を求める。ガウス—ルジャンドル法の場合、設定した積分点ごとに条件付き密度の値を計算した後、事前に計算したウエイトおよび積分点 $(\epsilon_{z1,i},\epsilon_{z2,j})$ の密度を用いて式 (13) により数値積分を行えばよい。いま、ϵ_{z1} の最小値・最大値を $\epsilon_{z1}^{\min},\epsilon_{z1}^{\max}$, ϵ_{z2} の最小値・最大値を $\epsilon_{z2}^{\min},\epsilon_{z2}^{\max}$ と置くと、$\boldsymbol{x}^*$ の同時密度は

$$
\begin{aligned}
&f(\boldsymbol{x}^{*D0}=\mathbf{0},\boldsymbol{x}^{*S0}=\mathbf{0},\boldsymbol{x}^{*D+},\boldsymbol{x}^{*S+})\\
&=\frac{\epsilon_{z1}^{\max}-\epsilon_{z1}^{\min}}{2}\frac{\epsilon_{z2}^{\max}-\epsilon_{z2}^{\min}}{2}\\
&\quad\times\sum_i\sum_j f(\boldsymbol{x}^{*D0}=\mathbf{0},\boldsymbol{x}^{*S0}=\mathbf{0},\boldsymbol{x}^{*D+},\boldsymbol{x}^{*S+}|\epsilon_{z1,i},\epsilon_{z2,j})h_1(\epsilon_{z1,i})h_2(\epsilon_{z2,j})
\end{aligned}
\tag{15}
$$

と計算できる。

今回の MDCEV モデルの場合、観測対象となる個人 n ごとに、実際の旅行回数

$\boldsymbol{x}_n$ を最適解 $\boldsymbol{x}_n^*$ とみなして同時密度を上記の手順により導出する。それらの積

$$\prod_{n=1}^{N} f(\boldsymbol{x}_n^{*D0} = \mathbf{0}, \boldsymbol{x}_n^{*S0} = \mathbf{0}, \boldsymbol{x}_n^{*D+}, \boldsymbol{x}_n^{*S+}) \tag{16}$$

がすべての $\boldsymbol{x}_n^*$ の同時密度となり、これを尤度関数として $\boldsymbol{\theta}$ についての最尤推定を行う。

3　MDCEV モデルの推定に対する並列計算の導入

式 (12) に基づく尤度関数の実際の計算には長い時間がかかる。これは、ヤコビアンおよび数値積分の導出に大量の計算が必要になるためである。

式 (12) にあるように、同時密度の計算ではヤコビ行列 $\boldsymbol{J}$ を計算する。ガウス—ルジャンドル法による数値積分ではすべての積分点 $(\epsilon_{z1}, \epsilon_{z2})$ についてヤコビ行列を計算する必要がある。したがって、積分点の数が増えるほど、ヤコビ行列の計算に必要な時間は増加する。また、ヤコビ行列の要素数は観測対象ごとに異なり、観測対象の旅行回数 $\boldsymbol{x}$ のうち 0 でないものが多いほど（すなわち、内点解の数が多いほど)、ヤコビ行列の要素数が増え計算にかかる時間は増える。さらに、計算したすべてのヤコビ行列について行列式を計算しなければならない。一般には、ヤコビ行列が大きいほど行列式の計算に多くの時間が必要となる。

例えば、積分点の数を 256 とし、ある観測対象について旅行回数 $\boldsymbol{x}$ のうち 0 でないものが 10 個含まれているとしよう。このとき、一つの積分点に対して $10 \times 10 = 100$ 個の要素を持つヤコビ行列を計算することになる。このヤコビ行列を積分点の数だけ計算するため、この観測対象だけでヤコビ行列の要素を全部で $100 \times 256 = 25600$ 個計算することになる。また、計算した 256 個のヤコビ行列について行列式を計算するための時間も必要となる。尤度関数を計算するためには、この計算をすべての観測対象に対して行わなければならない。以上より、すべての観測対象についてヤコビ行列を求めそれらの行列式を求めることは、尤度関数の計算における極めて大きなボトルネックとなることが明らかであろう。

式 (12) の二重積分はガウス求積法やモンテカルロ積分などの数値積分を用いて計算するが、その場合、式 (13) にあるように観測対象ごとに積分点の数だけ条件付き同時密度を計算しなければならない。例えば、後述の実証分析では標本の大きさが 829 であり, 積分点の数を 256 と設定する。このとき、同時密度（尤度関数の値）を求めるためには式 (14) の条件付き同時密度を全部で $829 \times 256 = 211712$ 回

計算する必要がある。また、式 (14) から分かるように、観光地の数 K が増加するほど、条件付き同時密度の計算にかかる時間は増加する。

このように、MDCEV モデルの場合、一回の尤度関数の計算につき大量の計算が要求されるために多大な計算時間が必要となる。準ニュートン法のような最適化法でパラメータ $\boldsymbol{\theta}$ を推定する場合、尤度関数のパラメータによる偏微分を数値近似するならば最適解を探索する過程で尤度関数を繰り返し計算しなければならない。したがって、パラメータの推定を効率的に行うためには尤度関数の計算時間の短縮は不可欠である。

そこで、MDCEV モデルにおける尤度関数の計算に対して、GPU(graphics processing unit) による並列計算を導入し計算効率の向上を試みる。上で述べたように、MDCEV モデルでの尤度の導出には大量の計算が必要となるため、その計算にはコンピュータを用いることが一般的である。現在、コンピュータにおける処理のほとんどは CPU(central processing unit) が担っている。これに対して、GPU は本来コンピュータにおける描画処理を行う演算装置であるが、GPU が多くのコア（演算用のプロセッサ）を持ち大量の並列計算を同時に実行できることから、近年では科学技術計算の分野で GPU による並列計算が導入されている。GPU に含まれるコア数は数百から数千に及び、一般的な CPU のコア数（1 個から 32 個程度）と比べて非常に多い。一方、GPU に含まれるコアは分岐予測などの高度な機能を持たない。したがって、単純な計算を大量に行う場合、GPU による並列計算の導入は CPU のみを用いる場合と比べて計算効率を大きく向上させうる。今回の MDCEV モデルにおける同時密度の計算は、ヤコビ行列の計算、ガンベル分布の密度・累積密度の計算、数値積分といった計算からなる。これらは単純な計算を大量に行うため GPU の導入により計算時間を大きく短縮できると考えられる。

GPU による並列計算を行う場合、現状では GPU の製造者ごとに異なる仕組みを用いる必要がある。例えば、NVIDIA 社製の GPU を使う場合は CUDA と呼ばれる仕組みを用いるのが一般的である。また、AMD 社製 GPU の場合は OpenCL や ROCm などの仕組みを使うことが多い。以下では、NVIDIA 社製の GPU を使い CUDA による並列計算を行うことを前提とする。CUDA により並列計算を効率的に行う場合、ホスト（CPU）―デバイス（GPU）間のデータ転送や GPU 内でのメモリアクセスなどを考慮する必要がある。以下の説明では、これらの技術的な詳細については省略する。CUDA による計算手法について、詳しくは NVIDIA (2019) や Cheng et al. (2015) などを参照されたい。

本研究における実際の尤度関数の計算について概略を以下で示そう。はじめに、

パラメータの値に依存しないデータ・数値を設定する。ガウス—ルジャンドル法の積分点数を設定し、積分範囲が $[-1,1]$ であるときの積分点 (a_i, a_j) とウエイト (w_i, w_j) を計算する。また、各種データから、旅行回数・旅行費用・旅行時間・個人属性・観光地属性などの数値を取得する。さらに、ヤコビアンの計算のために、観光回数が 0 でない旅行先の旅行回数・旅行費用・旅行時間を観測対象ごとに取り出しまとめておく。パラメータ推定において数値微分の計算方法によっては尤度を複数回計算することになるが、上記のデータ・数値はパラメータの値に依存せず推定の過程を通じて不変である。したがって、これらのデータ・数値は推定の最初期に一度だけ設定すればよい。

次に、尤度の計算に必要な各種の数値を求める。数値積分に用いる積分点 $(\epsilon_{z1,i}, \epsilon_{z2,j})$ を設定する。本研究では、σ_{z1} および σ_{z2} の値に基づいて ϵ_{z1} および ϵ_{z2} の所定のパーセント点を計算し、それらを数値積分の下限および上限として用いる。次節の計算では、ϵ_{z1} および ϵ_{z2} について 0.01% 点と 99.99% 点を計算しそれらを二重積分の下限・上限とする。すなわち、ϵ_{z1} の 0.01% 点および 99.99% 点を式 (13) の x_1, x_2 とし、ϵ_{z2} の 0.01% 点および 99.99% 点を式 (13) の y_1, y_2 とする。また、積分点数を 16 として計算した積分点 a_i, a_j から

$$\epsilon_{z1,i} = \frac{x_2 - x_1}{2} a_i + \frac{x_1 + x_2}{2}, \quad \epsilon_{z2,j} = \frac{y_2 - y_1}{2} a_j + \frac{y_1 + y_2}{2}$$

を計算し、それらを組み合わせた $(\epsilon_{z1,i}, \epsilon_{z2,j})$ を実際の積分点とする。この他に、パラメータ $(\boldsymbol{\delta}_D, \boldsymbol{\delta}_S)$ の値に基づいて、$(\boldsymbol{\delta}'_D \boldsymbol{q}, \boldsymbol{\delta}'_S \boldsymbol{q})$ を計算する。以上の値はパラメータの値に依存するため、新たなパラメータの値に基づいて尤度を求めるたびに計算しなければならない。

その後、すべての積分点についてヤコビ行列の各要素を計算しそれらヤコビ行列からヤコビアンを求める。その概略は図 1 の通りである。観光回数が 0 でない旅行先の旅行回数・旅行費用・旅行時間はすでに準備しており $(\boldsymbol{\delta}'_D \boldsymbol{q}, \boldsymbol{\delta}'_S \boldsymbol{q})$ も計算しているため、$\epsilon_{z1,i}, \epsilon_{z2,j}$ の値を与えれば式 (11) によりヤコビ行列を計算できる。したがって、事前に設定したすべての $(\epsilon_{z1}, \epsilon_{z2})$ の組についてヤコビ行列をまとめて並列に計算する。計算したすべてのヤコビ行列からヤコビアンを導出する際も並列計算を導入する。CUDA による開発に必要なライブラリやツールは CUDA Toolkit としてまとめられており、その中に cuBLAS という行列演算ライブラリが含まれている。cuBLAS には、倍精度小数点数からなる複数個の行列に対しまとめて QR 分解を行う cublasDgeqrfBatched 関数が用意されている。この関数を使い、複数のヤコビ行列についてまとめてヤコビアンを計算する。具体的には、複数のヤコビ

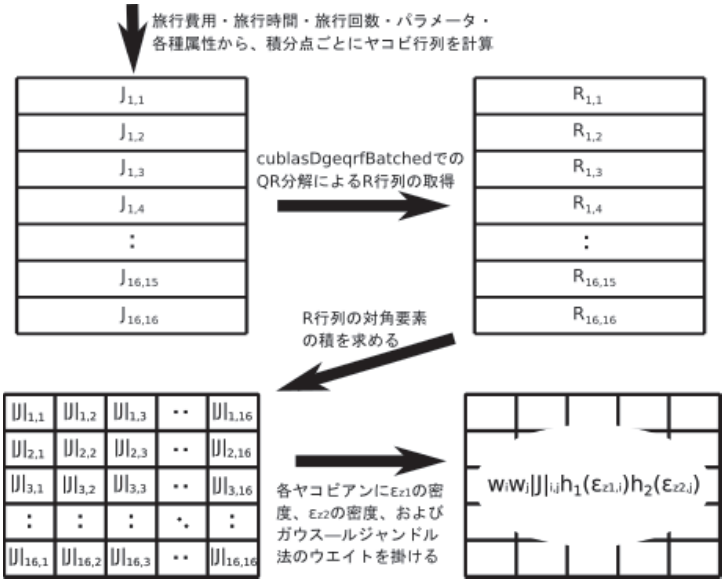

図 1: GPU によるヤコビアンの並列計算

行列を一つの 1 次元配列にまとめ、それを cublasDgeqrfBatched 関数の引数に渡して各ヤコビ行列の R 行列を得る。これら R 行列の対角要素の積を計算しヤコビアンをまとめて計算する。さらに、続く計算のために、計算した各ヤコビアンに $\epsilon_{z1,i}$ の密度、$\epsilon_{z2,j}$ の密度、および、ガウス—ルジャンドル法のウエイト (w_i, w_j) をかけておく。

一方、すべての積分点について条件付き密度 $f(\boldsymbol{x}^{*D0} = \mathbf{0}, \boldsymbol{x}^{*S0} = \mathbf{0}, \boldsymbol{x}^{*D+}, \boldsymbol{x}^{*S+} | \epsilon_{z1}, \epsilon_{z2})$ のヤコビアン以外の項を計算する。積分点ごとに式 (6) の W_k^D と式 (7) の W_k^S を求めた後、以下の P_k または g_k をすべての旅行先に

ついて計算する。

$$P_k = \begin{cases} \exp\left\{-\exp\left(-W_k^D/\sigma_x\right)\right\} \\ \qquad x_k^*\text{が日帰り旅行の回数であり、かつ、端点解} \\ \exp\left\{-\exp\left(-W_k^S/\sigma_x\right)\right\} \\ \qquad x_k^*\text{が宿泊旅行の回数であり、かつ、端点解} \end{cases}$$

$$g_k = \begin{cases} \sigma_x^{-1}\exp\left(-W_k^D/\sigma_x\right)\exp\left\{-\exp\left(-W_k^D/\sigma_x\right)\right\} \\ \qquad x_k^*\text{が日帰り旅行の回数であり、かつ、内点解} \\ \sigma_x^{-1}\exp\left(-W_k^S/\sigma_x\right)\exp\left\{-\exp\left(-W_k^S/\sigma_x\right)\right\} \\ \qquad x_k^*\text{が宿泊旅行の回数であり、かつ、内点解} \end{cases}$$

いま、ϵ_{z1} の積分点数を G_1, ϵ_{z2} の積分点数を G_2, 旅行先の数を $2\times K$ とするならば、この P_k または g_k を一つの観測対象につき $G_1\times G_2\times 2\times K$ 個計算することになる。

最後に、これまでに計算したヤコビアン・ガウス—ルジャンドル法のウエイト・ϵ_{z1} と ϵ_{z2} の密度の積、および、P_k と g_k の値を積分点ごとにすべて掛け合わせ、さらにそれらの総和に $(\epsilon_{z1}^{\max}-\epsilon_{z1}^{\min})/2\times(\epsilon_{z2}^{\max}-\epsilon_{z2}^{\min})/2$ を掛ける。その結果、一つの観測対象について式 (15) の同時密度が得られる（図 2)。

以上の計算をすべての観測対象について行うことで観測対象ごとの同時密度を計算し、それらの対数を合計することで対数尤度を求めることができる。GPU を使って実際に計算を行う場合は、上記の計算をすべての観測対象についてまとめて行うことで効率的な計算が可能となる。

4　実際の計算

前節で説明した計算方法に基づいて、実際のデータから MDCEV モデルの対数尤度を計算する。複数の計算方法および実行環境を設定し、それらのもとでの対数尤度の計算にかかる時間を比較することで計算効率について検討する。

以下では、個人に対するアンケート調査から得られたデータに基づいて MDCEV モデルの尤度を計算する。このアンケート調査は山口大学経済学部部局長裁量経費によるプロジェクト「地域観光振興に関する計量経済学的研究」において実施したもので、株式会社マクロミルによる Web 調査を通じてデータを収集した。このアンケートでは、各個人の年齢・性別・居住地等の個人属性と、2016 年から 2018 年の各年における所得・労働時間・各都道府県への旅行回数等を尋ねた。実際には、

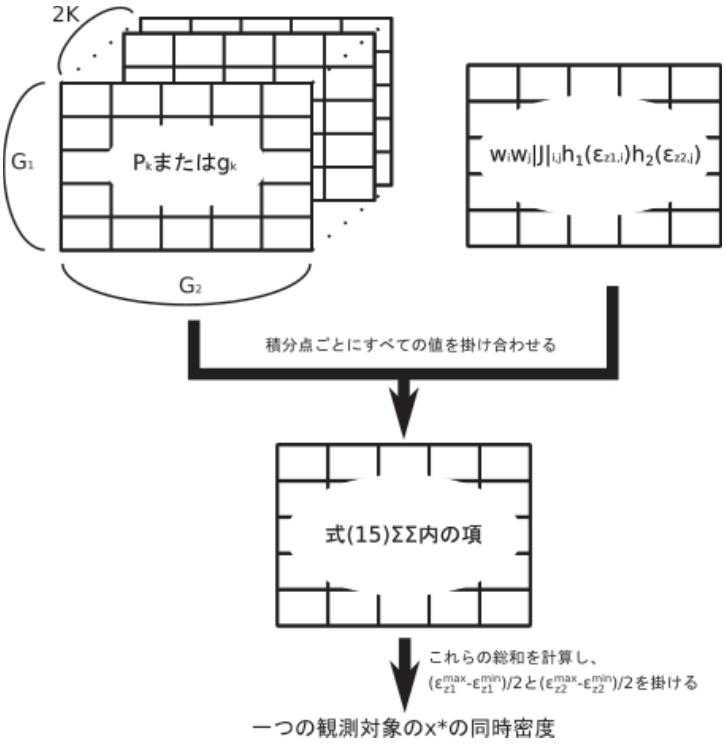

図 2: GPU による同時密度の並列計算

収集したデータから未回答項目や有効でない回答を除いた 829 人分のデータを用いて式 (16) の尤度を計算する。

各都道府県への旅行費用と旅行時間の設定では、株式会社ヴァル研究所の提供する「駅すぱあと Web サービス（スタンダード版）」のデータを用いる。駅すぱあと Web サービスにより、API を通じて二つの駅・港・空港間の経路情報を取得することが可能であり、経路情報には運賃・所要時間が含まれる。以下の計算では、駅すぱあと Web サービスを用いて各個人の居住地と観光先である都道府県との間の往復運賃・所要時間を求め、それらの値をもとに旅行費用・旅行時間を設定する。日帰り旅行の旅行費用 p_k^D はこの往復運賃を使う。日帰り旅行の旅行時間 t_k^D とし

て、「旅行・観光消費動向調査」（観光庁）の定義する国内日帰り旅行の所要時間の8時間と、駅すぱあとWebサービスから得た往復の所要時間とを比較し大きいほうの値を用いる。宿泊旅行の旅行費用p_k^Sは、駅すぱあとWebサービスで得た往復運賃と、居住地―目的地間の一人当たり宿泊費用を足したものを使う。「旅行・観光消費動向調査」（観光庁）における居住地―目的地間の一人当たり旅行消費額（旅行消費額/延べ旅行者数）に対して、全体の旅行消費額に占める宿泊費の割合を掛けたものを一人当たり宿泊費用とし、その2016年から2018年までの平均値を実際の計算に用いる。なお、居住地および目的地は「旅行・観光消費動向調査」の集計区分に従い、北海道・東北・関東・北陸信越・中部・近畿・中国・四国・九州・沖縄の10地域となる。また、宿泊旅行の旅行時間t_k^Sは、駅すぱあとWebサービスで求めた往復所要時間に、「旅行・観光消費動向調査」（観光庁）における目的地別（10地域区分）平均泊数の2016年から2018年までの平均値を足したものを使う。なお、今回の分析では1泊を24時間と換算している。例えば、山口県への宿泊旅行であれば、山口県までの往復運賃に居住地域―中国地方間の一人当たり宿泊費用の平均値を足したものをp_k^Sとし、山口県までの往復所要時間に中国地方での平均泊数の平均値（×24時間）を足したものをt_k^Sとする。

アンケート調査から得た各個人の各都道府県への旅行回数を$\boldsymbol{x}^*$とする。また、z_1^*としてアンケート調査で得た3年間の所得総額から3年間の総旅行費用を引いたものを、z_2^*として3年間の余暇時間から3年間の総旅行時間を引いたものを用いる。なお、アンケート調査の週当たり労働時間をもとに週当たりの余暇時間を$24 \times 7 -$ 週当たり労働時間 として計算し、これを52×3倍したものを3年間の余暇時間の合計としている。総旅行費用はp_k^Dとp_k^Sを各県への日帰りおよび宿泊旅行回数に掛けたものの総和、総旅行時間はt_k^Dとt_k^Sを各県への日帰りおよび宿泊旅行回数に掛けたものの総和としている。個人属性および観光地属性（$\boldsymbol{Q}$）として、年齢・性別・婚姻の有無・同居人数・教育年数・就労状況・自然観光資源数・歴史文化観光資源数・温泉資源数・都市観光資源数・一人当たり県民所得（対数）・第三次産業労働者比率を用いる。「観光資源台帳」（公益財団法人日本交通公社）の都道府県別観光資源数（資源ランクSとAの合計）を表1のように分類し、自然観光資源数・歴史文化観光資源数・温泉資源数・都市観光資源数として用いる。なお、「観光資源台帳」で郷土景観、建造物、集落・街に分類される観光地については、それらの特性に応じて自然観光資源・歴史文化観光資源・都市観光資源に再分類した。

今回の計算では標本の大きさnは829, 観光地の数Kは47, 二重積分の積分点数は先に述べた通り$16 \times 16 = 256$となる。なお、効用関数のパラメータが6個、

表 1: 観光資源の分類

本研究での分類	「観光資源台帳」（公益財団法人日本交通公社）の項目
自然観光資源	河川・峡谷、海岸・岬、岩石・洞窟、 郷土景観（自然観光に属するもの）、湖沼、 高原・湿原・原野、山岳、自然現象、植物、 庭園・公園
歴史文化観光資源	郷土景観（歴史文化観光に属するもの）、 建造物（歴史文化観光に属するもの）、史跡、 集落・街（歴史文化観光に属するもの）、 城跡・城郭・宮殿、神社・寺院・教会、 年中行事（祭り・伝統行事）
温泉資源	温泉
都市観光資源	テーマ公園・テーマ施設、 郷土景観（都市観光に属するもの）、 芸能・興行・イベント、 建造物（都市観光に属するもの）、 集落・街（都市観光に属するもの）、 食、動植物園・水族館、博物館・美術館

$\boldsymbol{\delta}'_D$ と $\boldsymbol{\delta}'_S$ に含まれるパラメータの数がそれぞれ 13 個（定数項を含む）であるため、パラメータの総数は $6 + 13 \times 2 = 32$ 個となる。

計算を行うハードウエア環境として、NVIDIA 社製 Tesla K40c を搭載した GPGPU ワークステーション（環境 1）、同社製 GeForce GTX 1060 を内蔵したノート型パーソナルコンピューター（環境 2）、および、同社製 GeForce RTX 2070 SUPER を Thunderbolt3 経由で接続したモバイル用ノート型パーソナルコンピューター（環境 3）の三つを用意した。各環境の構成は表 2 の通りである。すべての環境で、小数点のデータ型として倍精度小数点型を使う。環境 1 の GPU は科学技術計算やデータセンターでの処理に特化しており、倍精度小数点数の並列計算を高速に行うことが可能である。一方、環境 2 および環境 3 の GPU は画像処理に特化した一般用のもので、環境 1 の GPU と比較して倍精度小数点数の計算は遅い。また、環境 2 の GPU はノート型パーソナルコンピューター向けに調整されて

表 2: 計算に用いる各ハードウエア環境の構成

	環境 1	環境 2	環境 3
CPU	Intel Core i7-5960X(8C/16T)	Intel Core i7-7700HQ(4C/8T)	Intel Core i7-1065G7(4C/8T)
GPU	NVIDIA Tesla K40c	NVIDIA GeForce GTX 1060(3GB)	NVIDIA GeForce RTX 2070 SUPER
メモリ	16GB	16GB	16GB

おり、デスクトップ型パーソナルコンピューター用の同型の GPU と比較して処理能力は低い。環境 3 ではデスクトップ型パーソナルコンピューター用の GPU を使っているが、Thunderbolt3 経由で GPU を外部接続している。このため、GPU を内部接続または内蔵した環境 1・2 と比べて CPU―GPU 間のデータ転送速度は遅い。

上記の各環境で、並列化を行わず CPU のみ使う計算・CPU による並列化を行う計算・GPU による並列化を行う計算の 3 通りの計算を行う。CPU による並列化は、観測対象ごとの同時密度の計算を並列に行うよう実装する。CPU を使った計算では F#というプログラミング言語を使い、Array.Parallel モジュールによる並列化を行う[*5]。また、F#でのベクトル・行列演算を効率的に行うために、Math.NET Numerics という数値計算ライブラリを用いている。一方、GPU による並列化については前節の説明に基づいて実装する。CPU 側の処理では Microsoft Visual C++ を、GPU 側の処理では CUDA C++ を使う。実際には、各種データや数値を初期設定した後、CPU から GPU へパラメータの値のみ転送し、その値に基づいて GPU で尤度を計算した後、尤度を GPU から CPU に転送するという手順をとる。

表 3 は各環境・計算方法のもとで尤度の計算に要した時間であり、尤度の計算を 10 回行った場合の計算時間の平均値を示している。この結果から、どの環境でも

[*5] CPU による並列化については、多くの数値計算ソフトウエアやプログラミング言語でサポートされている。例えば、統計計算ソフトウエア R の場合は parallel という並列計算用のパッケージが標準で導入されている。また、数値計算ソフトウエア Ox ではループ処理を並列化する機能が搭載されている（2019 年時点で有料版のみ）。C++ では、OpenMP などの API や各種ライブラリが整備されており、C#や F#など.NET を基盤とする言語の場合は並列計算を行うためのクラス・モジュールが用意されている。

表 3: 各環境・計算方法における尤度の計算にかかった時間の平均値

	GPU	CPU(並列化なし)	CPU(並列化あり)
環境 1	0.1125 秒	3.6595 秒	1.2419 秒
環境 2	0.2032 秒	3.4141 秒	1.4678 秒
環境 3	0.1527 秒	3.0224 秒	1.5174 秒

GPU を使った計算が最も速いことが明らかである。これらの処理時間は初期設定にかかわる時間を含まず尤度の計算時間のみを測定している。また、GPU での計算時間には CPU—GPU 間のデータ転送の時間を含む。GPU による計算は、並列化を行う CPU による計算と比較して約 1/10 から約 1/7, 並列化を行わない CPU による計算と比較すると約 1/30 から約 1/17 程度計算時間を短縮できている。環境間で比較した場合、GPU の計算では、GPGPU ワークステーションを使う環境 1 が最も速い。これは、環境 1 の GPU が、他の環境の GPU と比べて倍精度小数点数の計算が高速であることを反映している。一方、CPU の計算では、並列化を行う場合だと環境 1 が最も速く、並列化を行わない場合では環境 3 が最も速い。環境 1 の CPU は他の環境と比較してコア数が多いため、並列化の導入による恩恵を強く受けているといえる。

興味深いのは、各環境間で GPU による計算時間の差が比較的小さい点である。表 3 から分かるように、倍精度小数点数の計算が得意な GPU を使った環境 1 と比較して、環境 2 や環境 3 の計算速度が著しく遅いわけではない。その理由については改めて考察する必要があるが、現時点での結論として、たとえ倍精度小数点数の計算が速くない一般用の GPU であっても尤度の計算効率を大きく向上できるといえよう[*6]。また、環境 3 では GPU を外部接続しているにもかかわらず、環境 2 よりも GPU による計算が速い。GPU を外部接続した場合と内蔵した場合とでは、前者のほうが処理効率は低いとされる。これは、GPU を外部接続することによりコンピュータ本体（CPU）と GPU 間のデータ転送にかかる時間が長くなるためである。GPU による計算は、CPU から GPU への入力データの転送・GPU での計算・GPU から CPU への出力データの転送という流れで処理を行うのが一般的

[*6] Tesla K40c のようないわゆるデータセンター用 GPU にはエラー補正など一般用 GPU にはない特徴がある。データセンター用 GPU が持つ特徴の詳細については NVIDIA (2014) などを参照。

である。GPU での計算時間は短いが CPU と GPU との間で頻繁なデータ送受信が発生するような状況（例えばゲームなど）では、一回の処理に占めるデータ転送時間の割合が大きい。この状況では、同じ性能の GPU を外部接続するときと内蔵するときとで前者のほうが処理時間が長くなる。しかし、今回の計算では、GPU での処理に比較的長い時間が必要でありデータ転送にかかる時間は相対的に小さい。また、CPU―GPU 間でやり取りするデータはパラメータの値と尤度のみであり転送するデータサイズも小さい。この場合、GPU での計算時間が全処理時間の大部分を占めることとなり、同性能の GPU を外部接続する場合と内蔵する場合とでその処理時間の差は小さくなる。つまり、今回の計算では、全体的な計算効率は CPU―GPU 間の転送速度よりも GPU の処理能力に強く依存するといえる。環境 3 の GPU は環境 2 よりも高速な処理が可能であるため、全処理時間では環境 3 が環境 2 よりも短くなったと考えられる。

以上の結果に基づき、並列計算の導入可能性について考察しよう。GPU による並列計算を導入する場合、計算の高速化というメリットと、ハードウエア・ソフトウエアの導入にかかるコストというデメリットとを考慮しなければならない。表 3 から分かるように、CPU による並列計算と比較して、GPU の導入は尤度の計算時間を約 1/10 から約 1/7 程度短くすることができる。一方、GPU での並列計算のためには、GPU を導入するだけでなく GPU での計算のための専用の言語・ソフトウエアを使う必要があり、そのためのコストが発生する。2019 年時点で、一般用途の GPU の価格は 5 万円から 20 万円程度、データセンター用 GPU の価格は 100 万円を超えるものもある。また、GPU で計算を行うためには CUDA や OpenCL などの専用の言語・システムを使うのが一般的である[*7]。CUDA および OpenCL 自体は無料で提供されているが、利用のための学習コストを考慮しなければならない。これらのメリットとデメリットとを比較したうえで GPU の導入を考えるべきであろう。尤度の計算にかかる時間がそれほど長くないならば、GPU の導入による計算速度の向上というメリットは比較的小さい。この場合は、GPU を導入せずに CPU による並列化を取り入れるだけでも十分な速度の向上が見込めるだろう。しかし、尤度の計算に多大な時間がかかる場合は GPU の導入による計算時間の大幅な短縮が予想されるため、導入にかかるコストを考慮してもなお GPU を使用することが望ましい可能性が高い。

[*7] 一部の数値計算用ソフトウエアでは、CUDA や OpenCL のラッパーという形で並列計算を行う関数を提供するものもある。

今回の計算で利用した環境のうち、環境 2 および環境 3 は比較的安価な環境である。これらの環境では、CPU のみによる計算効率の向上が難しいため GPU による並列計算を用いる意義は大きい。環境 3 のように一般的なモバイル用ノート型パーソナルコンピューターを使う場合でも、GPU を外部接続することによって計算の高速化を実現することができる。

一方で、CPU の並列計算を導入するだけでも、そうでない場合より約 1/3 から約 1/2 程度の計算時間の短縮を実現した。したがって、複数のマルチコア CPU を並列に接続するような高価な環境が利用できるならば、CPU による並列化でも高速な計算が可能であろう。

表 3 で示した計算時間は、MDCEV モデルのパラメータ推定に GPU を導入する際に計算時間をどの程度短くできるかの参考になるだろう。例えば、MDCEV モデルを最尤法で推定する際、最適化の手法として準ニュートン法を使い、尤度関数のパラメータによる偏微分を数値的に計算するとしよう。MDCEV モデルでは尤度の計算にかかる時間が長いために、全体の計算に対して尤度の計算が占める割合が大きい。その結果、パラメータ推定に必要な計算時間の GPU と CPU の比は、尤度の計算にかかる時間の比に近い値となるだろう。表 3 では、GPU と CPU（並列化あり）の計測時間の比が約 1/10 から約 1/7 程度である。そのため、CPU による並列計算から GPU での並列計算に移行することで、パラメータ推定にかかる時間も同程度の割合で短縮できるだろう。

本研究で示した計算方法やその実装には様々な改善点が存在しうる。これらについて今後も研究を継続する。また、上記のモデルおよびデータに基づく実際のパラメータ推定、および、旅行費用・旅行時間・各種属性の変化が旅行回数に与える影響のシミュレーションについては、稿を改めて考察したい。

■謝辞　本研究は、山口大学経済学部部局長裁量経費による研究プロジェクト「地域観光振興に関する計量経済学的研究」（研究代表者：諏訪 竜夫先生、共同研究者：野村 淳一先生、齋藤 英智先生、米岡 秀眞先生、福井 昭吾）における成果である。プロジェクトメンバーの先生方には、多くの有益なご助言をいただいた。また、プロジェクト代表者である諏訪 竜夫先生には、モデル構築および分析手法についての討論や各種データの収集などでご助力いただいた。ここに感謝の意を表する。

参考文献

Bhat, C. R. (2005) “A Multiple Discrete-Continuous Extreme Value Model: Formulation and Application to Discretionary Time-Use Decisions,” *Transportation Research Part B: Methodological*, Vol. 39, No. 8, pp. 679-707.

——— (2008) “The Multiple Discrete-Continuous Extreme Value (MDCEV) Model: Role of Utility Function Parameters, Identification Considerations, and Model Extensions,” *Transportation Research Part B: Methodological*, Vol. 42, No. 3, pp. 274-303.

Fang, H. A. (2008) “A Discrete-Continuous Model of Households’ Vehicle Choice and Usage, with an Application to the Effects of Residential Density,” *Transportation Research Part B: Methodological*, Vol. 42, No. 9, pp. 736-758.

Kuriyama, K., Y. Shoji, and T. Tsuge (2017) “The Value of Leisure Time of Weekends and Long Holidays: The Multiple Discrete-Continuous Extreme Value (MDCEV) Choice Model with Triple Constraints,” Forest Policy and Economics Working Paper 1701, Division of Natural Resource Economics, Kyoto University.

NVIDIA (2014) “Tesla K80 Datasheet,” URL: `https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/TeslaK80-datasheet.pdf`, accessed on 14th January 2020.

——— (2019) “CUDA Toolkit Documentation,” URL: `http://docs.nvidia.com/cuda/`, accessed on 27th January 2020.

Wales, T. J. and A. D. Woodland (1983) “Estimation of Consumer Demand Systems with Binding Non-Negativity Constraints,” *Journal of Econometrics*, Vol. 21, No. 3, pp. 263-285.

Cheng, John, Max Grossman, and Ty McKercher (2015)『CUDA C プロフェッショナルプログラミング』，森野慎也訳，インプレス．

A. C. チャン (1996)『現代経済学の数学基礎（下）』，大住栄治・高森寛・小田正雄・堀江義訳，シーエーピー出版．

福田大輔・力石真 (2013)「離散―連続モデルの研究動向に関するレビュー」，『土木

学会論文集 D3（土木計画学）』，第 69 巻，第 5 号，497-510 頁.