

# レポート提出票

科目名: 情報工学実験3

実験課題名: 教育システム開発-項目反応理論

実施日: 2023年 5月 18日

学籍番号: 4619027

氏名: キムドヒョン

共同実験者:

_____	_____
_____	_____
_____	_____
_____	_____

# 1 実験の要旨

情報工学実験2では、統計モデルを用いた分析におけるパラメータ推定について、項目反応理論を題材に Excel 上で学習・演習を行った。この実験では、e-Testing システムの実装については取り扱わず、理論的な学習に重点を置いた。しかし、今回は実践的な経験を積むために、項目反応理論を実装し、シミュレーションを行う。

# 2 実験の目的

今回の実験では、実験2の復習も兼ねて項目反応理論の関数を JavaScript を用いて、問題の特性や問題数が推定制度に与える影響をシミュレーションにて確認する。

# 3 実験の理論

項目反応理論 (IRT) は新しいテスト理論で、古典的なテスト理論である正答回数だけでなく、受験者の能力値  $\theta$  や問題の品質  $a$ ・難易度  $b$ などを考慮する。IRT は確率論を基礎としており、問題の特性や能力値の推定にはベイズの定理が使われる。IPA 情報処理技術者試験をはじめとする多くの国家試験では、e-Testing 化が進んでおり、IRT の導入がますます進むことが予想される。この本では、基本的な「2パラメータロジスティックモデル」について学ぶ。

## 3.1 問題と受験者の特性

ある能力値  $\theta(\theta \in \mathbb{R})$  を持つ受験者の存在を仮定する。このとき、受験者が問題  $i$  に正答する確率  $P_i(1 | \theta)$  は、IRT によると式 (1) のようにモデル化されている。

$$P_i(1 | \theta) = P(1 | \theta, a_i, b_i) = \frac{1}{1 + \exp(-Da_i(\theta - b_i))} \tag{1}$$

ここで、 $P_i(1 | \theta)$  は、受験者の能力が  $\theta$  であった場合に問題に対して正答 ( $x = 1$ ) する条件付き確率を表す。この関数は IRT では項目反応関数と呼ばれる。また、図1のような曲線を項目特性曲線 (ICC: Item Characteristic Curve) と呼ぶ。

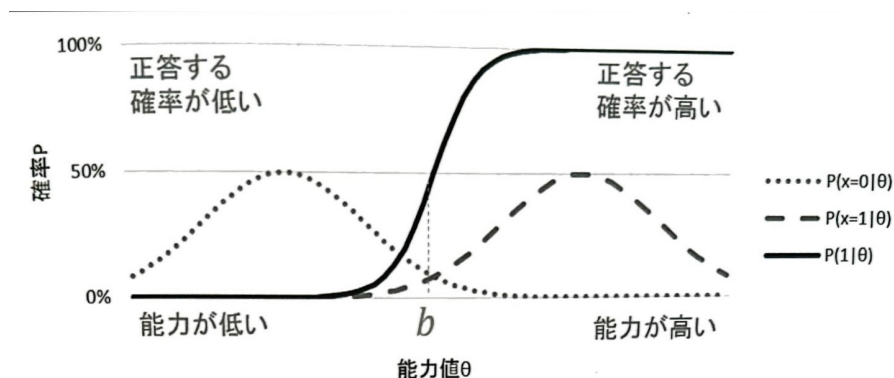


図 1: 項目特性曲線 (ICC)

$a(a > 0)$  は識別力パラメータ、 $b(b \in \mathbb{R})$  は困難度パラメータと呼ばれる。 $a_i$  が小さい問題  $i$  の回答  $x$  においては、能力  $\theta$  や難易度  $b_i$  によらず一定の正答率しか得ることができない。 $a$  は非線形 SVM でいう  $\gamma$  に相当し、小さければ項目の機能は低下するが、大きすぎると項目数が少ない場合に汎化誤差が高まりやすくなる。そのため、 $a$  は  $0.3 < a < 1.7$  程度であることが望ましいとされている。 $a$  と  $b$  は、 $\theta$  が明らかな場合にのみロジスティック回帰や線形 SVM を用いて求めることができる。 $\theta$  が不明な場合は EM アルゴリズムを用いた周辺最尤推定を使用する。 $D$  は項目反応関数  $P_i(1 | \theta)$  を正規分布の累積分布関数に近似するための係数であり、 $D = 1.7$  のとき非常によく近似することが知られている。

また、式 (1) は正答する確率を表している。ここで、ある問題に正答した場合を 1、誤答した場合を 0 とする変数  $x$  があるとする。すると、反応確率  $P_i(1 | \theta)$  は式 (2) のように表される。

$$P_i(1 | \theta) = P_i(1 | \theta)^x (1 - P_i(1 | \theta))^{1-x} \quad (2)$$

後述するベイズの定理では、これを尤度関数と呼称する。

### 3.2 能力値の推定

問題系列について正誤反応  $\mathbf{X} = (x_1, x_2, \dots, x_N)$  が与えられたとする。この時の能力値  $\theta$  は次式 (3) によって推定することができる。

$$\hat{\theta} = \arg \max_{\theta} P(\theta | \mathbf{X}) \quad (3)$$

$P(\theta | \mathbf{X})$  は、ある能力値  $\theta$  の受験者がそれぞれの問題  $i$  に  $\mathbf{X} = (x_1, x_2, \dots, x_N)$  のように反応したという条件の下で、その受験者の能力値  $\theta$  である確率を示す。 $\max$  は  $P(\theta | \mathbf{X})$  の最頻値であり、 $\operatorname{argmax}$  は尤も確率  $P$  が高くなる引数  $\theta$  の集合を表す。

ただし、IRT において  $P(\theta | \mathbf{X})$  は明らかでない。そこで、式 (4) に示すベイズの定理を用いる。

$$P(\theta | \mathbf{X}) = \frac{P(\mathbf{X} | \theta)P(\theta)}{P(\mathbf{X})} \quad (4)$$

ベイズ統計学においては、 $\theta$  を仮定（原因）、 $\mathbf{X}$  を結果（データ）と解釈する。また、 $P(\theta)$  を事前確率、 $P(\mathbf{X} | \theta)$  を尤度関数、 $P(\mathbf{X})$  を周辺尤度、 $P(\theta | \mathbf{X})$  を事後確率と呼称することがある（図 2）。

ここで、ある項目が他の項目の解答やヒントにならず、かつ問題群が一つ科目（例えば数学のみ）で構成されとした場合、尤度関数  $P(\mathbf{X} | \theta)$  は以下のように求められる。

$$P(\mathbf{X} | \theta) = P_1(x_1 | \theta)P_2(x_2 | \theta) \cdots P_N(x_N | \theta) = \prod_{i=1}^N P(x_i | \theta) = \prod_{i=1}^N P(x_i | \theta, a_i, b_i) \quad (5)$$

このような「潜在特性値  $\theta$  を固定したとき、異なる項目への反応  $x$  が統計的に互いに独立になる性質」を局所独立性と呼ぶ。この条件を仮定することで、項目間の正誤反応を同時確率とでできる。仮定の成立を積極的に評価するのは難しいが、主成分分析を用い、項目に対する反応の無相関性と、受験者の反応の一次元性を検証するケースが多い。

条件付き確率  $P(\theta | \mathbf{X})$  を尤度関数  $P(\mathbf{X} | \theta)$  のみで表現した最大引数  $\theta$  を推定値とする方法論を最尤推定、周辺尤度  $P(\mathbf{X})$  を考慮した最尤推定を最大事後確率（MAP 推定）、事前分布  $P(\theta)$  を考慮した最尤推定をベイズの推定と呼ぶ。ベイズの推定に準拠すれば以下の式 (6) で能力値が推定できる。

$$\hat{\theta} = \arg \max_{\theta} \frac{P(\mathbf{X} | \theta)P(\theta)}{P(\mathbf{X})} = \arg \max_{\theta} \left( \frac{P(\theta)}{P(\mathbf{X})} \prod_{i=1}^N P(x_i | \theta, a_i, b_i) \right) \quad (6)$$

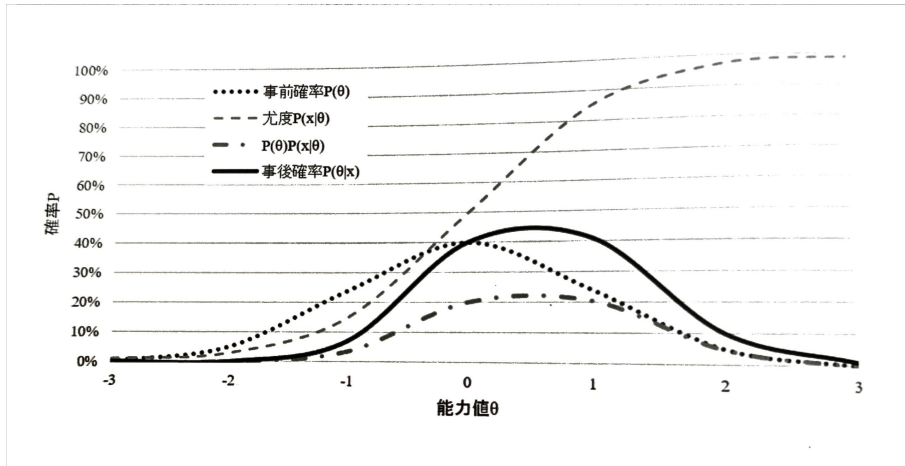


図 2: ベイズの定理における分布の変化

### 3.3 項目情報量

項目情報量とは、項目への反応から能力値を推定する差異のフィッシャー情報量である。以下のような式となる。

$$I_i(\theta) = I(\theta, a_i, b_i) = D^2 a_i^2 P(1 | \theta, a_i, b_i)(1 - P(1 | \theta, a_i, b_i)) \quad (7)$$

つまり、問題の識別力と正答確率・誤答確率の積である。さらに、複数の確率変数  $P(1 | \theta, a, b)$  がすべて独立という条件の下で、フィッシャー情報量は加法性が成立する。

$$I(\theta) = \sum_i I_i(\theta) \quad (8)$$

フィッシャー情報量は一般に最尤推定値  $\hat{\theta}$  の標準誤差  $se(\hat{\theta})$  と以下の関係を持つ。

$$se(\hat{\theta}) = I(\hat{\theta})^{-\frac{1}{2}} \quad (9)$$

そのため情報量を算出することで、推定能力値のに対する誤差を見積もることが可能である。

情報量は式 (8) のとおり、問題数が多ければ多いほど増加していくと考えられる、ただし、受験者に問題を 1000 問解かせるというのは人間工学的な問題が発生するため、実運用上は項目数の最小化と情報量の最大化という 2 つの最適化問題を解く。研究分野で言うと情報量は「テスト自動構成」と「適応テスト」において活用され、出題項目数の大幅な削減に寄与する。

情報量はアイテムバンクからテストを構成する際の基準に用いられている。図 3 は実験 2 で用いた問題集合のテスト情報量であるが、 $\theta = 1$  付近の能力値を推定するのに適したテストで

あるといえる。一般的にはナップザック問題を解くことで、試験目的に最適な項目の組み合わせを求める。例えば、テスト（項目集合）の集合を節点 $i$ 、情報量が異なるか問題重複が多いテスト間に存在する枝 $E$ 、これらで定まるグラフを $G = (V, E)$ とする。この時、グラフ $G$ の独立集合は「問題が異なるにも関わらず情報量が等質」というテスト集合となる。 $G$ の最大独立集合を求める問題を最大クリーク問題と呼び、これによって複数の等質テストを自動で構成できる。

テスト構成はテスト前に行われるが、適応テストはテスト中に適用される。適応テストは推定能力値 $\hat{\theta}$ に対して次式のように推定語さを最小化する項目を出題する。

$$i_{\text{next}} \leftarrow \arg \max_i I(\hat{\theta}, a_i, b_i) \quad (10)$$

情報量は尤度関数の微分が含まれることから、式(10)は勾配法的一种である。図4より、推定能力値 $\hat{\theta}$ が $N(b, a^{-1})$ の範囲に収まっていることがわかる。適応テストは、出題項目数を1/3まで削減することが可能であり、最近では採用試験などでも用いられている。

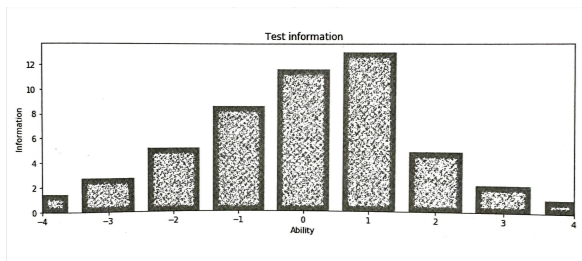


図 3: テスト情報量

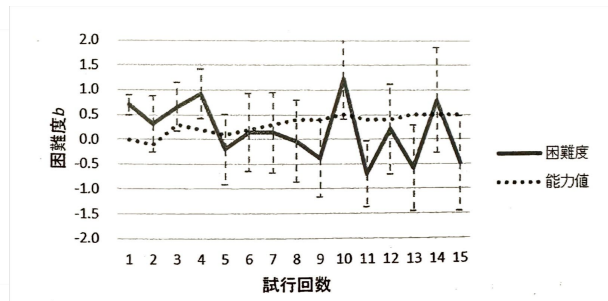


図 4: 適応テスト

## 4 課題

- 演習 1-1- 正規分布の値を出力する関数 `practice()` を作成し、`min = -3, max = 3, step = 1` における値を出力しなさい。また、値を Excel などコピーし、グラフ化した際に正規分布が表示されるか確認しなさい。
- 演習 1-2- `responseProbability(0, 1, 1, 2) ≈ 0.846`  
`icc(1, 1, 0, -3, 3, 1) ≈ (0.006, 0.032, 0.154, 0.500, 0.846, 0.968, 0.994)`  
 となることを確認しなさい。
- 演習 1-3- `bayes([1], [ a:1, b:0 ], -2, 2, 1) ≈ (0.004, 0.075, 0.403, 0.413, 0.105)`  
`estimation([1], [ a:1, b:0 ], -2, 2, 1) = 1` となることを確認しなさい。
- 演習 1-4- `information(0, [ a:1, b:0 ]) = 0.7225` となることを確認しなさい。
- 課題 1-1- 仮想実験を実施し、問題数 `Q_NUM` と受験者数 `E_NUM` が推定精度に与える影響を確認しなさい。
- 課題 1-2- 受験者の能力値を一様分布からコーシー分布に変更し、誤差について考察しなさい。

4.1 演習 1-1

表 1: min = -3, max = 3, step = 1 を入力した場合

Index	0	1	2	3	4	5	6
範囲	-3	-2	-1	0	1	2	3
出力	0.0044	0.054	0.242	0.3989	0.242	0.054	0.0044

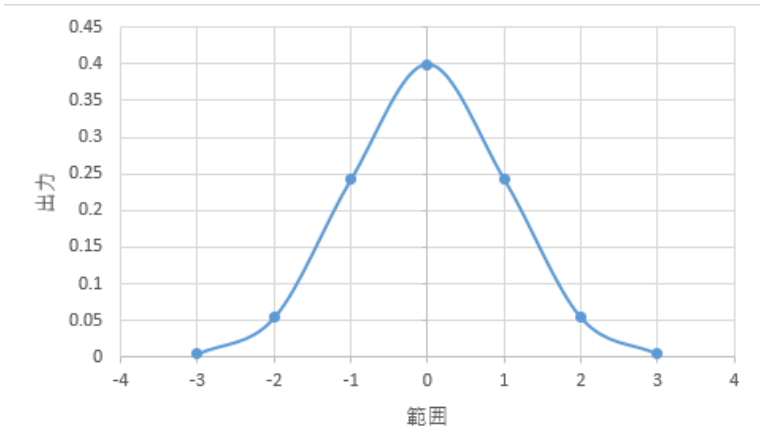


図 5: 表 1 の結果をグラフ

4.2 演習 1-2

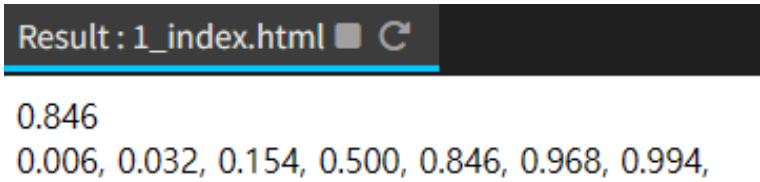


図 6: 演習 1-2 の実装結果

4.3 演習 1-3

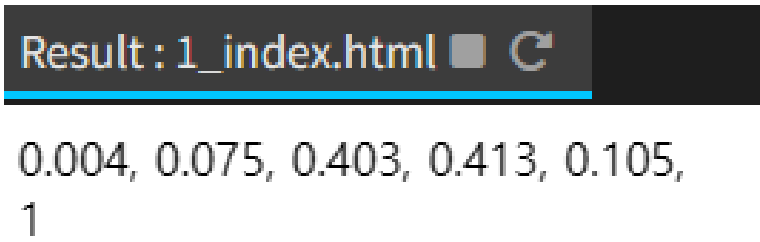


図 7: 演習 1-3 の実装結果

4.4 演習 1-4

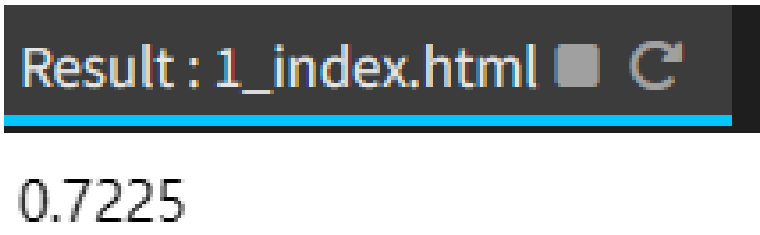


図 8: 演習 1-4 の実装結果

4.5 課題 1-1

表 2: 受験者数だけを変化したとき

Q_NUM	E_NUM	1 回目	2 回目	3 回目	4 回目	5 回目	平均
30	10	0.36566	0.439892	0.355727	0.326317	0.310163	0.359552
30	50	0.33126	0.292943	0.343432	0.264051	0.336234	0.313584
30	100	0.331752	0.301499	0.299271	0.400985	0.252495	0.317201
30	150	0.325455	0.298142	0.375762	0.304007	0.353558	0.331385
30	200	0.275015	0.267367	0.301082	0.369844	0.355432	0.313748

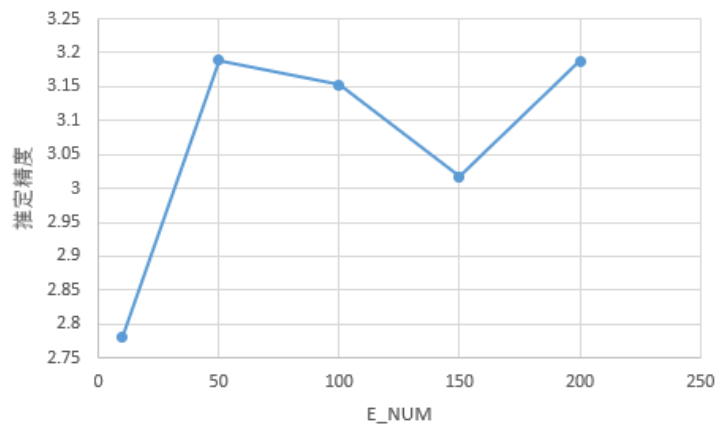


図 9: 問題数は固定し受験者数を変化した場合の推定精度

表 2 の結果をみると、E\_NUM が増加することにつれ、推定精度が変化することが確認できる。一般的に E\_NUM が増えれば推定精度も向上される傾向があった。その理由として考えられるのは、標本の大きさが増加したことと、受験者が増加して応答されたデータの多様性が増えたことなどがある。

表 3: 問題数だけを変化したとき

Q_NUM	E_NUM	1 回目	2 回目	3 回目	4 回目	5 回目	平均
30	100	0.313782	0.265795	0.323251	0.314089	0.299684	0.303320
50	100	0.252395	0.244538	0.251966	0.268183	0.209797	0.245376
70	100	0.197272	0.182054	0.201055	0.194511	0.187262	0.192431
100	100	0.161129	0.168807	0.199783	0.168558	0.201353	0.179926
150	100	0.146181	0.141304	0.122881	0.137281	0.126178	0.134765

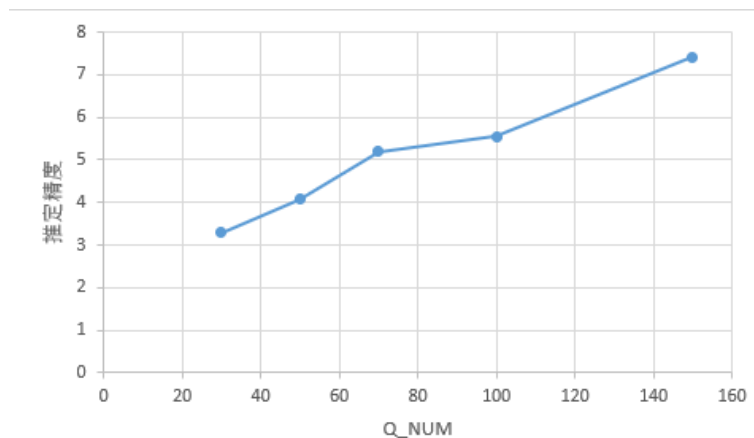


図 10: 受験者は固定し問題数を変化した場合の推定精度

表 3 は E\_NUM を固定し、Q\_NUM だけを変化したとき得られた結果である。問題数が多くなることで推定精度も高くなったことがわかる。その理由として、問題数が増加されることで受験者の能力と特性を多様に測定でき、より正確に受験者を把握できるようになりこれが推定精度の向上につながると思われる。

## 4.6 課題 1-2

表 4: コーシー分布に変えたときの結果

分布	Q_NUM	E_NUM	1 回目	2 回目	3 回目	4 回目	5 回目	平均
課題 1-1	50	100	0.33126	0.292943	0.343432	0.264051	0.336234	0.313584
コーシー分布 (n=0.1)	50	100	0.351918	0.406419	0.741365	0.288394	0.183044	0.394228
コーシー分布 (n=1)	50	100	0.796162	5.644426	1.53772	1.735447	0.663512	2.075453



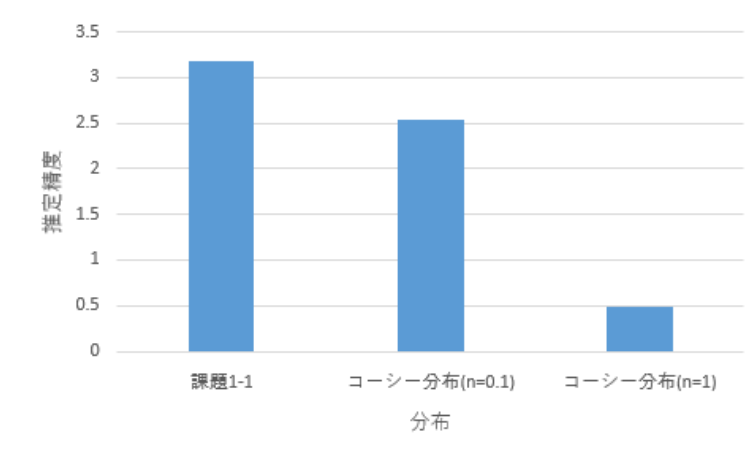


図 11: 各分布別推定精度

表 4 は受験者の能力値を一様分布からコーシー分布に変えたとき得られた平均誤差である。また、図 9 は各平均誤差の逆数をとって推定精度を計算しグラフ化したものである。コーシー分布に変えることで精度が低くなったことから一様分布を利用した方が精度が高いと思われる。しかし、コーシー分布は裾の重い分布であり、極端な値がより頻繁に生成されるため、表 4 のようにシミュレーションごとのばらつきが大きいことがわかる。またそういう特性のため、シミュレーション回数を大きくしてその平均誤差を求めれば一様分布より正確な値が得られると考えられる。したがって、コーシー分布を用いると推定の安定性は低下するが、シミュレーション回数を大きくすれば推定の精度は向上する可能性がある。

## 5 まとめ

今回の実験では、テスト理論に関連する知識を復習し、JavaScript を使用して能力値の推定を行うシミュレーションを実装した。さらに、問題の特性や問題数、受験者数などのパラメータが推定誤差にどのような影響を与えるかを確認した。

シミュレーションでは、与えられた問題の特性や受験者の応答パターンを基に、能力値を推定した。さまざまなパラメータ設定（問題数、受験者数など）において、推定誤差を評価した。これにより、問題や受験者の数が推定の精度にどのような影響を与えるかを把握することができた。

## A 付録

Listing 1: 1\_index.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
</head>

<body>
  <div id="result"></div>
  <script src="1_functions.js"></script>
</body>

</html>
```

```
1 const result = document.getElementById('result');
2 //演習 1-1
3 const norm = (theta) => {
4   return Math.exp(-theta * theta / 2) / Math.sqrt(2 * Math.PI);
5 }
6 const normDist = (min, max, step) => {
7   const dist = [];
8   for (let theta = min; theta <= max; theta += step)
9     dist.push(norm(theta) * step);
10  return dist;
11 }
12
13 const practice1 = () => {
14   normDist(-3, 3, 1).forEach(value => {
15     result.innerHTML += value.toFixed(4) + ', ';
16   })
17 }
18 practice1();
```

Listing 2: 演習 1-1

```
1 const result = document.getElementById('result');
2 //演習 1-2
3 const correctProbability = (theta, a, b) => {
4   return 1 / (1 + Math.exp(-1.7 * a * (theta - b)));
5 }
6 const responseProbability = (x, theta, a, b) => {
7   const p = correctProbability(theta, a, b);
8   return Math.pow(p, x) * Math.pow(1 - p, 1 - x);
9 }
```

```

9
10 }
11 const icc = (x, a, b, min, max, step) => {
12   const iccDist = [];
13   for (let theta = min; theta <= max; theta += step) {
14     iccDist.push(responseProbability(x, theta, a, b));
15   }
16   return iccDist;
17 }
18 const practice2 = () => {
19   result.innerHTML = responseProbability(0, 1, 1, 2).toFixed(3) + '<br>';
20   icc(1, 1, 0, -3, 3, 1).forEach(value => {
21     result.innerHTML += value.toFixed(3) + ', ';
22   })
23 }
24 practice2();

```

Listing 3: 演習 1-2

```

1 const result = document.getElementById('result');
2 //演習 1-3
3 const itemBank = [
4   { a: 1, b: 0 },
5   { a: 0.5, b: 0.3 }
6 ];
7
8 const bayes = (x, itemBank, min, max, step) => {
9   const dist = normDist(min, max, step); //P θ ()
10  x.forEach((eachX, index) => {
11    const item = itemBank[index]; // から一つの問題セットを取得$itemBank
12    const likelihoodDist = icc(eachX, item.a, item.b, min, max, step);
13    dist.forEach((_, theta, arr) =>
14      arr[theta] *= likelihoodDist[theta]); //P θ Π ()P(X θ |)
15  });
16  //周辺尤度の計算
17  const marginalLikelihood = dist.reduce((a, b) => a + b); //P θ ()
18  dist.forEach((_, theta, arr) =>
19    arr[theta] /= marginalLikelihood); //P θ Π ()P(X θ |)
20  return dist;
21 }
22
23 const argmax = arr => arr.indexOf(arr.reduce((a, b) => Math.max(a, b)));
24
25 const estimation = (x, itemBank, min, max, step) => {

```

```

26   const probability = bayes(x, itemBank, min, max, step);
27   return min + argmax(probability) * step;
28 }
29
30 const practice3A = () => {
31   bayes([1], [{ a: 1, b: 0 }], -2, 2, 1).forEach(value => {
32     result.innerHTML += value.toFixed(3) + ', ';
33   })
34 }
35 const practice3B = () => {
36   result.innerHTML = estimation([1], [{ a: 1, b: 0 }], -2, 2, 1);
37 }
38 practice3A();
39 practice3B();

```

Listing 4: 演習 1-3

```

1  const result = document.getElementById('result');
2  //演習 1-4
3  const information = (theta, itemBank) => {
4    let info = 0;
5    itemBank.forEach(item => {
6      const p = correctProbability(theta, item.a, item.b);
7      info += 1.7 * 1.7 * item.a * item.a * p * (1 - p);
8    });
9    return info;
10 }
11
12 const standardError = (theta, itemBank) => {
13   return 1.0 / Math.sqrt(information(theta, itemBank));
14 }
15
16 const practice4 = () => {
17   result.innerHTML = information(0, [{ a: 1, b: 0 }]).toFixed(4);
18 }
19 practice4();
20

```

Listing 5: 演習 1-4

```

1  const result = document.getElementById('result');
2  /**
3   * シミュレーション実験
4   * @param {number} Q_NUM 問題数
5   * @param {number} E_NUM 受験者数

```

```

6  * @return {number} 平均誤差
7  */
8  const simulation = (Q_NUM, E_NUM) => {
9    //アイテムバンク生成
10   const itemBank = [];
11   for (let i = 0; i < Q_NUM; i++) {
12     itemBank.push({
13       a: Math.random() * 2, //[0,2)
14       b: (Math.random() - 0.5) * 6 //[-3,3)
15     });
16   }
17
18   //受験者生成
19   const examinee = [];
20   for (let e = 0; e < E_NUM; e++) {
21     examinee.push((Math.random() - 0.5) * 6); //[-3,3)
22   }
23
24   //受験者ごとの誤差
25   const error = [];
26   for (const theta of examinee) {
27     const x = []; //正誤
28     for (const item of itemBank) {
29       x.push(correctProbability(theta, item.a, item.b) > Math.random() ? 1 : 0);
30     }
31     error.push(Math.abs(theta - estimation(x, itemBank, -3, 3, 0.1)));
32   }
33
34   //平均誤差
35   return error.reduce((a, b) => a + b) / E_NUM;
36 }
37
38 result.innerHTML = '平均誤差: ' + simulation(50, 100);

```

Listing 6: 課題 1-1

```

1  const result = document.getElementById('result');
2
3  const cauchy = (x0, eta) => {
4    return x0 + eta * Math.tan(Math.PI * (Math.random() - 0.5));
5  }
6
7  /**
8   * シミュレーション実験

```

```

9  * @param {number} Q_NUM 問題数
10 * @param {number} E_NUM 受験者数
11 * @return {number} 平均誤差
12 */
13 const simulation = (Q_NUM, E_NUM, eta) => {
14   //アイテムバンク生成
15   const itemBank = [];
16   for (let i = 0; i < Q_NUM; i++) {
17     itemBank.push({
18       a: Math.random() * 2, //[0,2)
19       b: (Math.random() - 0.5) * 6 //[-3,3)
20     });
21   }
22
23   //受験者生成コーシー分布()
24   const examinee = [];
25   for (let e = 0; e < E_NUM; e++) {
26     examinee2.push(cauchy(0, eta));
27   }
28   //受験者ごとの誤差
29   const error = [];
30   for (const theta of examinee) {
31     const x = []; //正誤
32     for (const item of itemBank) {
33       x.push(correctProbability(theta, item.a, item.b) > Math.random() ? 1 : 0);
34     }
35     error.push(Math.abs(theta - estimation(x, itemBank, -3, 3, 0.1)));
36   }
37
38   //平均誤差
39   return error.reduce((a, b) => a + b) / E_NUM;
40 }
41
42 result.innerHTML = '平均誤差: ' + simulation(50, 30);

```

Listing 7: 課題 1-2