

項目反応理論を用いた受験者 およびテストの評価

熊本大学大学院自然科学教育部博士前期課程
機械数理工学専攻数理工学教育プログラム
210D8551 西郷 虎太郎

序文

現代において何をするにも評価というものは付きものである。その評価に関する一つの測定方法として項目反応理論が存在している。これは従来の採点、評価方法とは大きく違うものであり見えない潜在的なところまで測定できるかもしれないというものである。主に TOEIC や共通試験といった大規模で行われる試験の採点に用いられることが多い。まずは、従来の採点方法との違いを説明する。従来のテストは、簡潔に言うと先に配点が決められておりそれをもとに合計得点が決まるといったものだ。しかし、これでは次のような課題に直面する。試験の点数が同じ2人を考える。一方は難易度の低い問題ばかりを正解している。他方は、難易度が低い問題はもちろん、難易度が高い問題も含めて正解している。この場合、2人の能力は等しいと言えるだろうか。また、次のような場合も考えられる。全国模試の学校平均が昨年より上がった学校がある。果たしてこの学校の学力が上がったと言えるだろうか。もしかしたら問題が簡単になっただけかもしれない。このような課題を、解決するために項目反応理論が用いられる。その理論にも続いて、多次元化を目指したものである。以下は各章の内容を紹介している。

第1章では、テストとは何か、から始まり受験者に対する従来の評価方法の問題点を上げたうえで、それを解決するために用いる項目反応理論の基礎を紹介している。その際に用いる主な4つのパラメータ、識別力 a 、困難度 b 、偶然率 c 、能力値 θ についての説明も行っている。それらからなる項目特性関数は従来の評価の課題を克服していることを説明している。

また、一般に知られている正規累積モデルやロジスティックモデルを用いた理論の紹介も行っている。使用するパラメータによってモデル式が1母数ロジスティックモデルから4母数ロジスティックモデルまで分かれる。現実でテストの評価、能力値の推定を行うことを考えると3母数ロジスティックモデルを使用することが多い、ということを紹介したうえで、第3章での実際の推定の話に繋がりをやすくしている。

第2章では、従来の評価方法の一つにある項目特性図を用いて、実際にテストを実施し受験者の評価を試みた。項目特性図とは、受験者を点数の高い順に一定のグループに分け、その中で各問題の正答率を観察するものである。項目特性図における型の分類を紹介したうえで、実際に行ったテストの項目特性図を観察し、考察を行っている。第3章では、最尤推定法やベイズ推定法の推定値と項目特性図を関連付けた考察も行っている。

第3章では、実際に各パラメータの推定はどのように行っていくのかを紹介する。まずは、最尤推定法、ベイズ推定法などの推定法の基礎を紹介したうえで、さらに細かく周変化や同時推定を用いた推定法を考える。最尤推定法は、あるデータが得られる確率が最も高くなる各パラメータを推定する方法である。最尤推定法の長所はある程度のデータが集まると推定値の信頼性が上がる場所であり、短所は逆にデータが少ないと推定値の信頼度が下がる場所にある。

また、ベイズ推定は、ベイズの定理の考え方にに基づき、ある事象が起こる確率の分布と観測されたデータから不確定的な事象を推定する方法である。この推定法の

長所は最尤推定法とは逆に少ないデータでもある程度信頼できる推定値を得られることにある。しかし、短所は使用する確率分布の信頼性に依存するところにある。

この章の最後には、第2章で用いたデータを実際に、最尤推定法とベイズ推定法を用いて改めて観察した。推定した後に、項目特性図との比較も行っている。また、このテストの受験者が30人と少ないことを利用して、最尤推定法とベイズ推定法の推定値の比較も行っている。結果としては、この規模の受験者数ではベイズ推定法の方が安定して推定値を求めることができた。

第4章について、ここまでの章では、受験者の能力値(テストにおいては頭の良さ)は、単一的なものとして考えている。しかし、実際には単一的でないと考えられる。例えば、数学において頭の良さとは、「計算が早い」、「論理的な思考が得意」など複数の構成要素が上げられる。この章では、そのことを踏まえて、第1章で用いたロジスティックモデルの能力値 θ を多次元化したモデルを考えている。今まで使っていたモデルをうまく拡張することで、多次元化されたロジスティックモデルでも今までと変わらず推定できるようにしている。

最後に第5章では、多次元ロジスティックモデルを使用した推定のシミュレーションを行っている。より実際のテストの運用に近づけるために、テスト作成時は過去に出題された問題を使用することを考える。過去に出現されている問題は各パラメータが既知であると仮定すると未知の場合に比べて、受験者の能力値の推定値が安定するのかなどなどの考察を行っている。これを考えることはテストにおいて過去問題がいくらかの割合で出題されている TOEIC などの試験の運用の裏付けになると考えている。

目次

1	はじめに	1
1.1	項目特性関数	2
1.2	正規累積モデル	2
1.3	ロジスティックモデル	3
2	従来のテストの評価方法	6
2.1	項目特性図	6
2.2	正答分析	7
3	能力値や項目パラメータの推定	9
3.1	最尤推定法	9
3.2	同時最尤推定法	10
3.3	周辺最尤推定法	11
3.4	ベイズ推定法	12
3.4.1	ベイズの定理	13
3.4.2	EAP 法 (Expected a posteriori 法)	14
3.5	推定値と考察	15
3.5.1	ベイズ推定	15
3.5.2	最尤推定法との比較	18
4	ロジスティックモデルの多次元化	19
5	多次元ロジスティックモデルを用いた能力値推定のための問題選定 (仮)	20
5.1	項目パラメータと能力値の推定値	22
6	最後に	23
	参考文献	23
	付録 1 第 2 章で使用了問題	23
	付録 2 推定に使用了プログラム	25

1 はじめに

この章では、準備として項目反応理論を述べるうえで必要となる用語や記号について紹介していく。

テストとは、多数の問題から構成されるものである。テスト以外のことも考えることがあるため以下より項目と示す。項目は、テストを構成する最小の単位である。受験者 $i (i = 1, 2, 3, \dots, N)$ の項目 $j (j = 1, 2, 3, \dots, n)$ に対する反応を正答を 1、誤答を 0 として u_{ij} として表す。ここで文字 i は受験者の添え字、 j は項目の添え字である。式にすると以下ようになる。

$$u_{ij} = \begin{cases} 1 & (\text{正解のとき}) \\ 0 & (\text{正解でないとき}) \end{cases}$$

この u_{ij} を (i, j) 成分にもつ $N \times n$ 行列 U を反応行列という。式にすると

$$U = \begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & & \vdots \\ u_{N1} & \cdots & u_{Nn} \end{pmatrix}$$

のようになる。

ここで受験者 i の反応、つまり、第 i 行だけを抜き出したベクトルを \mathbf{u}_i とし、受験者 i の反応データとする。項目反応理論では、受験者の反応行列から項目と受験者の両方の特性を評価することを目指す。受験者の学力を表すパラメータを θ で表す。さらに、受験者 i の能力値は θ_i で表す。学力以外にも活用する場合を考えて、今後は受験者の能力値と表現する。一方で、各項目の性質を表すパラメータとして、識別力、困難度、偶然率などを導入する。このパラメータを項目パラメータと表現する。以下に示すのは各項目パラメータの説明である。

識別力 推定したい受験者の能力値 θ をその項目でどのくらい正確に推定できているのかを表したパラメータである。以下より、項目 j の識別力を a_j で表す。

困難度 項目 j の難しさを表すパラメータであり、 b_j で表現する。2 に近いほど能力値 θ が高くないと正答できない難しい項目とされる。

偶然率 能力値 θ に関係なく偶然正答になる確率を表すパラメータである。一般的には、当て推量と呼ばれることもあるが以下では、偶然率という言葉を用いている。項目 j の偶然率を c_j で表す。例えば 4 択の選択問題ならばランダムに回答すると $\frac{1}{4}$ で正答すると考えられるので $c_j = \frac{1}{4}$ と考えることができる。しかし実際には、ほぼ選ばれることのない選択肢があることを考えると $\frac{1}{4}$ より大きいと考えることもできる。

1.1 項目特性関数

テストの性質は、項目の性質に依存する。各項目に対する受験者群ごとの正答率は、縦軸に正答確率、横軸にテスト得点を記述した項目特性図によって示される。しかし、項目特性図は受験者の分布に大きく依存しているという欠点がある。(群の分け方で結果が変わることがあるということ) これでは実社会におけるテストの運用には耐えることができない。

TOEIC を例に考える。年に数回開催され、世界中で実施される。同一の人物が複数回受験する可能性があるため、同じ項目は使用できても、同じ項目の集まりで構築されたテストは2度と使用できない。このため、テストの平均点や各項目に対する正解率はテストごとに毎回異なる。開催される地域によって受験者の分布も異なり、地域ごとの英語力も異なってくる。したがって、前述した項目分析や単純な偏差値などでは、受験者全体での相対的評価をすることが難しい。

上記の困難を克服するために、項目反応理論 (Item Response Theory) の考え方を導入する (以下より IRT と記載)。これは受験者の反応行列 U から、最も可能性の高い能力値 θ を推定する、ということである。この能力値 θ は、直接観測はできない潜在特性と呼ばれ受験者群に依存しないものである。受験者の真の能力とも言える。能力値 θ の受験者が項目 j に正解する確率を縦軸に、そして横軸に能力値 θ を配する関数、項目特性関数 (Item Characteristic Function) を使う。その準備としてまずは正規累積モデルについて考える。

1.2 正規累積モデル

項目特性図の弱点として、受験者集団の能力値 θ の分布に依存するというものがある。人数で群に分けているため、同点の受験者をどちらの群に入れるかなどで結果が変わることがある。これを克服するためには、受験者集団に依存しない能力値 θ を導入する必要がある。以下では、(潜在的) 能力値とする。横軸に能力値 θ 、縦軸に正答確率を配する。このとき、横軸の能力値 θ は直接観測できない値であることから、関数形は具体的に定まるものではない。そこで、標準正規分布の累積分布関数を利用して

$$\Phi(f(\theta)) = \int_{-\infty}^{f(\theta)} \phi(z) dz \quad (1.2.1)$$

と置ける。このように横軸に能力値 θ 、縦軸に項目への正答確率を配した関数を項目特性関数とする。また上記の式を特に正規累積モデルという。このとき、 $f(\theta)$ は θ に関する単調増加関数を選ぶと $\Phi(f(\theta))$ も単調増加関数になる。

1 母数正規累積モデル 項目 j の性質として重要な一つがどのくらい難しいのかである。この性質を表現するために、 θ の関数を

$$f(\theta) = a(\theta - b_j) \quad (1.2.2)$$

として項目 j の項目特性関数を

$$p_j(\theta) = \Phi(a(\theta - b_j)) \quad (1.2.3)$$

とする。ここで b_j は項目の困難度、つまり難しさを表している。これが項目特性関数にはグラフが右寄り (難しい) か左寄り (簡単) かどうかで現れる。また a は項目に共通している値であるため、 b_j の単一の値によって項目の性質が決まる。このモデルを 1 母数正規累積モデルという。

1.3 ロジスティックモデル

正規累積モデルでは式中に積分を含んでいる。そのため実際の推定の際に計算が複雑になってしまう。そこで、ロジスティック分布の分布関数を利用した項目特性関数が提案された。提案されたモデルは

$$\int_{-\infty}^{f(\theta)} \phi(z) dz \simeq \frac{1}{1 + \exp(-D \times f(\theta))} \quad (1.3.1)$$

のように近似を利用するモデルであった。この式の右辺をロジスティックモデルという。ただし、

$$D = 1.7 \quad (1.3.2)$$

のときに近似の差が 0.01 以下になることが知られていて、以下より $D = 1.7$ として用いる。

1 母数ロジスティクスモデル 能力 θ をもつ人が、ある項目 j に正解する確率を $p_j(\theta)$ と表現する。この正答確率 $p_j(\theta)$ を複数のパラメータで表すと次のようになる。

$$p_j(\theta) = \frac{1}{1 + \exp(-Da(\theta - b_j))} \quad (1.3.3)$$

これを 1 パラメータロジスティックモデルという。

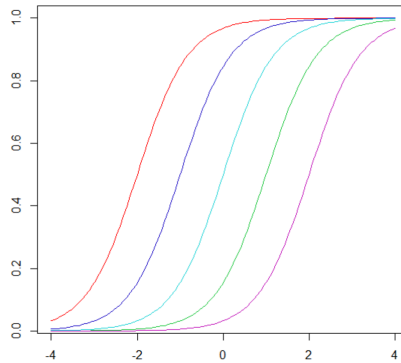


図 1: a は 1 で固定、 b は $-2, -1, 0, 1, 2$

a は既知の場合、未知の場合を含めて全ての項目に共通している値であるため、項目特性関数は項目同士で交わることはない。項目特性関数によって表される確率 $p_j(\theta)$ は、特定の受験者の正答確率ではない。例えば $\theta = 0.0$ の受験者の母集団を考えたときの正答確率を表している。これが、従来の考え方との大きな違いである。また、困難度が高くなるほど、グラフは右に移動していく。

2 母数ロジスティックモデル a_j と b_j を使って表した式を 2 母数ロジスティックモデルという。式にすると以下ようになる。

$$p_j(\theta) = \frac{1}{1 + \exp(-Da_j(\theta - b_j))} \quad (1.3.4)$$

ここで、 a_j は基本的に識別力は正の値だけを扱う。数学的には $-\infty$ から ∞ で定義されるが、 a_j が負の値をとると、

$$f(\theta) = a_j(\theta - b_j) \quad (1.3.5)$$

が単調減少関数になる。これは、能力値 θ が高いほど正答確率が低くなることを意味しているが、そのような項目は項目自身に問題があるとして判断され、削除されるからである。また、困難度 b について、能力値 θ の高い受験者ばかりだと正答率は上がるため、項目の困難度 b_j は受験者の能力値 θ_i によって変化することがあると言える。従って 1 パラメータロジスティックモデルのときとは違い、項目特性関数は項目同士で交わることがある。下図のように、 a の値が大きくなるほどグラフの傾きが大きくなる。つまり、識別力 a の値が大きくなると、能力値 θ の値の少しの変化でも、大きく正答確率が変わることになる。このことから、パラメータ a は識別力を表すパラメータを表していることが分かる。

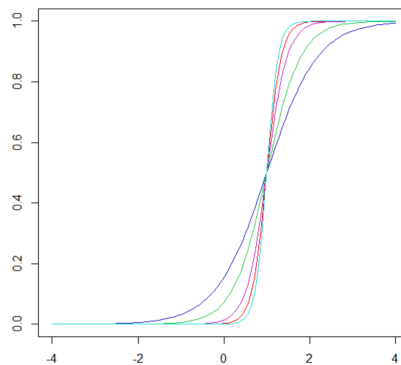


図 2: b は 1 で固定、 a は $-4, -2, 0, 2, 4$

3 母数ロジスティックモデル a_j と b_j と c_j を使って表した式を 3 母数ロジスティックモデルという。式にすると以下ようになる。

$$p_j(\theta) = c_j + \frac{1 - c_j}{1 + \exp(-Da_j(\theta - b_j))} \quad (1.3.6)$$

2 母数ロジスティックモデルに加えて、偶然率 c_j が追加されたモデルである。これは、下方漸近線といわれることもある。どんなに能力値 θ の低い受験者でも偶然正答することを考えると、正答確率は0にはならないので結果、項目特性関数の各項目の正答率は c_j から 1 の間で推移する。下図のように c の値が大きくなるほど、グラフの下限が上がっていく。

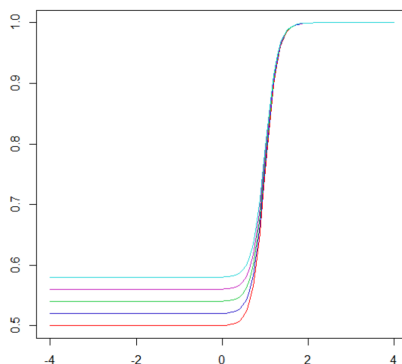


図 3: a は 4 で固定、 b は 1 で固定、 c は 0.58, 0.56, 0.54, 0.52, 0.5

1 母数ロジスティックモデルでは、すべての項目の識別力は等しく、かつ偶然率はないものと仮定している。2 母数ロジスティックモデルでは、すべての項目に偶然率はないものと仮定している。したがって、モデルの表現力という観点では 3 母数ロジスティックモデルが優れているといえる。しかしながら、1 母数ロジスティックモデルには多くの受験者を必要としないのに対して、3 母数ロジスティックモデルでは多くの受験者を必要とする。常に、3 母数ロジスティックモデルが優れているとは言い切ることができない。

4 母数ロジスティックモデル a_j と b_j と c_j と d_j を使って表した式を 4 母数ロジスティックモデルという。式にすると以下になる。

$$p_j(\theta) = c_j + \frac{d_j - c_j}{1 + \exp(-Da_j(\theta - b_j))} \quad (1.3.7)$$

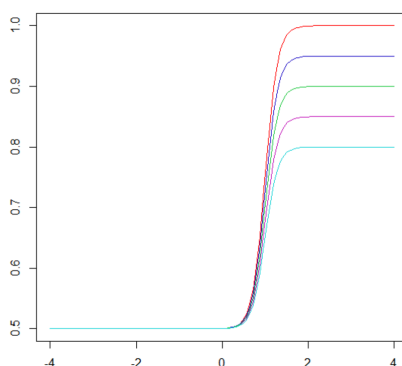


図 4: a は 4 で固定、 b は 1 で固定、 c は $\frac{1}{2}$ 、 d は 1, 0.95, 0.9, 0.85, 0.8

項目の中には、どんなに能力値 θ の高い受験者がいたとしても正答確率が 1.0 にならないものもあるだろうとして、そのような状況を取り扱う。パラメータ d_j は、上方漸近線を表しており、項目特性関数では d_j 以上の値をとらない。ただし、これは実用的に使われた実績はあまりない。

識別力 a_j について、1 パラメータロジスティックモデルのときは 1 とされるので、基本的には能力値 θ の低い受験者には常に難しく、能力値 θ の高い受験者には常に易しい項目だと仮定されている。どの、得点群においても正答率がおなじである問題があることを考えると、実際のテストにおいては 2 パラメータロジスティックモデルで推定することも困難であることが分かる。偶然正解することを考えると一般的には 3 パラメータロジスティックモデルで推定することが多くなる。

2 従来のテストの評価方法

2.1 項目特性図

従来の分析方法として項目特性図を用いた正答分析と誤答分析がある。項目特性図とは、(流儀は多々あるが今回は) 横軸にテストの得点ごとに分けられた群番号、縦軸には群ごとの正答率を付したグラフである。グラフの形によっていくつかの型に分けられた。ここで、テストの得点で群に分けるとは合計得点をもとに高いものから第 5 群、低い者たちを第 1 群とし振り分けることである。型については以下の通りである。

型の分類

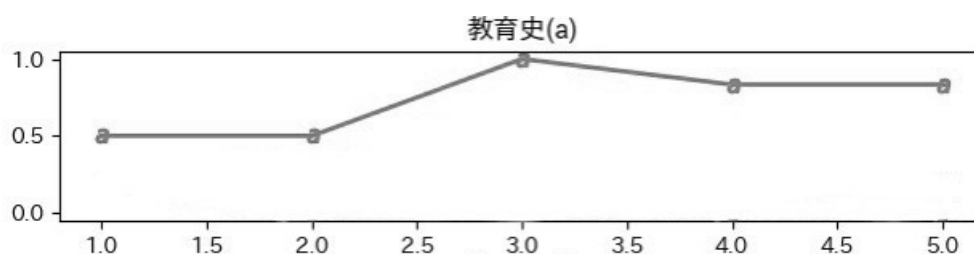
- G 型 **高識別力項目** 最下位群から最上位群にかけて急激に正答率が上がっている項目
- L 型 **下位識別力項目** 下位群は正答率が非常に低く、そこから中位群にむけて正答率が高くなり、それ以降は高い正答率で安定する項目
- H 型 **上位識別力項目** 中位群までは安定的に正答率が低く、そこから正答率が急激に上昇する項目
- E 型 **高通過率項目** 正答率のグラフが高い値から始まり全体的に高い値で張り付いている項目
- M 型 **中通過率項目** テスト得点の高い者、低い者も正答率が中間の値を取っている変わらない項目
- D 型 **低通過率項目** 正答率が全体的に低い値に張り付いている項目
- B 型 **右下がり項目** 上位群より下位群の方が正答率が高い項目

実際に項目特性図を使って項目の分析をする。

- 実験概要… 教職教養に関する問題を8問出題し、その反応を項目特性図によって、各選択肢をみて正答分析を行ってみる。
- 条件設定… 教職教養に関する問題を教育学部に在籍しているもしくは在籍していたもの30人を対象に行った。教職教養の勉強に広く用いられる時事通信社の「教職教養の演習問題」から8題出題する。問題は巻末に掲載する。第1問、第2問は教育史に関する問題、第3問は教育原理に関する問題、第4問、第5問はは教育法規に関する問題、第6問、第7問、第8問は教育心理に関する問題が出題されている。

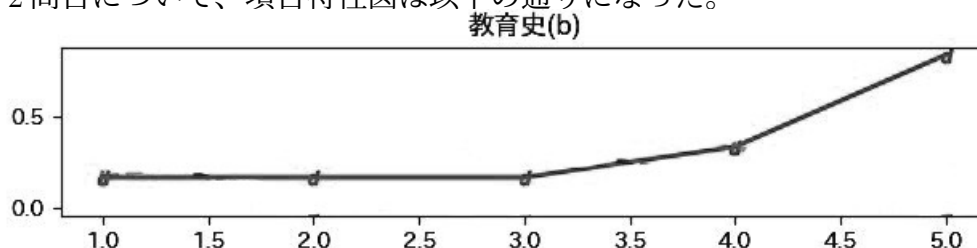
2.2 正答分析

○ 第1問目について、項目特性図は以下の通りになった。



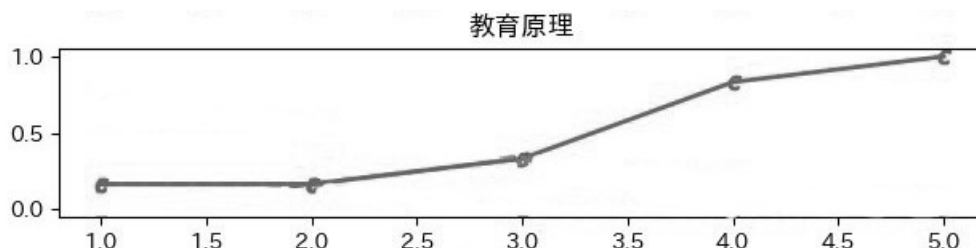
正答率は中位群までは上昇していて、上位群では多少差はあるが高い正答率を有している。下位識別力項目に近い項目特性図をしている。

○ 第2問目について、項目特性図は以下の通りになった。



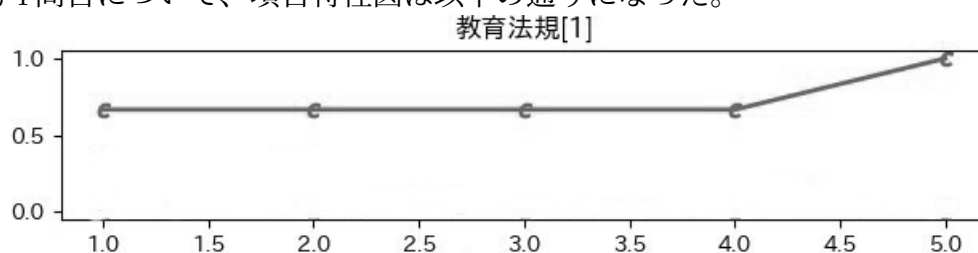
下位群から中位群までは正答率は低い、そこから上位群にかけて急激に上昇している。1問目より下位識別力項目の項目特性図をしている。

○ 第3問目について、項目特性図は以下の通りになった。



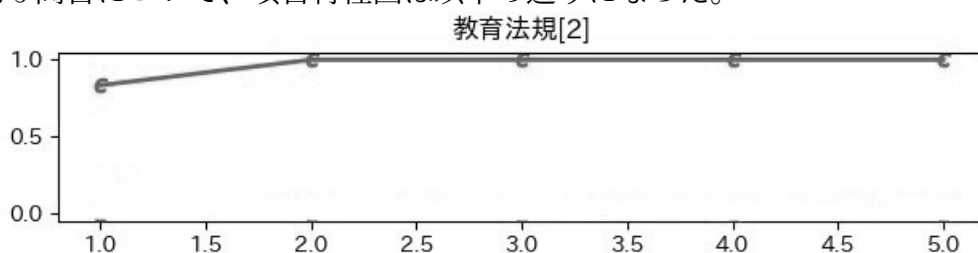
下位群から上位群に行くにつれて順当に正答率が上がっている。項目特性図は高識別力項目と言える。

○ 第4問目について、項目特性図は以下の通りになった。



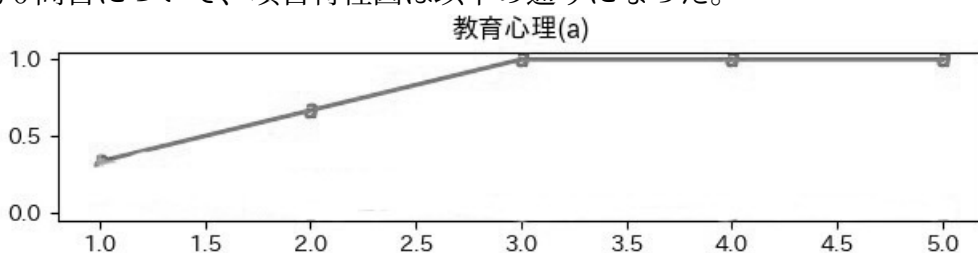
正答率はどの群も安定して6割を超えて最上位群は正答率が1である。項目特性図は高通過率項目に近い。

○ 第5問目について、項目特性図は以下の通りになった。



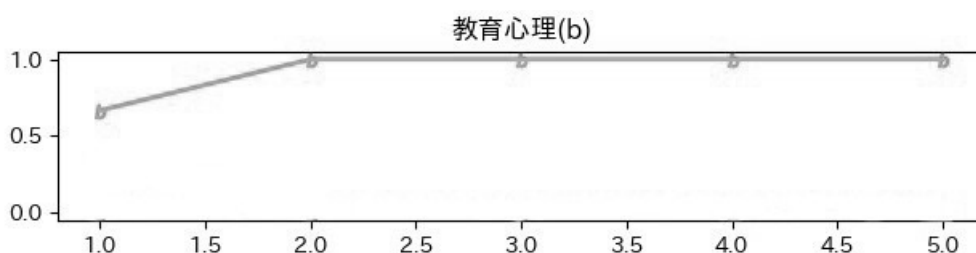
正答率は全体的に高位に張り付いている。4問目より高通過率項目である。

○ 第6問目について、項目特性図は以下の通りになった。



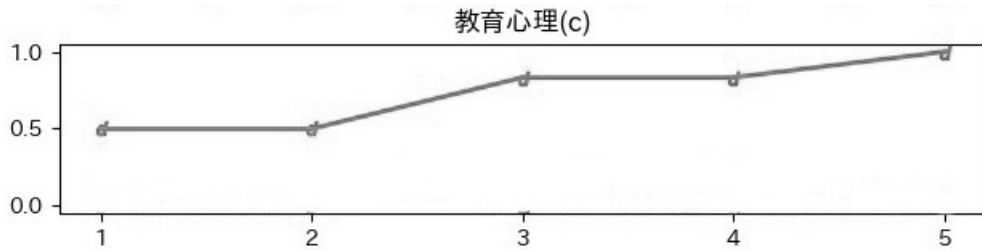
正答率は下位群から中位群にかけては上昇しており、上位群では安定した正答率を有している。型としては、下位識別力項目である。

○ 第7問目について、項目特性図は以下の通りになった。



5問目と似ていて正答率は全体的に高位に張り付いている。高通過率項目である。

○ 第8問目について、項目特性図は以下の通りになった。



下位群から上位群に行くにつれて順当に正答率が上がっている。項目特性図は高識別力項目と言える。

3 能力値や項目パラメータの推定

3.1 最尤推定法

ここでは項目パラメータが既知の場合に、項目パラメータと反応行列から能力値 θ の推定値を求める方法を紹介する。

受験者 i の能力値 θ_i は現実においては未知である。一方で、反応行列は収集して実際の値として固定し用いることができる。そこで、反応行列を既知として、能力値 θ_i を変数として扱うことにする。このとき $p_j(\mathbf{u}_i|\theta_i)$ に対して、 $L(\theta_i|\mathbf{u}_i)$ を確率ではなく尤度と呼び、その尤度関数を、

$$L(\theta_i|\mathbf{u}_i) = \prod_{j=1}^n p_j(\theta_i)^{u_{ij}} (1 - p_j(\theta_i))^{1-u_{ij}} \quad (3.1.1)$$

と定義する。このとき、尤度が最大になるときの項目パラメータの値を、パラメータの推定値として利用する。このような推定法を最尤推定法という。つまり、この推定法は実際のデータが最も得られやすいようなパラメータの値を決める方法である。

また、最尤推定値の性質として以下のことがあげられる。

- 受験者 i の反応ベクトル $\mathbf{u}_i = (u_{i1}, u_{i2}, \dots, u_{in})$ から計算される最尤推定値は同じタイプの母数であっても、一般に1母数、2母数、3母数モデルでそれぞれに値が異なる。項目特性関数が異なるため、尤度関数の形状も異なるので最大値が一致しないのは自然である。
- 全問正答、全問誤答の最尤値はそれぞれ $\theta_i = \infty, \theta_i = -\infty$ となり事実上は解が求まらない。

次に、具体的に式 (3.1.1) を最大化する $\hat{\theta}_i$ を考える。そこで、式 (3.1.1) の対数をとる、

$$\log L(\theta_i | u_i) = \sum_{j=1}^n [u_{ij} \log p_j(\theta_i) + (1 - u_{ij}) \log q_j(\theta_i)] \quad (3.1.2)$$

を考える。この関数を対数尤度関数という。この対数尤度関数最大にする $\hat{\theta}_i$ を求める解法が考えられる。

3.2 同時最尤推定法

前節では、項目パラメータは既知である場合について考えてきた。ここでは、受験者たちの反応行列 U から受験者の能力値 θ と項目パラメータを同時に推定する方法を考える。受験者 i の能力値 θ_i が与えられたときの θ_i の尤度の分布は

$$L(\theta_i, a_j, b_j, c_j | u_{ij}) = p_j(\theta_i)^{u_{ij}} (1 - p_j(\theta_i))^{1-u_{ij}} \quad (3.2.1)$$

のように定義される。ここで項目同士は局所独立を考えている。局所独立とは、項目同士が互いに正答率に影響していない状態のことである。本節では、能力値 θ だけではなく、項目パラメータも未知であることを仮定しているので、 θ_i の尤度の分布を以下のように書き直す。

$$L(\theta_i, \mathbf{a}, \mathbf{b}, \mathbf{c} | u_i) = \prod_{j=1}^n p_j(\theta_i)^{u_{ij}} (1 - p_j(\theta_i))^{1-u_{ij}} \quad (3.2.2)$$

となる。3 母数ロジスティックモデルを用いて考える。 n 個の長方形の和によって $L(\theta_i, a_j, b_j, c_j | u_{ij})$ の値を近似することができる。ここで、

$$\begin{aligned} \mathbf{a} &= (a_1, a_2, \dots, a_n)^t \\ \mathbf{b} &= (b_1, b_2, \dots, b_n)^t \\ \mathbf{c} &= (c_1, c_2, \dots, c_n)^t \\ \boldsymbol{\theta} &= (\theta_1, \theta_2, \dots, \theta_N)^t \end{aligned}$$

とし、受験者の反応は互いに独立と考えられることから、能力値 θ と項目パラメータが与えられた条件の下で、 n 個の項目に対する N 人の受験者の反応行列 U が観察される尤度は、

$$\begin{aligned} L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | U) &= \prod_{i=1}^N L(\theta_i, \mathbf{a}, \mathbf{b}, \mathbf{c} | u_i) \\ &= \prod_{i=1}^N \prod_{j=1}^n L(\theta_i, a_j, b_j, c_j | u_{ij}) \end{aligned} \quad (3.2.3)$$

と表現される。この尤度を最大にする各項目パラメータの値のベクトルが最尤推定値である。またこのような方法が最尤推定法であった。ここで最大値を求めるために、対数変換を行うと

$$\begin{aligned}
\log L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | \mathbf{U}) &= \log \prod_{i=1}^N \prod_{j=1}^n L(\theta_i, a_j, b_j, c_j | u_{ij}) \\
&= \log \prod_{i=1}^N \prod_{j=1}^n p_j(\theta_i)^{u_{ij}} (1 - p_j(\theta_i))^{1-u_{ij}} \\
&= \sum_{i=1}^N \sum_{j=1}^n [u_{ij} \log p_j(\theta_i) + (1 - u_{ij}) \log q_j(\theta_i)] \quad (3.2.4)
\end{aligned}$$

となる。この関数の最大化を考える。尤度関数を最大化するパラメータの値と、対数尤度関数を最大化するパラメータの値は等しい。対数尤度関数を最大化する解は、変数で微分して0とおいた方程式を変数に関して解くことによって求まる。変数が複数あるので、各変数 θ_i, a_j, b_j, c_j で偏微分した結果を0とおき、

$$\begin{aligned}
\frac{\partial}{\partial \theta_i} \log L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | \mathbf{U}) &= \frac{\partial}{\partial \theta_i} \sum_{i=1}^N \sum_{j=1}^n [u_{ij} \log p_j(\theta_i) + (1 - u_{ij}) \log(1 - p_j(\theta_i))] \\
&= D \sum_{j=1}^n \left\{ u_{ij} \left\{ (p_j(\theta_i) - c_j) \frac{-a_j \exp(a_j(\theta_i - b_j))}{1 + c_j \exp(a_j(\theta_i - b_j))} \right\} \right. \\
&\quad \left. + (1 - u_{ij}) \frac{a_j(p_j(\theta_i) - c_j)}{c_j \exp(a_j(\theta_i - b_j))} \right\} \\
&= D \sum_{j=1}^n \frac{a_j(p_j(\theta_i) - c_j)(u_{ij} - p_j(\theta_i))}{(1 - c_j)p_j(\theta_i)} = 0 \quad (3.2.5)
\end{aligned}$$

$$\frac{\partial}{\partial a_j} \log L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | \mathbf{U}) = \frac{D}{1 - c_j} \sum_{i=1}^N \frac{(\theta_i - b_j)(p_j(\theta_i) - c_j)(u_{ij} - p_j(\theta_i))}{p_j(\theta_i)} = 0 \quad (3.2.6)$$

$$\frac{\partial}{\partial b_j} \log L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | \mathbf{U}) = \frac{-Da_j}{1 - c_j} \sum_{i=1}^N \frac{(p_j(\theta_i) - c_j)(u_{ij} - p_j(\theta_i))}{p_j(\theta_i)} = 0 \quad (3.2.7)$$

$$\frac{\partial}{\partial c_j} \log L(\boldsymbol{\theta}, \mathbf{a}, \mathbf{b}, \mathbf{c} | \mathbf{U}) = \frac{1}{1 - c_j} \sum_{i=1}^N \frac{(u_{ij} - p_j(\theta_i))}{p_j(\theta_i)} = 0 \quad (3.2.8)$$

上記の式の連立方程式を解いた値が最尤推定値となる。この方法は能力値 θ と項目パラメータを同時に推定するという意味で特に同時最尤推定法と呼ぶ。

3.3 周辺最尤推定法

同時最尤推定法では、尤度関数を微分した式が多くなってしまうという欠点がある。最大化する尤度関数の中から、能力値 θ を消去することができれば、最大化に

関するパラメータを大幅に減らすことができる。また受験者を追加してデータを増やしても未知数が増えることはないので推定値が安定するというメリットもある。つまり、反応行列が与えられたときの項目パラメータだけの尤度関数を構成すればよい。この考えを周辺最尤推定法と呼ぶ。

受験者の能力値 θ を尤度関数から消去し項目パラメータを推定する方法を考える。豊田秀樹 (2012) から引用して説明する。例中では、説明変数は離散型として 4 つの値 $\theta = 1.0, 0.5, -0.5, -1.0$ しかとらないと仮定する。例えば、それらを「優」、「良」、「可」、「不可」とみなしそのような学生が大学の中に 0.3, 0.2, 0.2, 0.3 の確率で存在しているものとする。このとき、母集団から受験者を選び出し、反応ベクトル \mathbf{u}_i を得る。 $L(\theta_i, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i)$ の能力値 θ_i に具体的な尺度値を代入して、 $g(\theta_i)$ との積和を求める。式は、

$$\begin{aligned} L(\theta, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) &= g(1.0)L(1.0, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) + g(0.5)L(0.5, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) \\ &\quad + g(-0.5)L(-0.5, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) + g(-1.0)L(-1.0, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) \\ &= 0.3L(1.0, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) + 0.2L(0.5, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) \\ &\quad + 0.2L(-0.5, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) + 0.3L(-1.0, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) \end{aligned} \quad (3.3.1)$$

のように導く。このように局外パラメータを確率分布から消去するために局外パラメータの分布と局外パラメータで条件つけられた確率分布との積和を計算することで能力値 θ_i を消去することができた。このことを、周辺化による局外パラメータの消去という。

能力値の階級の幅を狭めることができれば実質離散型として扱うことができる。例えば、小学 1 年生の体重を記録する際に、10kg から 35kg までを 0.2kg 刻みの離散変数として扱うことができる。そこで、 K 個の値をとる階級値 θ_k の分布 $g(\theta_k)$ を考える。これを、多項分布という。添え字 k は受験者ではなくヒストグラムの階級値についていることに注意する。このとき、母集団から 1 人の受験者を抽出し受験させ \mathbf{u}_i という結果を得る尤度は、

$$L(\mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) = \sum_{k=1}^K g(\theta_k)L(\theta_k, \mathbf{a}, \mathbf{b}, \mathbf{c}|\mathbf{u}_i) \quad (3.3.2)$$

と表現され、能力値 θ は消去される。ただしこの場合は能力値 θ の母集団におけるヒストグラムが既知であることが必要である。

3.4 ベイズ推定法

最尤推定法では、全問正答全問誤答の場合は、推定することができなかった。項目パラメータの分布を利用した推定方法に、ベイズ推定法がある。これは、全問正答、誤答の尺度値を推定することができる。まずは、ベイズの定理から説明する。

3.4.1 ベイズの定理

ある確率事象を A 、 B とする。 $P(B)$ は事象 B が起こる確率であり。 $P(A|B)$ は事象 B が起こる仮定の上で、事象 A が起こる確率とする。二つの積は、積事象の確率とし $P(A \cap B)$ とする。これらの関係を式にすると、

$$P(A \cap B) = P(B)P(A|B) \quad (3.4.1)$$

となる。両辺を $P(B)$ で割ると、

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (3.4.2)$$

となる。ここで A と B は一般的な事象を表しているので、式 (3.4.1) の A と B を入れ替えると、

$$P(A \cap B) = P(A)P(B|A) \quad (3.4.3)$$

である。これを式 (3.4.2) に代入すると、

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (3.4.4)$$

となり、この式をベイズの定理という。これを項目反応理論に応用するために式を変形していく。式 (3.4.4) を A を H に、 B を D に置き換える。

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)} \quad (3.4.5)$$

さらに、標本空間 Ω の事象 D には複数の原因 $H_i (i = 1, 2, 3, \dots, M)$ があると仮定する。原因 H_i について以下のような仮定を考える。

- $\forall i, j (i \neq j)$ について $H_i \cap H_j = \emptyset$
- $H_1 \cup H_2 \cup \dots \cup H_M = \Omega$

上記の2つを二つを仮定すると、事象が起こる確率 $P(D)$ は以下のような式になる。

$$P(D) = P(D \cap H_1) + P(D \cap H_2) + \dots + P(D \cap H_M) \quad (3.4.6)$$

ここで、 $P(D)$ は次のように整理することができる。

$$P(D) = P(D|H_1)P(H_1) + P(D|H_2)P(H_2) + \dots + P(D|H_M)P(H_M) \quad (3.4.7)$$

式 (3.4.7) を式 (3.4.5) に代入すると、

$$\begin{aligned} P(H_i|D) &= \frac{P(H_i)P(D|H_i)}{P(D|H_1)P(H_1) + P(D|H_2)P(H_2) + \dots + P(D|H_M)P(H_M)} \\ &= \frac{P(D|H_i)P(H_i)}{\sum_{i=1}^M P(D|H_i)P(H_i)} \end{aligned} \quad (3.4.8)$$

となる。ここで $P(H_i|D)$ は事後確率、 $P(H_i)$ を事前確率という。さらに、 $P(D|H_i)$ は原因 H_i の下で結果 D が得られる確率を意味し尤度という。

3.4.2 EAP 法 (Expected a posteriori 法)

次に EAP 法を紹介する。これは、 θ_i の事後分布の密度関数の θ_i に関する平均値を推定値とする推定法である。これは、2 乗損失もとでは事後分布の平均が 2 乗損失を最小にするという事実に基づいた推定法である。式にすると、

$$E[\theta_i | \mathbf{u}_i] = \int_{-\infty}^{+\infty} \theta_i \frac{p(\mathbf{u}_i | \theta_i) \pi(\theta_i)}{\int_{-\infty}^{+\infty} p(\mathbf{u}_i | \theta_i) \pi(\theta_i) d\theta_i} d\theta_i = \frac{\int_{-\infty}^{+\infty} \theta_i \pi(\theta_i) p(\mathbf{u}_i | \theta_i) d\theta_i}{\int_{-\infty}^{+\infty} \pi(\theta_i) p(\mathbf{u}_i | \theta_i) d\theta_i} \quad (3.4.9)$$

となる。ただし、 $\pi(\theta)$ は θ の事前分布の密度関数である。式中の積分は区分別求積法によって求めることができる。

ある区間 $[a, b]$ を M 等分した点 (幅は $\Delta\theta$) を $\theta_m (m = 1, 2, \dots, M, M+1)$ とすると

$$\int_b^a f(\theta) d\theta \simeq \sum_{m=1}^M f(\theta) \times \Delta\theta \quad (3.4.10)$$

によって左辺を N 個の長方形の面積の和に近似することができる。式 (3.5) 中の区間は $[-\infty, \infty]$ であるが、 $\pi(\theta)$ は標準正規分布の密度関数を仮定しているので、積分区間は $[-3, 3]$ や $[-4, 4]$ など十分であると考えられる。式にすると、

$$\hat{\theta}_i \simeq \frac{\sum_{m=1}^M \theta_m p(\mathbf{u}_i | \theta_m) \pi(\theta_m) \Delta\theta}{\sum_{m=1}^M p(\mathbf{u}_i | \theta_m) \pi(\theta_m) \Delta\theta} \quad (3.4.11)$$

と近似できる。これを、能力値 θ だけでなく、項目パラメータに用いることで推定値を求めることができる。そのためには、項目パラメータの事前分布を仮定しておく必要がある。

能力値 θ の事前分布 能力値 θ の事前分布には正規分布が仮定されることが多い。正規分布のパラメータは平均 μ_θ と分散 σ_θ^2 である。

識別力 a の事前分布 識別力の推定値は数学的には負の値もとれるが、識別力の事前分布には正の範囲で定義される対数正規分布が仮定されることが多い。

困難度 b の事前分布 困難度の事前分布にも正規分布が仮定されることが多い。

偶然率 c の事前分布 偶然率の推定値は確率を表現するパラメータである。このため事前分布はベータ分布が仮定されることが多い。ベータ分布のパラメータは α, β であり、密度関数は

$$\frac{x^{\alpha-1} (1-x)^{\beta-1} \Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \quad (3.4.12)$$

ベータ分布の最頻値は、

$$\frac{\alpha - 1}{\alpha + \beta - 2} \quad (3.4.13)$$

であることが知られている。選択形式の問題の場合、偶然率の尤もらしい目安は選択肢数 k の逆数 $\frac{1}{k}$ と考えられる。目安 $\frac{1}{k}$ を事前分布の最頻値の一致させるならば、例えば、

$$\alpha = 20 \times \frac{1}{k} + 1 \quad (3.4.14)$$

$$\beta = 20 \times \left(1 - \frac{1}{k}\right) + 1 \quad (3.4.15)$$

とする。5肢選択ならば $c = 0.2$ とし $\alpha = 5, \beta = 17$ のベータ分布を考える。

3.5 推定値と考察

3.5.1 バイズ推定

第2章のテストの反応行列を用いてバイズ推定法で項目パラメータと能力値 θ を推定する。

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}^t$$

ただし、人数 \times 問題の大きさの行列を使うために実際の計算には行列を転置したものをを用いる。2パラメータロジスティックモデルと3パラメータロジスティックモデルを用いて推定する。使用するプログラムは付録に掲載する。それぞれのパラメータの推定値は以下のようになった。

	困難度	識別力
問題 1	0.329	0.369
問題 2	0.767	0.843
問題 3	1.091	0.705
問題 4	1.019	0.766
問題 5	0.197	3.083
問題 6	0.958	0.534
問題 7	0.259	0.431
問題 8	-0.18	3.212

	困難度	識別力	偶然率
問題 1	1.414	1.622	0.350
問題 2	0.958	1.674	0.140
問題 3	1.275	1.787	0.137
問題 4	1.274	1.785	0.136
問題 5	0.026	2.970	0.038
問題 6	1.334	1.779	0.200
問題 7	1.414	1.623	0.350
問題 8	-0.487	3.175	0.044

表 1: 2 パラメータロジスティックモデル 表 2: 3 パラメータロジスティックモデル

困難度の推定値に関して、2 パラメータロジスティックモデルと 3 パラメータロジスティックモデルを比べると 2 パラメータロジスティックモデルよりも 3 パラメータロジスティックモデルの結果の方が最大値と最小値の差が 2 パラメータロジスティックモデルでは 1.271、3 パラメータロジスティックモデルでは 1.901 と広がっている。つまり、2 パラメータロジスティックモデルで推定して難しいと推定された項目は 3 パラメータロジスティックモデルではより難しいと判断され、簡単と推定された問題はより簡単と推定される結果となった。これは偶然率がパラメータに入ったことで、難しい問題は実力で正解することがより難しくなっていることに起因していると考えられる。

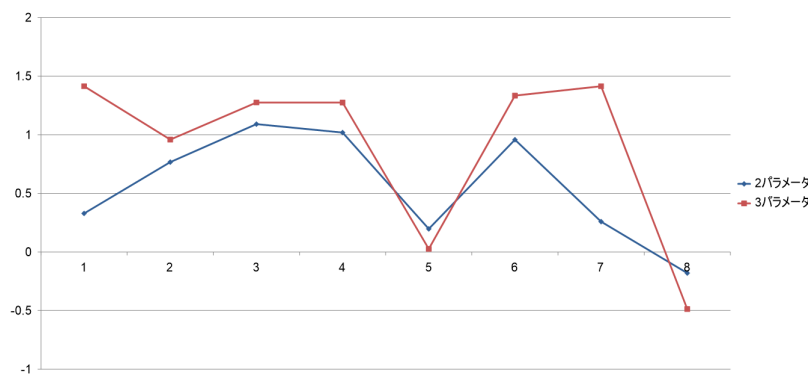


図 5: 困難度の推定値

また、他にも以下のようなことが分かる。

- 1 第5問と第8問がともに高い値が出ている。第5問と第8問はともに偶然率が低く偶然正解することが少ないことを考えるとほとんどの受験者が実力通り

の結果が出ていて能力値 θ をしっかりと測れる項目である。

- 2 第4問と第5問に関しては同じ高通過率項目、つまり多くの受験者が正解している項目であるが困難度は第4問の方が高く、識別力は第5問の方が高い。さらに、項目特性図を比べると第5問の方が正答率がより高位に張り付いた状態になっていて、第4問は誤答も万遍なくある状態になっている。つまり、第4問の方がしっかりと選択肢が機能していて「良い」問題といえる。

また図7は、能力値 θ の推定値をグラフ化したものである。困難度がより高く推定された3パラメータロジスティックモデルの能力値 θ の推定値のほうが、2パラメータロジスティックモデルの能力値 θ の推定値より、低く推定されている。ここで、能力値 θ の推定値を標準化しその値を用いて、グラフに表すと、図8のようになった。

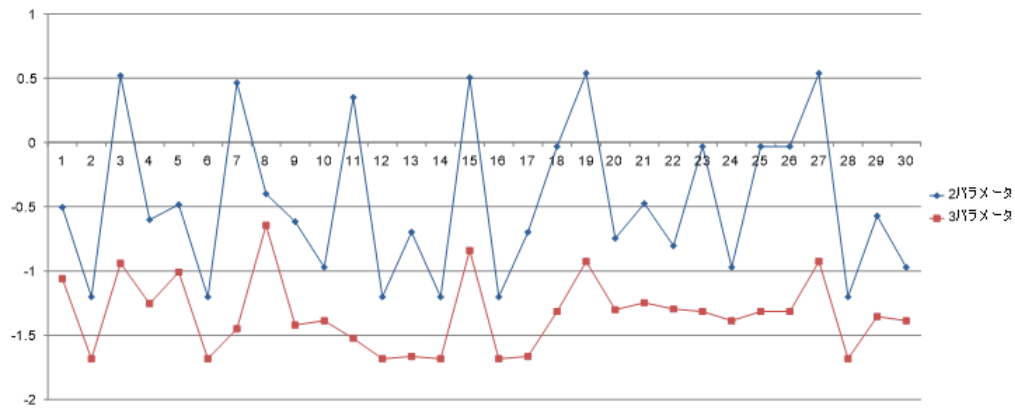


図 6: 能力値 θ の推定値

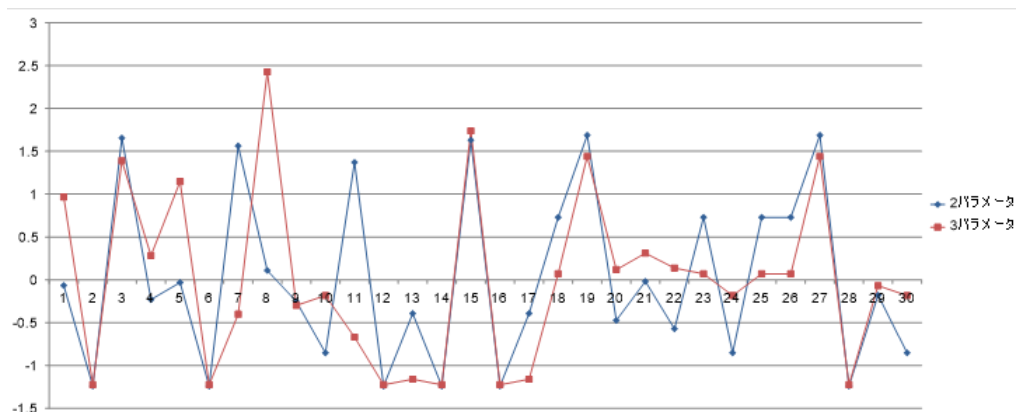


図 7: 能力値 θ の推定値 (標準化 ver)

また、能力値 θ の推定値については以下のようなことがわかる。

- 1 標準化すると2パラメータも3パラメータも似たような値を推定している。
 - 2 受験者7と受験者8や、受験者22と受験者23の関係のように2パラメータも3パラメータで能力値 θ の大小関係が逆転している箇所が見られる。
- 各受験者の正誤を表した反応ベクトルを調べると受験者7と受験者8について、総合得点(8問中3問正解)は同じであるが、偶然率の値が高い問題1に受験者7は誤答、受験者8は正答している。また、偶然率の値が低い問題5,8に受験者7は正答、受験者8は誤答している。これらのことから2パラメータで推定したとき、2人の受験者はしっかり実力で3問正解しているため、困難度が高い問題により正解している受験者7の方が受験者8よりも高い評価を受けているが、偶然率を踏まえると、より実力で正解している受験者8の方が評価されている

3.5.2 最尤推定法との比較

ここで、最尤推定法も用いた推定値との比較もしていく。まずは、最尤推定法を用いた推定値とベイズ推定法の値(2パラメータ)を表にまとめたものと能力値 θ を標準化しグラフに表したものを以下に示す。

	困難度	識別力
問題 1	0.329	0.369
問題 2	0.767	0.843
問題 3	1.091	0.705
問題 4	1.019	0.766
問題 5	0.197	3.083
問題 6	0.958	0.534
問題 7	0.259	0.431
問題 8	-0.18	3.212

	困難度	識別力
問題 1	-6.87	0.640
問題 2	-2.87	0.841
問題 3	-0.87	3.999
問題 4	0.461	2.705
問題 5	1.461	1.091
問題 6	2.261	0.671
問題 7	2.928	0.266
問題 8	3.499	0.194

表 3: ベイズ推定法

表 4: 最尤推定法

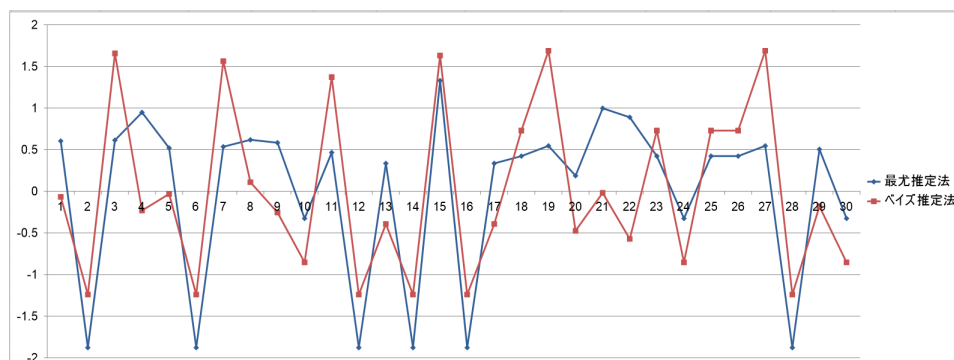


図 8: 能力値 θ の推定値 (標準化 ver)

全問不正解の受験者がいるので、最尤推定法の性質上、能力値の推定値は $-\infty$ になる。グラフは下限を設定しているので、実際は -2 よりもずっと小さい値を示している。ベイズ推定法と比べても最尤推定法は項目パラメータも安定していないことが分かる。さらに、受験者数も30人と少ないため最尤推定法は適していない。

4 ロジスティックモデルの多次元化

これまで考えてきたモデルは項目が受験者の単一の能力値を測定している仮定が置かれていた。仮に項目が潜在能力値を2つ以上測定しているという仮定が成り立つならば、前章までで考えてきたロジスティックモデルの能力値 θ を多次元で表現したロジスティックモデルを考えることができる。例えば、数学ならば、「計算力」と「論理力」の2つの能力が考えられる。実際のテストの運用ではこのような状況が十分に考えられる。そこで、多次元ロジスティックモデルを考えていく。初めに、一般に知られている多次元ロジスティックモデルを紹介する。

多次元ロジスティックモデル 今回考える多次元ロジスティックモデルは2パラメータロジスティックモデルを多次元の能力値のケースへ一般化したものである。2パラメータ多次元ロジスティックモデルは次のように表現される。

$$p_j(\theta) = \frac{1}{1 + \exp(-\mathbf{a}_j^t \boldsymbol{\theta}_i + b_j)} \quad (4.1.1)$$

ここで \mathbf{a} は項目ごとの識別力を表すベクトルである。また、 $\boldsymbol{\theta}$ も受験者ごとの多次元の能力値を並べたベクトルである。つまり式中の $\mathbf{a}_j^t \boldsymbol{\theta}_i + b_j$ は次のように表現することができる。

$$\mathbf{a}_j^t \boldsymbol{\theta}_i + b_j = \sum_{l=1}^L a_{jl} \theta_{il} + b_j \quad (4.1.2)$$

能力が2次元なものであると仮定して考えると

$$\boldsymbol{\theta} = (\theta_1, \theta_2)^t \quad (4.1.3)$$

のように項目が受験者の違う2つ能力値を測定していると仮定する。しかし、2次元で考えるがゆえの問題も出てくる。2つの能力値のうちどちらか1つの能力値を推定するための基準となる問題が必要になる。問題の識別力ベクトルの向きが決まっていないと、推定したい能力値が正確に測れていないのではないかと考えられる。また、2つの識別力ベクトルは一次独立のものを仮定することで1次元の話と区別するようにする。前章までに使用した各パラメータの事後密度の期待値を用いたベイズ推定法を使用して能力値や項目パラメータを推定する。

5 多次元ロジスティックモデルを用いた能力値推定のための問題選定(仮)

実際にテストを作成する際に問題は作成者が意図を持って採用される。受験者の能力を正確に推定しようとするならば、過去に実施していて、項目パラメータが既知である問題ばかりそろえれば能力値の推定値は安定しそうである。しかし、すべて過去に実施されたテストと同じ問題では、内容は全く理解できていないが過去問題を暗記しただけの人は、能力値が高く推定されてしまう。このようなことを防ぐためには、すべて過去問題と同じ問題というわけにもいかない。そこで考えるのは、過去問題もある程度入れつつ、新しい問題も採用するテスト作成である。そのとき過去問題は

- 識別力が高いものを選ぶべきなのか
- 困難度が高いものを選ぶべきなのか
- 過去問題は全体の中でどのくらいの割合で入れるべきなのか

などを考えなければならない。

ここで受験者の能力値を測るのは、新しく採用された問題だけでよいのかを考える。前章にある通り、数学における「計算力」と「論理力」の2つの能力が項目から測定されると考えると、その2つの能力が測れているとする基準となる項目が必要になると考えられる。例えば、過去問題の中で「計算力」「論理力」の2つの能力を測定できた問題20問と新しく採用された項目パラメータが未知である問題30問でテストを作る。まず、

- (1) 過去問題から新しく採用された問題が2つの能力をどのくらい測れているのかなどの項目パラメータを推定する。
- (2) 新しい問題だけを用いて受験者の能力値を推定する。

という流れである。このように過去問題はあくまでも受験者の能力値を推定するための問題の項目パラメータを推定するために使うことで、毎回同じ問題を使わなくても新しい問題で受験者の能力値が測れるのではないかと考える。

この章では、計算機を使って項目パラメータが既知である問題をどのくらいの割合で入れると受験者の能力値が安定するのかをシミュレーションしていく。状況設定は100人の受験者に100問の問題を解いてもらう。そのうち、一定の割合で項目パラメータが既知である問題を入れる。今回は過去問題が25問、50問、75問の3パターンで計算していく。計算の流れとしては、

- 1 過去問題と反応行列から新しい問題の項目パラメータを推定
- 2 項目パラメータが推定された新しい問題を使って受験者の能力値を推定

の2段階に分ける。ここで以下の2点に注意されたい。

- 過去問題のパラメータは既知扱いであるので、推定の対象外である。
- 受験者の能力値の推定には過去問題は使用せず、新しい問題だけを使用する。

ここで、各項目パラメータに関しては以下のように設定した。

- 識別力は a_1 と a_2 とし2つのベクトルを考える。 a_1 と a_2 はそれぞれ別の能力を測っていると考ええるために、一次独立な2つのベクトルを仮定する。今回は a_1 は最小値を0.5、最大値を2とし均等な間隔で50個並べ、それを2つ繋げたベクトルとし、 a_2 は最大値を1とし a_1 と同じ並べ方をしたベクトルとした。

$$\mathbf{a}_1 = (0.5, 0.53061, 0.56122, \dots, 2, 0.5, 0.53061, 0.56122, \dots, 2)$$

$$\mathbf{a}_2 = (0.5, 0.53061, 0.56122, \dots, 1, 0.5, 0.53061, 0.56122, \dots, 1)$$

- 困難度 b は最小値 -1 、最大値 1 とし100問分間隔が均等になるように並べたベクトルとした。

$$\mathbf{b} = (-1, -0.9797, -0.9595, \dots, 1)$$

- 能力値 θ の設定値は識別力と同じく θ_1 と θ_2 とし2つのベクトルが一次独立になるように設定する。一つは最小値 -2 、最大値 2 とし昇順で100人分均等に並べ、他方は最初の50人は -1 、残り50人は 1 として並べた。

$$\mathbf{\theta}_1 = (-2, -1.9595, -1.9191, \dots, 2)$$

$$\mathbf{\theta}_2 = (-1, -1, -1, \dots, -1, 1, 1, \dots, 1)$$

これらを用いて擬似的な反応行列を生成する。過去問を100問中25問採用する場合を例に上げ具体的な計算の流れを説明する。

- (1) 新しい問題の項目パラメータを推定する。実際には、1回の計算で出てきた推定値を初期値として更新する際に、過去問題の25問のパラメータは更新せず、設定値のままにしておき、新しい問題だけの項目パラメータだけを更新する。そして、更新された初期値を用いて2回目の計算を行う。その際も過去問題の項目パラメータは更新しない。それによって、新しい問題だけの項目パラメータを推定する。
- (2) 新しい問題の項目パラメータが決定したら、その問題を用いて受験者の能力値を推定する。
- (3) 上2つの計算を100回行い推定値が一定の誤差に収まるように収束するまで計算を続ける。
- (4) 推定値の信頼度を上げるためにこの試行を100回行い、それぞれの値の平均値を求める。

次の節で実際に行った結果と考察を述べる。

5.1 項目パラメータと能力値の推定値

以下に示すのは、能力値 θ_1 、 θ_2 の推定値と設定値との誤差の平均を、過去問題の割合毎にまとめたものである。

	過去問題 0 問	過去問題 25 問	過去問題 50 問	過去問題 75 問
θ_1	0.397705342	0.408443905	0.416983791	0.520276119
θ_2	0.488909176	0.459919565	0.5123526	0.51364787

推定値に関して以下のようなことがわかった。

- 能力値 θ_1 の推定値に関しては過去問を全く入れなかった場合が設定値との誤差が一番小さくなった
- 能力値 θ_2 の推定値に関しては過去問題を 100 問中 25 問採用した場合が、一番設定値との誤差が少なくなった。
- θ_1 に関しての過去問題 0 問と過去問題 25 問の差は 0.01、そして、 θ_2 に関しての過去問題 0 問と過去問題 25 問の差は 0.03 となっている。

上記のことより以下のことが考察できる。

- △ 能力値を推定する際には、過去問題は使用せず、新しい問題だけ使用するので、過去問 25 問だと能力値を推定する問題が 75 問と多く推定値が安定したと考えられる。
- △ 今回既知として扱った問題は識別力の値が高いものから選んでいるため、過去問題が 50 問や、75 問だと識別力の値が低い値などを含んでしまうため推定値の精度が下がってしまったと考えられる。逆に過去問題を 25 問だけ既知とすると識別力の値が低くても 1.234 あるので推定値の精度も他に比べて上がったと考えられる。

項目パラメータに関して、推定している個数がそれぞれで違うので一概には比較できない。そこで、1 問目から 25 問目までの推定値の平均を比較してみる。

	過去問題 0 問	過去問題 25 問	過去問題 50 問	過去問題 75 問
a_1	0.357032998	0.365192399	0.575955796	0.897635541
a_2	0.581752051	0.459429589	0.660244633	0.877362495
b	0.200210509	0.256411709	0.308571305	0.547870155

- 項目パラメータ a_1 に関して、差は0.01程、過去問題0問の方が過去問題25問の方が、小さくなっている。同じように、 a_2 では0.13程過去問題25問が小さく、 b は0.5ほど過去問題を入れない方が小さくなっている。以上のことから人数が100人と少ないため項目パラメータ、特に識別力の値は安定しないと考えられる。

6 最後に

最後の推定のシミュレーションに関して、今回は項目パラメータの値が高い順に採用した。項目パラメータが既知の問題の取り方は他にも、低い方から取ったり平均が0になるように取ったりとパターンはあるので次の機会ではこれらのパターンも試したい。

本章で紹介している通り、付録にR言語を用いたスクリプトを掲載してある。読者自身がテストを運用していく際に、ぜひ使用して分析、評価していただきたい。

本研究を進めるにあたり、岩佐学准教授には指導教員として終始熱心なご指導をいただきました。心から感謝いたします。お礼申し上げます。

参考文献

- 豊田秀樹 (2012). 項目反応理論 [入門編]
- 住 政二郎 (2015). 適応型テストへの応用—ベイズ EAP 推定法とフィッシャー情報量の概説—
- 前川眞一. 項目反応理論概説

付録1 第2章で使用した問題

- (1) 次の文と関係の深い人物を記号で答えてください。アメリカの教育学者。プラグマティズムの立場から生産活動を基礎とする労作学校を主張・実施した。

(a) デューイ (b) フレーベル (c) シュプランガー (d) スペンサー

- (2) 次の文と関係の深い人物を記号で答えてください。従来の「臨画帳」を写すだけの美術教育に反対し、自由画中心の教育を提唱した。

(a) 木下竹次 (b) 芦田恵之介 (c) 鈴木三重吉 (d) 山本鼎

- (3) 次の文で間違っているものを答えてください。

(a) 特別支援学校の設置義務は都道府県にある。

- (b) 児童福祉司は、児童相談所長の命を受けて、児童の保護その他児童福祉に関する事項について相談に応じ、専門技術に基づき必要な指導を行う等児童福祉増進に努めるものとされている。
- (c) アメリカの教育学者で近代教授学の父であるとされるのがコメニウスであり、彼はその著書「大教授学」においてすべての民衆を対象にした一般教育を構想し、一斉授業も提唱した。
- (d) いじめが止んでいる状態というのは、その行為が止んでいる状態が少なくとも3カ月継続していることをいう。

(4) 次の選択肢の中から正しいものを答えてください。

- (a) 普通免許状は18歳未満のもの、高等学校を卒業しないものや禁錮以上の刑に処せられたものにも、条件次第では授与される。
- (b) 教育職員検定は、受験者の人物、学力、実務および身体について、ランダムに選ばれた現職の教員が行う。
- (c) 教員免許更新制は、教員として必要な資質を保持できるように、定期的に最新の知識技能を身につけることで社会の尊厳と信頼を得ることを目指すものである。
- (d) 教員はすべて、免許取得50年後に免許更新講習を受けその課程を修了しなければ、免許の有効期間を更新できず免許状はその効力を失う。

(5) 次の空欄に当てはまる語句を選びなさい。すべて国民は、法律の定めるところにより、その保護する子女に(1)を受けさせる義務を負う。(日本国憲法)

- (a) 教育 (b) 義務教育 (c) 普通教育 (d) 特別教育

(6) 次の文は何について述べたものか答えてください。

人間が健全な発達をするためには人間社会で多様な経験をすることが必要なことを示してくれた。ヴィクトールと名づけられたその子は、会話はほぼ不可能な状態であった。これは、神経言語学で言われるある期間を過ぎてしまったためだといわれている。この子を題材にフランスでは映画化されそのタイトルは当時、映画界を賑わせた。

- (a) アヴェロンの野生児 (b) 野生の少年
(c) 言語獲得の臨界期 (d) フィリップピネル

(7) 次の文は何について述べたものか答えてください。

記憶のまとまりとしての単位

(a) メモリ (b) チャンク (c) ユニット (d) ミラー

(8) 次の文は何について述べたものか教えてください。

学習曲線を描いたときに、難易度によって途中で学習の成果が向上しなくなる現象

(a) スランプ (b) 停滞減少 (c) 負の強化 (d) プラトー

付録2 推定に使用したプログラム

Listing 1: 最尤推定法

```
1  IRT <- function(data,epsilon) {
2    n <- length(data[,1])
3    z <- length(data[1,])
4    a_0 <- runif(15,min = 0.5,max = 2)
5    a_0 <- a_0*(prod(a_0))^( -1 / z)
6    b_0 <- c()
7    for (d in 1:z){
8      bb <- sum(data[,d])/n
9      b_0 <- c(b_0,c(bb))
10   }
11   b_0 <- b_0-mean(b_0)
12   theta_0 <- rep(0, n)
13
14   f_theta <- function(a, b) {
15     nouryoku <- c()
16     for (j in 1:n){
17       cut <-c(data[j,])
18       P <- function(x) {
19         cutt <- 0
20         for (i in 1:z){
21           if (cut[i]==1) {
22             cutt <- cutt + log(1/(1 + exp((-1.7) * a[i] * (x - b[i]
23               ]))))
24           }
25           else{
26             cutt <- cutt + log(1 - 1/(1 + exp((-1.7) * a[i] * (x
27               - b[i]))))
28           }
29         }
30         return (cutt)
31       }
32     }
33     #par(new = T)
34     #curve(P,xlim = c(-4,4),ylim = c(-10,0))
35   }
```

```

32  nouryoku <- c(nouryoku,c(optimize(P,interval = c(-4,4),
    maximum = TRUE)$maximum))
33  }
34  return(nouryoku)
35  }
36
37
38  f_sikibetu <- function(theta,b){
39  sikibetu <- c()
40  for (i in 1:z){
41  cut <-c(data[,i])
42  P <- function(a) {
43  cutt <- 0
44  for (j in 1:n){
45  if (cut[j]==1) {
46  cutt <- cutt + log(1/(1 + exp((-1.7) * a * (theta[j] -
    b[i]))))
47  }
48  else{
49  cutt <- cutt + log(1 - 1/(1 + exp((-1.7) * a * (theta[
    j] - b[i]))))
50  }
51  }
52  return (cutt)
53  }
54  sikibetu <- c(sikibetu,c(optimize(P,interval = c(0,10),
    maximum = TRUE)$maximum))
55  #sikibetu <- sikibetu * (prod(sikibetu))^(1/z)
56  }
57  return(sikibetu)
58  }
59
60  f_konnando <- function(theta, a) {
61  konnando <- c()
62  for (i in 1:z){
63  cut <-c(data[,i])
64  P <- function(b) {
65  cutt <- 0
66  for (j in 1:n){
67  if (cut[j]==1) {
68  cutt <- cutt + log(1/(1 + exp((-1.7) * a[i] * (theta[j]
    - b))))
69  }
70  else{
71  cutt <- cutt + log(1 - 1/(1 + exp((-1.7) * a[i] * (
    theta[j] - b))))
72  }

```

```

73         }
74     return (cutt)
75 }
76 konnando <- c(konnando,c(optimize(P,interval = c(-2,2),
77     maximum = TRUE)$maximum))
78 konnando <- konnando-mean(konnando)
79 }
80 return(konnando)
81 }
82 for (i in 1:100){
83     theta_1 <- f_theta(a_0,b_0)
84     a_1 <- f_sikibetu(theta_1,b_0)
85     b_1 <- f_konnando(theta_1,a_1)
86     difference_theta <- max(abs(theta_1 - theta_0))
87     difference_a <- max(abs(a_1 - a_0))
88     difference_b <- max(abs(b_1 - b_0))
89     if(difference_theta < epsilon & difference_a < epsilon &
90         difference_b < epsilon){
91         print(i)
92         break()
93     }
94     theta_0 <- theta_1
95     a_0 <- a_1
96     b_0 <- b_1
97 }
98 return(list(a_0, b_0, theta_0))
99 }

```

Listing 2: ベイズ推定法

```

1 expectation_theta <- function(data, a, b, c) {#nolint
2     estimates_theta <- c()
3     for (j in 1:dim(data)[1]) {
4         cut <- c(data[j, ])
5         P <- function(theta) {#nolint
6             cutt <- 1
7             for (i in 1:50) {
8                 if (cut[i] == 1) {
9                     cutt <- cutt * LP3(theta, a[i], b[i], c[i])
10                }
11            } else{
12                cutt <- cutt * (1 - LP3(theta, a[i], b[i], c[i]))
13            }
14        }
15        return (cutt)
16    }
17    cunt_numerator <- 0

```

```

18   cunt_denominator <- 0
19   for (h in 1:41) {
20     cunt_numerator <-cunt_numerator+((-4 + (h-1) * 0.2) * P(-4
      + (h-1) * 0.2) * prior_distribution_theta(-4 + (h-1)
      * 0.2) * 0.2) # nolint
21
22     cunt_denominator <- cunt_denominator + (P(-4 + (h-1) *
      0.2) * prior_distribution_theta(-4 + (h-1) * 0.2) *
      0.2)#nolint
23   }
24   estimates_theta <- c(estimates_theta, c(cunt_numerator /
      cunt_denominator))
25   }
26   return(estimates_theta)
27   }
28
29   expectation_a <- function(data, b, theta, c, a) {
30     estimates_a <- c()
31     for (i in 1:dim(data)[2]) {
32       cut <- c(data[, i])
33       P <- function(a) {
34         cutt <- 1
35         for (j in 1:dim(data)[1]) {
36           if (cut[j] == 1) {
37             cutt <- cutt * (LP3(theta[j], a, b[i], c[i])) #nolint
38           }
39           else{
40             cutt <- cutt * (1 - LP3(theta[j], a, b[i], c[i]))#
              nolint
41           }
42         }
43       }
44     }
45     cunt_numerator <- 0
46     cunt_denominator <- 0
47     for (h in 1:40) {
48       cunt_numerator <-cunt_numerator+(((h) * 0.1) * P((h) *
        0.1) * prior_distribution_a((h) * 0.1) * 0.1) # nolint
49
50       cunt_denominator <- cunt_denominator + (P((h) * 0.1) *
        prior_distribution_a((h) * 0.1) * 0.1)#nolint
51     }
52     estimates_a <- c(estimates_a, c(cunt_numerator /
      cunt_denominator))
53   }
54   a[1:50] <- estimates_a[1:50]
55   return(a)

```



```

56   }
57
58
59
60   expectation_b <- function(data, a, theta, c, b) {
61     estimates_b <- c()
62     for (i in 1:dim(data)[2]) {
63       cut <- c(data[, i])
64       P <- function(b) {#nolint
65         cutt <- 1
66         for (j in 1:dim(data)[1]) {
67           if (cut[j] == 1) {
68             cutt <- cutt * (LP3(theta[j], a[i], b, c[i]))
69           }
70           else{
71             cutt <- cutt * (1 - LP3(theta[j], a[i], b, c[i]))
72           }
73         }
74         return(cutt)
75       }
76       cunt_numerator <- 0
77       cunt_denominator <- 0
78       for (h in 1:40) {
79         cunt_numerator <- cunt_numerator + ((-4 + (h-1) * 0.2) * P
          (-4 + (h-1) * 0.2) * prior_distribution_b(-4 + (h-1) *
            0.2) * 0.2)#nolint
80         cunt_denominator <- cunt_denominator + (P(-4 + (h-1) *
          0.2) * prior_distribution_b(-4 + (h-1) * 0.2) * 0.2)#
          nolint
81       }
82       estimates_b <- c(estimates_b, cunt_numerator / cunt_denominator
83         )
84     }
85     b[1:50] <- estimates_b[1:50]
86     return(b)
87   }
88
89   expectation_c <- function(data, a, theta, b, c) {
90     estimates_c <- c()
91     for (i in 1:dim(data)[2]) {
92       cut <- c(data[, i])
93       P <- function(c) {#nolint
94         cutt <- 1
95         for (j in 1:dim(data)[1]) {
96           if (cut[j] == 1) {
97             cutt <- cutt * LP3(theta[j], a[i], b[i], c)

```

```

98     }
99     else{
100         cutt <- cutt * (1 - LP3(theta[j], a[i], b[i], c))
101     }
102     }
103     return(cutt)
104 }
105 cunt_numerator <- 0
106 cunt_denominator <- 0
107 for (h in 1:40) {
108     cunt_numerator <- cunt_numerator + (((h - 0.5) / 40) * P((h
109         - 0.5) / 40) * prior_distribution_c((h - 0.5) / 40) /
110         40)#nolint
111     cunt_denominator <- cunt_denominator + (P((h - 0.5) / 40)
112         * prior_distribution_c((h - 0.5) / 40) / 40)#nolint
113 }
114 estimates_c <- c(estimates_c, cunt_numerator / cunt_denominator
115 )
116 }
117 c[1:50] <- estimates_c[1:50]
118 return(c)
119 }
120
121 bayesianEstimates <- function(data, epsilon, e) {#nolint
122     n <- length(data[, 1]) 人数#
123     z <- length(data[1, ]) 問題数#
124     a_0 <- seq(0.5, 2, length = z)
125     b_0 <- seq(-2, 2, length = z)
126     theta_0 <- seq(-2, 2, length = n)
127     c_0 <- rep(1 / 4, n)
128     for (i in 1:100) {
129         b_1 <- expectation_b(data, a_0, theta_0, c_0, b_0)
130         a_1 <- expectation_a(data, b_1, theta_0, c_0, a_0)
131         c_1 <- expectation_c(data, a_1, theta_0, b_1, c_0)
132         theta_1 <- expectation_theta(data, a_1, b_1, c_1)
133         difference_theta <- max(abs(theta_1 - theta_0))
134         difference_a <- max(abs(a_1[1:50] - a_0[1:50]))
135         difference_b <- max(abs(b_1[1:50] - b_0[1:50]))
136         difference_c <- max(abs(c_1[1:50] - c_0[1:50]))
137         if(difference_theta < epsilon & difference_a < epsilon &
138             difference_b < epsilon & difference_c < epsilon){#nolint
139             print(i)
140             break ()
141         }
142         theta_0 <- theta_1
143         a_0[1:50] <- a_1[1:50]
144         b_0[1:50] <- b_1[1:50]

```

```

140     c_0[1:50] <- c_1[1:50]
141     print(c(i))
142     setTxtProgressBar(pb, i)
143   }
144   return(c(theta_0, a_0, b_0, c_0))
145 }
146
147 ave_cul <- function(e) {
148   t_lis <- rep(0, length = num_users)
149   a_lis <- rep(0, length = num_items)
150   b_lis <- rep(0, length = num_items)
151   c_lis <- rep(0, length = num_items)
152
153   a <- seq(0.5, 2, length = num_items)
154   b2 <- seq(-2, 2, length = num_items)
155   c_0 <- rep(1 / 4, num_items)
156
157   theta_mat <- matrix(0, nrow = num_users, ncol = num_k)
158   theta_mat[, 1] <- seq(-2, 2, length = num_users)
159   t1 <- theta_mat[, 1]
160   reaction_3_o <- matrix(0, nrow = num_users, ncol = num_items)
161   for (j in 1:num_users) {
162     for (i in 1:num_items) {
163       num <- LP3(t1[j], a[i], b2[i], c_0[i])
164       reaction_3_o[j, i] <- rbinom(1, 1, num)
165     }
166   }
167   data_3_o <- reaction_3_o
168   for (z in 1:e) {
169     print(z)
170     result <- bayesianEstimates(data_3_o, 0.001, e)
171     t_lis <- t_lis + result[1:100]
172     a_lis <- a_lis + result[101:200]
173     b_lis <- b_lis + result[201:300]
174     c_lis <- c_lis + result[301:400]
175   }
176   return(list(t_lis / rep(e, length = length(t_lis)),
177             a_lis / rep(e, length = length(a_lis)),
178             b_lis / rep(e, length = length(b_lis)),
179             c_lis / rep(e, length = length(c_lis)),
180             write.csv(t_lis, "t_50_lis.csv"),
181             write.csv(a_lis, "a_50_lis.csv"),
182             write.csv(b_lis, "b_50_lis.csv"),
183             write.csv(c_lis, "c_50_lis.csv")))
184 }

```

Listing 3: ベイズ推定法 (多次元)

```

1  LP2_3 <- function(a1, a2, b, t1, t2) { # nolint
2    return(1 / (1 + exp((a1 * t1) + (a2 * t2) + b)))
3  }
4
5  prior_distribution_t <- function(x) {
6
7    return((1 / sqrt(2 * pi)) * exp(-1 * ((x^2) / 2)))
8
9  }
10
11 prior_distribution_a <- function(x) {
12
13   return((1 / ((sqrt(2 * pi)) * x)) * exp(-1 * (((log(x))^2) /
14     2)))
15 }
16
17 prior_distribution_b <- function(x) {
18
19   return((1 / (sqrt(2 * pi))) * exp(-1 * (((x)^2) / 2)))
20
21 }
22
23 prior_distribution_w <- function(x) {
24   return((1 / sqrt(2 * pi)) * exp(-1 * ((x^2) / 2)))
25 }
26
27 num_items <- 100
28 num_users <- 100
29 num_k <- 2
30
31 f_t1_bay <- function(data, t2, b, a1, a2) {
32 estimates_t1_list <- c()
33 for (i in 1:dim(data)[1]) {
34 P <- function(theta) {
35   cutt <- 0
36   for (j in 1:100) {
37     if (data[i, j] == 1) {
38       cutt <- cutt + log(LP2_3(a1[j], a2[j], b[j],
39         theta, t2[i]))
39     }
40     else{
41       cutt <- cutt + log(1 - LP2_3(a1[j], a2[j], b[j],
42         ], theta, t2[i])) # nolint
43     }
44   }
45 }

```

```

44         return(exp(cutt))
45     }
46     cunt_numerator <- 0
47     cunt_denominator <- 0
48     for (h in 1:41) {
49         x <- P(-4 + (h - 1) * 0.2)
50         cunt_numerator <- cunt_numerator + ((-4 + (h - 1) * 0.2) * x *
51             prior_distribution_t(-4 + (h - 1) * 0.2) * 0.2) # nolint
52         cunt_denominator <- cunt_denominator + (x *
53             prior_distribution_t(-4 + (h - 1) * 0.2) * 0.2) # nolint
54     }
55     estimates_t1_list <- c(estimates_t1_list, c(cunt_numerator /
56         cunt_denominator))
57     #estimates_t1_2 <- (estimates_t1_list - mean(estimates_t1_list)) /
58         sd(estimates_t1_list)
59     return(estimates_t1_list)
60 }
61
62 f_t2_bay <- function(data, t1, b, a1, a2) {
63     estimates_t2_list <- c()
64     for (i in 1:dim(data)[1]) {
65         Q <- function(theta) {
66             cutt <- 0
67             for (j in 1:100) {
68                 if (data[i, j] == 1) {
69                     cutt <- cutt + log(LP2_3(a1[j], a2[j], b[j], t1
70                         [i], theta))
71                 }
72                 else{
73                     cutt <- cutt + log((1 - LP2_3(a1[j], a2[j], b[j]
74                         ], t1[i], theta)))
75                 }
76             }
77             return(exp(cutt))
78         }
79         cunt_numerator <- 0
80         cunt_denominator <- 0
81         for (h in 1:41) {
82             x <- Q(-4 + (h-1) * 0.2)
83             cunt_numerator <- cunt_numerator + ((-4 + (h-1) * 0.2) * x *
84                 prior_distribution_t(-4 + (h-1) * 0.2) * 0.2) # nolint
85             cunt_denominator <- cunt_denominator + (x *
86                 prior_distribution_t(-4 + (h-1) * 0.2) * 0.2) # nolint
87         }
88     }
89     return(estimates_t2_list)
90 }

```

```

83 estimates_t2_list <- c(estimates_t2_list, c(cunt_numerator /
      cunt_denominator))
84 }
85 #estimates_t2_2 <- (estimates_t2_list - mean(estimates_t2_list)) /
      sd(estimates_t2_list)
86     return(estimates_t2_list)
87 }
88
89
90
91 f_a1_bay <- function(data, t1, t2, b, a1, a2) {
92 estimates_a1_list <- c()
93 for (j in 1:dim(data)[2]) {
94 S <- function(a1) {
95     cutt <- 0
96     for (i in 1:dim(data)[1]) {
97         if (data[i, j] == 1) {
98             cutt <- cutt + log(LP2_3(a1, a2[j], b[j], t1[i]
              ], t2[i]))
99         }
100         else{
101             cutt <- cutt + log((1 - LP2_3(a1, a2[j], b[j],
              t1[i], t2[i])))
102         }
103     }
104     return(exp(cutt))
105 }
106 cunt_numerator <- 0
107 cunt_denominator <- 0
108 for (h in 1:40) {
109     x <- S((h) * 0.1)
110     cunt_numerator <-cunt_numerator+(((h * 0.1) ) * x *
      prior_distribution_a(h * 0.1)) * 0.1 # nolint
111
112     cunt_denominator <- cunt_denominator + (x *
      prior_distribution_a(h * 0.1) * 0.1)#nolint
113 }
114 estimates_a1_list <- c(estimates_a1_list, c(cunt_numerator /
      cunt_denominator))
115 }
116 a1[1:100] <- estimates_a1_list[1:100]
117 return(a1)
118 }
119
120
121 f_a2_bay <- function(data, t1, t2, b, a1, a2) {
122 estimates_a2_list <- c()

```

```

123 for (j in 1:dim(data)[2]) {
124   V <- function(a2) {
125     cutt <- 0
126     for (i in 1:dim(data)[1]) {
127       if (data[i, j] == 1) {
128         cutt <- cutt + log(LP2_3(a1[j], a2, b[j], t1[i]
129                               ], t2[i]))
130       }
131       else{
132         cutt <- cutt + log((1 - LP2_3(a1[j], a2, b[j],
133                                     t1[i], t2[i])))
134       }
135     }
136     return(exp(cutt))
137   }
138   cunt_numerator <- 0
139   cunt_denominator <- 0
140   for (h in 1:40) {
141     x <- V((h) * 0.1)
142     cunt_numerator <-cunt_numerator+(((h * 0.1) ) * x *
143                                     prior_distribution_a(h * 0.1) * 0.1) # nolint
144   }
145   cunt_denominator <- cunt_denominator + (x *
146                                     prior_distribution_a(h * 0.1) * 0.1)#nolint
147 }
148 estimates_a2_list <- c(estimates_a2_list, c(cunt_numerator /
149 cunt_denominator))
150 }
151 a2[1:100] <- estimates_a2_list[1:100]
152 return(a2)
153 }
154
155 f_b_bay <- function(data, t1, t2, a1, a2, b) {
156   estimates_b_list <- c()
157   for (j in 1:dim(data)[2]) {
158     T <- function(b) {
159       cutt <- 0
160       for (i in 1:dim(data)[1]) {
161         if (data[i, j] == 1) {
162           cutt <- cutt + log(LP2_3(a1[j], a2[j], b, t1[i]
163                                   ], t2[i]))
164         }
165         else{
166           cutt <- cutt + log(1 - LP2_3(a1[j], a2[j], b,
167                                       t1[i], t2[i]))
168         }
169       }
170     }
171   }
172 }

```

```

163         }
164         return(exp(cutt))
165     }
166     cunt_numerator <- 0
167     cunt_denominator <- 0
168     for (h in 1:40) {
169         x <- T(-4 + (h - 1) * 0.2)
170         cunt_numerator <- cunt_numerator + ((-4 + (h-1) * 0.2) * x *
171             prior_distribution_b(-4 + (h-1) * 0.2) * 0.2)#nolint
172         cunt_denominator <- cunt_denominator + (x *
173             prior_distribution_b(-4 + (h-1) * 0.2) * 0.2)#nolint
174     }
175     estimates_b_list <- c(estimates_b_list, c(cunt_numerator /
176         cunt_denominator))
177 }
178
179
180 experiment_multi_bay_1 <- function(data, epsilon, e) {
181     num_k <- 2
182     num_users <- dim(data)[1]
183     num_items <- dim(data)[2]
184
185     theta_mat <- matrix(0, nrow = num_users, ncol = num_k)
186     theta_mat[, 1] <- seq(-2, 2, length = num_users)
187     theta_mat[, 2] <- cbind(seq(-1, -1, length = (num_users / 2)),
188         seq(1, 1, length = (num_users / 2)))
189     t1 <- theta_mat[, 1]
190     t2 <- theta_mat[, 2]
191
192     a1_2 <- c(seq(0.5, 2, length = num_items / 2), seq(0.5, 2, length
193         = num_items / 2))
194     a2_2 <- c(seq(0.5, 1, length = num_items / 2), seq(0.5, 1, length
195         = num_items / 2))
196     b_2 <- seq(-1, 1, length = num_items)
197
198     for (i in 1:100) {
199         b_0 <- f_b_bay(data, t1, t2, a1_2, a2_2, b_2)
200         a1_0 <- f_a1_bay(data, t1, t2, b_0, a1_2, a2_2)
201         a2_0 <- f_a2_bay(data, t1, t2, b_0, a1_0, a2_2)
202         t1_0 <- f_t1_bay(data, t2, b_0, a1_0, a2_0)
203         t2_0 <- f_t2_bay(data, t1, b_0, a1_0, a2_0)
204         difference_t1 <- max(abs(t1_0 - t1))
205         difference_t2 <- max(abs(t2_0 - t2))

```



```

204     difference_a1 <- max(abs(a1_0 - a1_2))
205     difference_a2 <- max(abs(a2_0 - a2_2))
206     difference_b <- max(abs(b_0 - b_2))
207     if (difference_t1 < epsilon & difference_t2 < epsilon &
        difference_a1 < epsilon & difference_a2 < epsilon &
        difference_b < epsilon) { # nolint
208         print(i)
209         break ()
210     }
211     t1 <- t1_0
212     t2 <- t2_0
213     a1_2[1:100] <- a1_0[1:100]
214     a2_2[1:100] <- a2_0[1:100]
215     b_2[1:100] <- b_0[1:100]
216     print(c(i))
217     setTxtProgressBar(pb, i)
218 }
219 print(e)
220 return(c(t1, t2, a1_2, a2_2, b_2))
221 }
222
223 ave_cul <- function(e) {
224     t1_lis <- rep(0, length = num_users)
225     t2_lis <- rep(0, length = num_users)
226     a1_lis <- rep(0, length = num_items)
227     a2_lis <- rep(0, length = num_items)
228     b_lis <- rep(0, length = num_items)
229     a1_2 <- c(seq(0.5, 2, length = num_items / 2), seq(0.5, 2, length
        = num_items / 2))
230     a2_2 <- c(seq(0.5, 1, length = num_items / 2), seq(0.5, 1, length
        = num_items / 2))
231     b_2 <- seq(-1, 1, length = num_items)
232
233     theta_mat <- matrix(0, nrow = num_users, ncol = num_k)
234     theta_mat[, 1] <- seq(-2, 2, length = num_users)
235     theta_mat[, 2] <- cbind(seq(-1, -1, length = (num_users / 2)),
        seq(1, 1, length = (num_users / 2)))
236     t1_2 <- theta_mat[, 1]
237     t2_2 <- theta_mat[, 2]
238     reaction_3_n <- matrix(0, nrow = num_users, ncol = num_items)
239     for (j in 1:num_users) {
240         for (i in 1:num_items) {
241             num <- LP2_3(a1_2[i], a2_2[i], b_2[i], t1_2[j], t2_2[j])
242             reaction_3_n[j, i] <- rbinom(1, 1, num)
243         }
244     }
245     data_3_n <- reaction_3_n

```

```

246 for (z in 1:e) {
247     print(z)
248     result <- experiment_multi_bay_1(data_3_n, 0.001, e)
249     t1_lis <- t1_lis + result[1:100]
250     t2_lis <- t2_lis + result[101:200]
251     a1_lis <- a1_lis + result[201:300]
252     a2_lis <- a2_lis + result[301:400]
253     b_lis <- b_lis + result[401:500]
254 }
255     return(list(t1_lis / rep(e, length = length(t1_lis)),
256                t2_lis / rep(e, length = length(t2_lis)),
257                a1_lis / rep(e, length = length(a1_lis)),
258                a2_lis / rep(e, length = length(a2_lis)),
259                b_lis / rep(e, length = length(b_lis)),
260                write.csv(t1_lis, "t1_1_lis.csv"),
261                write.csv(t2_lis, "t2_1_lis.csv"),
262                write.csv(a1_lis, "a1_1_lis.csv"),
263                write.csv(a2_lis, "a2_1_lis.csv"),
264                write.csv(b_lis, "b_1_lis.csv")))
265 }

```
