レポート提出票

科目名: 情報工学実験3

実験課題名: 課題3　教育システム

実施日: 2024年 6月 6日

学籍番号: 4622045

氏名: 小澤　翼

共同実験者:

1　要旨

項目反応理論を実装し、シミュレーションを行ったことで分布の違いで結果が変わってくることを理解した。

2　目的

JavaScript の扱い方を学び、項目反応理論を実装し、シミュレーションを行う。

3　理論

3.1　IRT

項目反応理論（Item Response Theory, IRT）は、教育測定や心理測定の分野で広く用いられる理論であり、テストやアンケートの項目と受験者の能力（または特性）との関係をモデル化するための枠組みである。IRT は、受験者の潜在能力を推定するために使用され、各項目の特性を評価するためにも役立つ。

3.1.1　基本概念

IRT の基本的な考え方は、テスト項目への正答確率が受験者の能力と項目の特性に依存するというものである。受験者の能力を θ（シータ）で表し、各項目の特性をパラメータとして表している。

3.1.2　IRT モデル

IRT にはいくつかの異なるモデルがありますが、最も基本的なものは以下の通りである。

1-パラメータ・ロジスティックモデル（1PL）

- **難易度パラメータ**（b）：各項目の難易度を表している。
- 正答確率は以下のように表される。

$$P(X=1|\theta,b)=\frac{1}{1+\exp(-(\theta-b))} \tag{1}$$

2-パラメータ・ロジスティックモデル（2PL）

- **難易度パラメータ**（b）と**識別力パラメータ**（a）：識別力は項目が受験者の能力をどれだけ区別できるかを示している。
- 正答確率は以下のように表される。

$$P(X=1|\theta,a,b)=\frac{1}{1+\exp(-a(\theta-b))} \tag{2}$$

3-パラメータ・ロジスティックモデル（3PL）

- **難易度パラメータ**（b）、**識別力パラメータ**（a）、**推測パラメータ**（c）：推測パラメータは項目に正答する最低限の確率を示している。
- 正答確率は以下のように表される。

$$P(X=1|\theta,a,b,c)=c+\frac{1-c}{1+\exp(-a(\theta-b))} \tag{3}$$

3.1.3　項目特性曲線（ICC）

項目特性曲線（Item Characteristic Curve, ICC）は、受験者の能力に対する項目の正答確率を示す曲線である。これにより、各項目の難易度や識別力、推測の影響を視覚的に理解することができる。図 1 はその 1 例である。

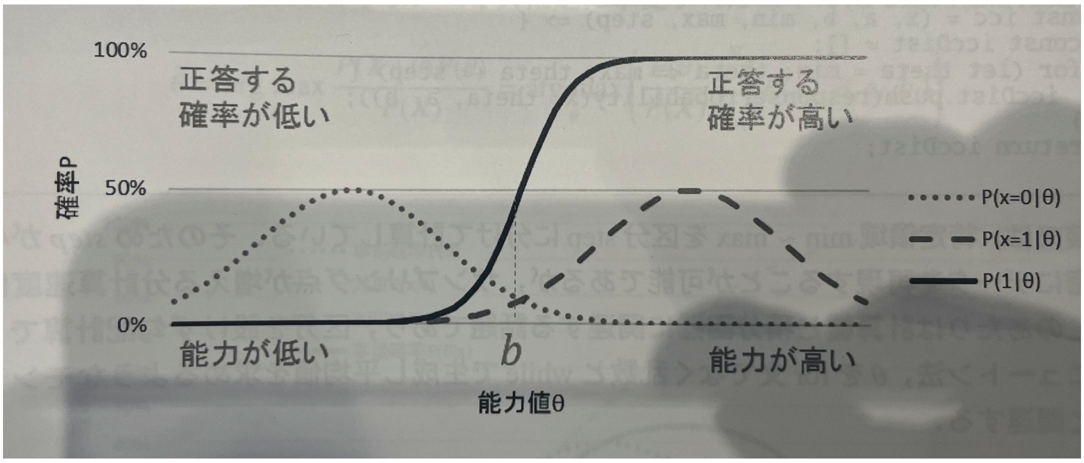

図 1: 項目特性曲線（ICC）

3.2　能力値推定

項目反応理論（Item Response Theory, IRT）では、受験者の潜在能力（θ）を推定するために、テスト項目への反応データが用いられる。能力値推定は、各受験者の能力がどの程度であるかを数値的に表現することを目的としている。

3.2.1　尤度関数

IRT における能力値推定の基本概念は尤度関数（Likelihood Function）である。尤度関数は、与えられた能力値 θ のもとで、観測された反応データが得られる確率を表している。具体的には、受験者の反応パターン $X=(x_1,x_2,\ldots,x_N)$ に対する尤度関数は次のように表される。

$$L(\theta)=P(X|\theta)=\prod_{i=1}^{N}P(x_i|\theta) \tag{4}$$

ここで、$P(x_i|\theta)$ は、受験者の能力値 θ に対する項目 i への反応 x_i の確率である。

3.2.2　最尤推定法（MLE）

最尤推定法（Maximum Likelihood Estimation, MLE）は、尤度関数を最大化する能力値 θ を推定する方法である。具体的には、尤度関数 $L(\theta)$ を最大化する θ を求める。通常、対数尤度関数を用いて計算が行われる。

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \ln L(\theta) \tag{5}$$

3.2.3　ベイズ推定

ベイズ推定は、事前分布（Prior Distribution）と観測データに基づいて事後分布（Posterior Distribution）を求め、能力値 θ を推定する方法である。ベイズの定理に基づき、事後分布は次のように表される。

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} \tag{6}$$

ここで、$P(\theta)$ は事前分布、$P(X|\theta)$ は尤度、$P(X)$ はデータの周辺尤度である。事後分布 $P(\theta|X)$ は、観測データ X に基づいて θ の確率分布を更新している。

ベイズ推定では、事後分布の期待値や最頻値を能力値の推定値として用いることが一般的である。事後分布の期待値 $E[\theta|X]$ は次のように計算される。

$$E[\theta|X] = \int \theta P(\theta|X) d\theta \tag{7}$$

また、事後分布の最頻値は次のように計算される。

$$\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|X) \tag{8}$$

ここで、$\hat{\theta}_{MAP}$ は最尤推定法における推定値と同様に、観測データに基づいて最も尤もらしい能力値を示している。

4 課題

4.1 演習１ー１

正規分布の値を出力する関数 practice1() 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 などにコピーし，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。

結果は、以下の図 2 である。

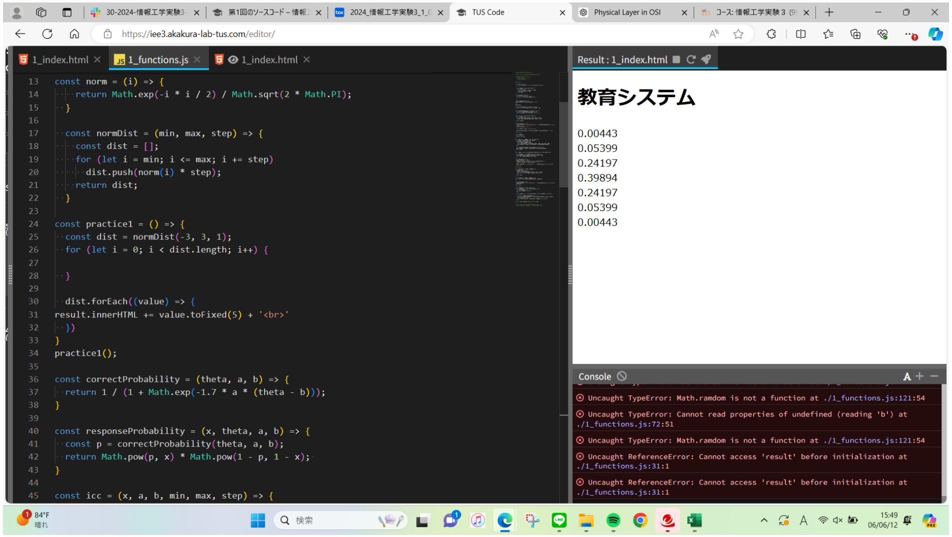

図 2: 演習１ー１の結果

4.2 演習１ − 2

responseProbability(0,1,1,2)≃0.846 と 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。

結果は、以下の図 3 である。

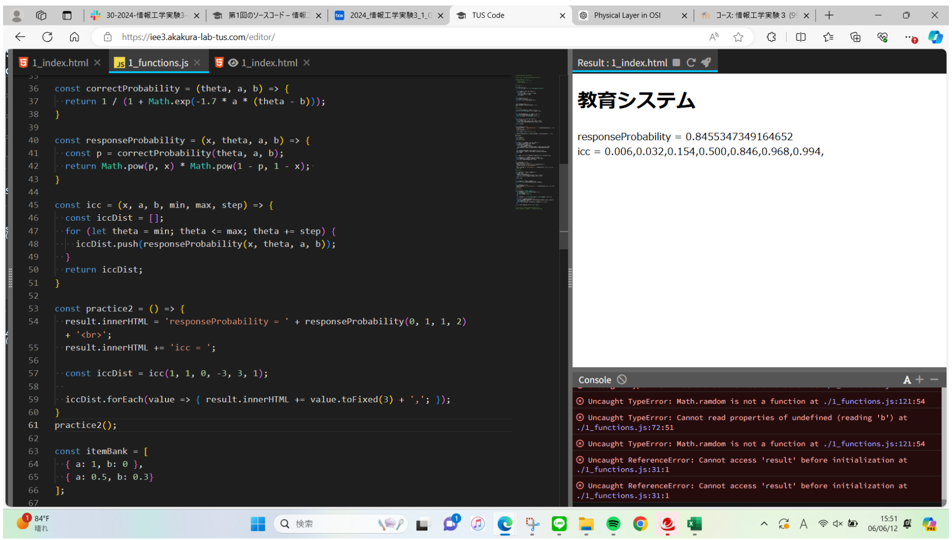

図 3: 演習１ー２の結果

4.3　演習１－３

bayes([1], [a: 1, b: 0], -2, 2, 1)≃(0.004, 0.075, 0.403, 0.413, 0.105) と estimation([1], [a: 1, b: 0], -2, 2, 1)=1 となることを確認しなさい。

結果は、以下の図 4 である。

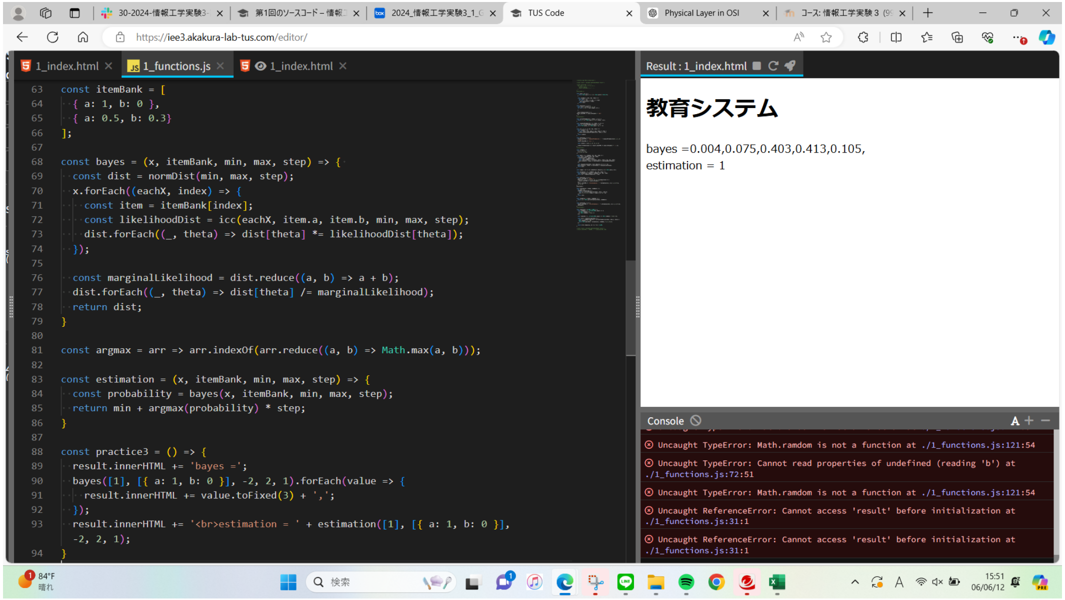

図 4: 演習１ー３の結果

4.4　演習１－４

infomation(0, [a: 1, b: 0])=0.7225 となることを確認しなさい。

結果は、以下の図 5 である。

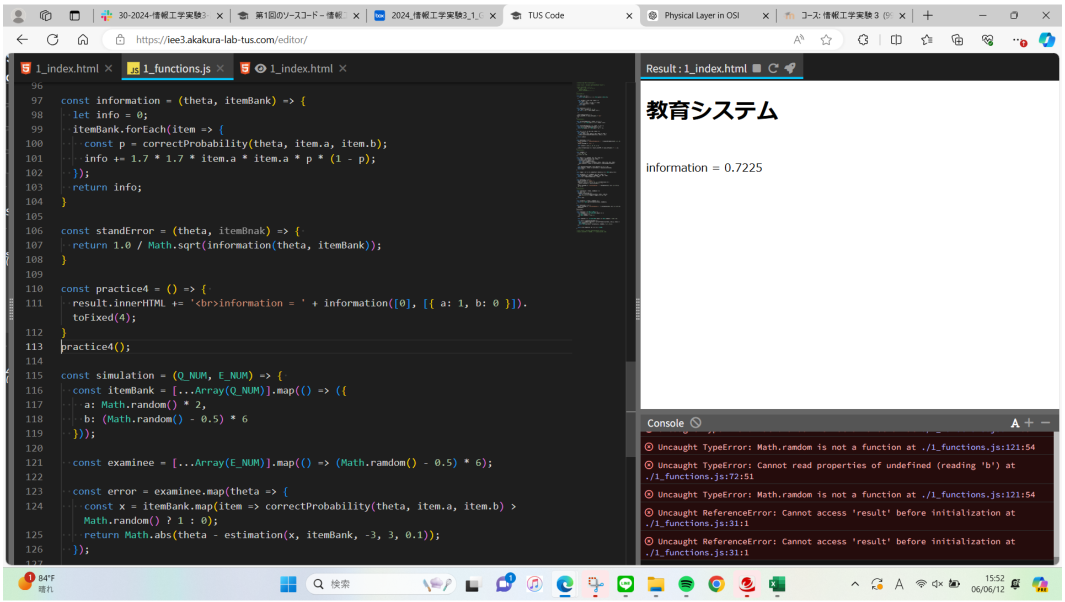

図 5: 演習１ー４の結果

4.5 課題１ー１

仮想実験を実施し、問題数 Q_NUM と受験者数 E_NUM を推定精度に与える影響を確認しなさい。

結果は、以下の表 1,2、図 6,7 である。

表 1: 受験者数を 50 に固定したとき

受験者数	問題数	1 回目	2 回目	3 回目	4 回目	5 回目	平均
50	10	0.7760	0.5230	0.5670	0.4940	0.6232	0.5970
50	50	0.2176	0.2842	0.2259	0.2138	0.2274	0.2338
50	100	0.1737	0.1680	0.1241	0.1669	0.1610	0.1587
50	150	0.1376	0.1557	0.1244	0.1491	0.1155	0.1364
50	200	0.1227	0.1273	0.0981	0.0927	0.1082	0.1098

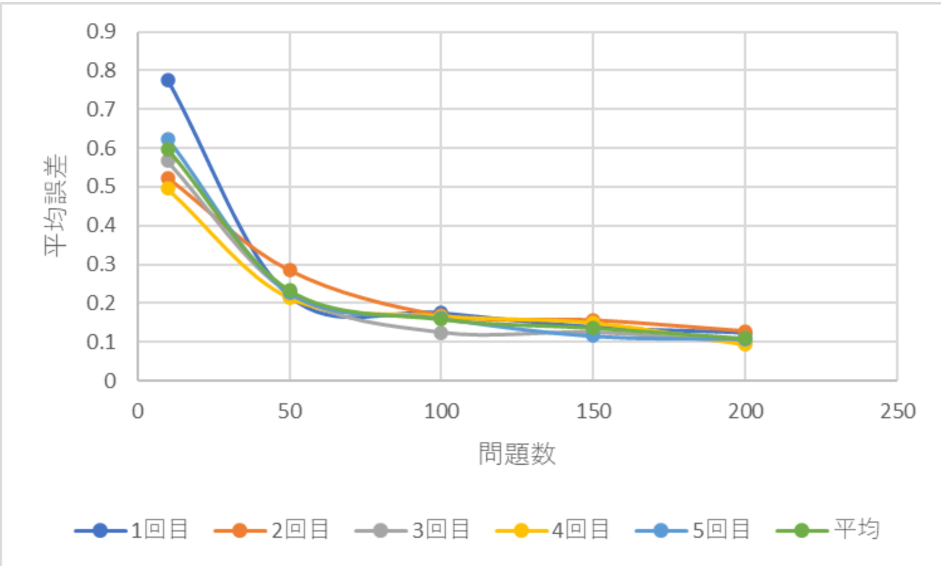

図 6: 受験者数を 50 に固定したとき

表 2: 問題数 100 にを固定したとき

問題数	受験者数	1 回目	2 回目	3 回目	4 回目	5 回目	平均
100	30	0.1514	0.1237	0.1947	0.1620	0.1577	0.1579
100	60	0.1543	0.1889	0.1711	0.1882	0.1534	0.1712
100	90	0.1607	0.1595	0.1542	0.1791	0.1446	0.1596
100	120	0.1864	0.1674	0.1803	0.1839	0.1532	0.1742
100	150	0.1637	0.1644	0.1698	0.1595	0.1578	0.1631

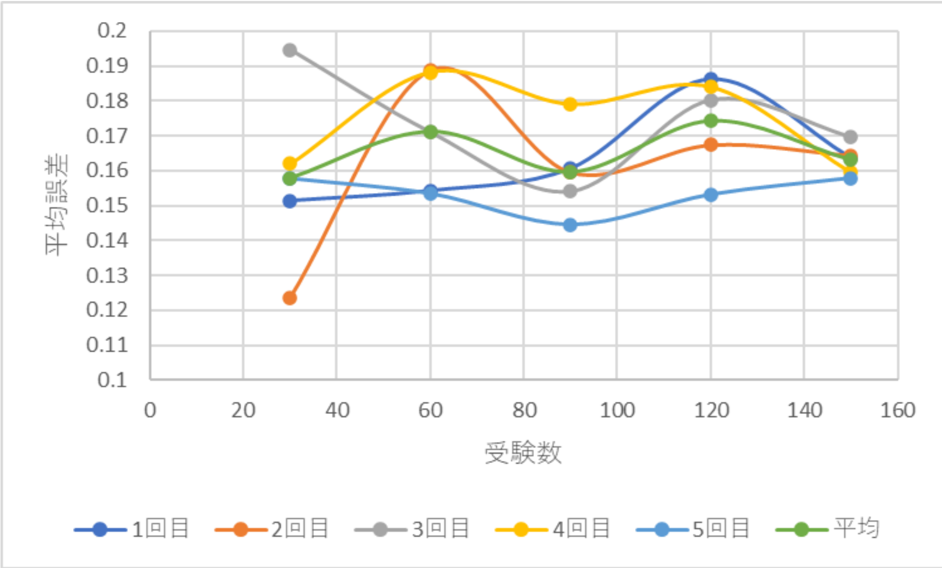

図 7: 問題数を 100 に固定したとき

受験者数を固定すると問題数が増えると誤差が減り、問題数を固定すると受験者数を固定したときに比べて誤差の変動は少ないことが分かった。

これは、大数法則に基づいて問題数が増えると誤差が減っていると考えている。グラフから問題数が 100 辺りから収束していることが分かるので、どの事象においてもデータを 100 取ると大数法則に基づいて考えることが出来るだろうと思われる。

4.6 課題１ー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。

結果は、以下の表 3、図 8 である。

表 3: 能力値をコーシー分布に変更したとき

分布	問題数	受験者数	1 回目	2 回目	3 回目	4 回目	5 回目	平均
課題 1 － 1	100	30	0.1514	0.1237	0.1947	0.1620	0.1577	0.1579
コーシー分布 ($\eta = 0.1$)	100	30	0.1342	0.1339	0.1523	0.9246	0.2640	0.3218
コーシー分布 ($\eta = 1.0$)	100	30	0.4566	0.1149	0.8698	0.1774	0.3581	0.3954

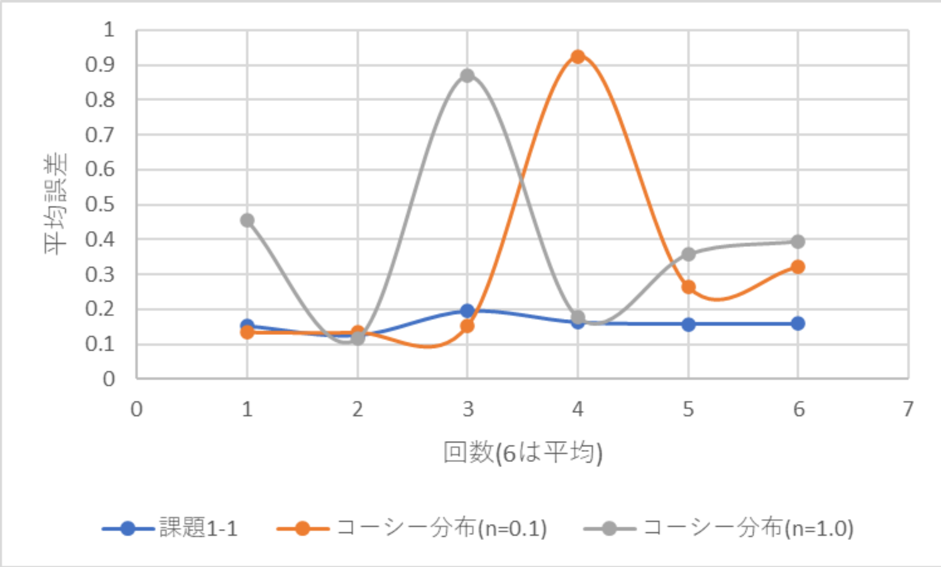

図 8: 能力値をコーシー分布に変更したとき

コーシー分布の方が η に関わらず、平均誤差がまばらで平均値も大きい結果となった。平均誤差がまばらになってしまった原因として考えられるのは、一様分布は分散が定義されており、特定の範囲内でランダムな値が等しく発生する一方で、コーシー分布は分散が定義されておらず、異常値を含んだデータとなっているため平均誤差がまばらになってしまったと考えられる。平均値が大きくなった理由もこのまばらが原因である。η が大きい方が平均誤差がまばらで平均値も大きくなっているが、これは η が大きくなればなるほど分布が広くなるからである。

4.7　課題１ー３

受験者の能力値が課題１－２で設定した分布に従う場合、テスト設計においてどのような点に注意する必要があるか、考えなさい。

η による変化が激しいため η の数値を気にしてテスト設計をする必要があると思われる。

5　まとめ

シミュレーションを実装することで、一様分布を用いるとデータ数がいくつあれば、ある程度の値に収束するか確認することが出来た。また、一様分布とコーシー分布でどちらの方が今回において優れているかや η の数値によってどのように変化するかも確認することが出来た。

A　演習１‐１

```
const norm = (i) => {
    return Math.exp(-i * i / 2) / Math.sqrt(2 * Math.PI);
  }

const normDist = (min, max, step) => {
    const dist = [];
    for (let i = min; i <= max; i += step)
      dist.push(norm(i) * step);
    return dist;
  }

const practice1 = () => {
  const dist = normDist(-3, 3, 1);
  for (let i = 0; i < dist.length; i++) {

  }

  dist.forEach((value) => {
result.innerHTML += value.toFixed(5) + '<br>'
  })
}
practice1();
```

B　演習１－２

```
const correctProbability = (theta, a, b) => {
  return 1 / (1 + Math.exp(-1.7 * a * (theta - b)));
}

const responseProbability = (x, theta, a, b) => {
  const p = correctProbability(theta, a, b);
  return Math.pow(p, x) * Math.pow(1 - p, 1 - x);
}

const icc = (x, a, b, min, max, step) => {
  const iccDist = [];
  for (let theta = min; theta <= max; theta += step) {
    iccDist.push(responseProbability(x, theta, a, b));
  }
  return iccDist;
}

const practice2 = () => {
  result.innerHTML = 'responseProbability = ' + responseProbability(0, 1, 1,
      2) + '<br>';
  result.innerHTML += 'icc = ';

  const iccDist = icc(1, 1, 0, -3, 3, 1);

```

```
  iccDist.forEach(value => { result.innerHTML += value.toFixed(3) + ','; });
}
practice2();
```

C　演習１－３

```
const itemBank = [
  { a: 1, b: 0 },
  { a: 0.5, b: 0.3}
];

const bayes = (x, itemBank, min, max, step) => {
  const dist = normDist(min, max, step);
  x.forEach((eachX, index) => {
    const item = itemBank[index];
    const likelihoodDist = icc(eachX, item.a, item.b, min, max, step);
    dist.forEach((_, theta) => dist[theta] *= likelihoodDist[theta]);
  });

  const marginalLikelihood = dist.reduce((a, b) => a + b);
  dist.forEach((_, theta) => dist[theta] /= marginalLikelihood);
  return dist;
}

const argmax = arr => arr.indexOf(arr.reduce((a, b) => Math.max(a, b)));

const estimation = (x, itemBank, min, max, step) => {
  const probability = bayes(x, itemBank, min, max, step);
  return min + argmax(probability) * step;
}

const practice3 = () => {
  result.innerHTML += 'bayes =';
  bayes([1], [{ a: 1, b: 0 }], -2, 2, 1).forEach(value => {
    result.innerHTML += value.toFixed(3) + ',';
  });
  result.innerHTML += '<br>estimation = ' + estimation([1], [{ a: 1, b: 0
      }], -2, 2, 1);
}
practice3();
```

D　演習１－４

```
const information = (theta, itemBank) => {
  let info = 0;
  itemBank.forEach(item => {
    const p = correctProbability(theta, item.a, item.b);
    info += 1.7 * 1.7 * item.a * item.a * p * (1 - p);
  });
```

```
  return info;
}

const standError = (theta, itemBnak) => {
  return 1.0 / Math.sqrt(information(theta, itemBank));
}

const practice4 = () => {
  result.innerHTML += '<br>information = ' + information([0], [{ a: 1, b: 0
      }]).toFixed(4);
}
practice4();
```

E 課題

```
const simulation = (Q_NUM, E_NUM) => {
  const itemBank = [...Array(Q_NUM)].map(() => ({
    a: Math.random() * 2,
    b: (Math.random() - 0.5) * 6
  }));

  const examinee = [...Array(E_NUM)].map(() => (Math.ramdom() - 0.5) * 6);

  const error = examinee.map(theta => {
    const x = itemBank.map(item => correctProbability(theta, item.a, item.b)
        > Math.random() ? 1 : 0);
    return Math.abs(theta - estimation(x, itemBank, -3, 3, 0.1));
  });

  return error.reduce((a, b) => a + b) / E_NUM;
}

const result = document.getElementById('result');
result.innerHTML = '平均誤差:' + simulation(50, 100)
```