2017 年度　尾山ゼミ論文

選択肢問題の妥当性

経済学部経営学科 3 年　土屋直之

1. 研究テーマ

選択肢問題は採点の容易さなどから幅広い試験で用いられており、多くの大学生が受験してきたセンター試験の殆どが選択肢問題であったし、様々な資格試験においても選択肢問題が中心である。しかし、選択肢問題だと「あてずっぽう」な解答によって正解してしまうこともあり、果たして正しく能力を図ることができているのかという点に疑問が残る。実際、「知識なしでも選択肢問題を正解する方法」などといった本や参考書も出回っており、能力を図るという本来の試験の目的が果たせていない場面も多くみられるように思われる。

そこで、こういった「テクニック」の存在を別としても、選択肢問題が「受験生の能力を図る試験」として妥当性があるのかを、講義で行ったマッチングのアルゴリズムを参考にしながら、実際にシミュレーションを行って考察し、どのような選択肢問題を作るのがよいのかを検証した。その上で、東京大学の現在の入学試験の方式の是非に関して考察を行った。

2. 先行研究

多肢選択肢問題に関する先行研究として、Haladyna, Downing, Rodriguez（2002）がある。Haladyna et al.（2002）では、選択肢の作り方に関する研究を行い、選択肢問題の項目に関するガイドラインを検証した。それ以降、実証的な研究による検証が行われ、Shizuka, Takeuchi, Yashima, Yoshizawa（2006）では、選択肢の数について 3 択と 4 択の場合を比較し、正解選択肢を含めて 3 択で十分であるとした。

以上のように、選択肢問題の研究では、どのように正解選択肢を誤答選択肢と区別できないようにするかに主眼が置かれてきた。本論文では、受験者が正解選択肢を誤答選択肢と区別できないということを前提に、Shizuka et al.（2006）と同じく、選択肢の数について 3 択と 4 択の場合を比較し、シミュレーションを行う。

3. 実装と結果

受験者が 1 つの選択肢に対して 8 割の確率で正誤を判定できるとする。これは、試験が要求する能力全体のうち、8 割を受験者が理解しているということに等しい。受験者は、正解だと分かる選択肢があればそれを選択肢、それが分からない場合には間違っていると分かる選択肢を排除し、残りの選択肢の中から「あてずっぽう」に選択するとする。

まず、3 択問題を考える。問題数を 100 とする。受験者は、正解選択肢が分かる場合はそれを選択し、正解選択肢が分からない場合でも、誤答選択肢が 2 つ分かる場合、誤答でない方を選択することで、正解することができる。そして、誤答選択肢が 1 つしか分からない場合や 1 つも分からない場合には、「あてずっぽう」に選択することになるため、前者の場合には 1/2 の確率で、後者の場合には 1/3 の確率で正解することができる。これを実装すると、次のようになる。

```
In [15]: a = 100  # 問題数
n = 3  # 選択肢数
d = 80  # 理解率

answer = Vector{Int64}(a)
answer[1:end] = 0  # 解答

k = 1:100
b = 1
while b in 1:a
    t = rand(k)
    if t <= d
        answer[b] = 1
    else
        count = 0
        m = 1
        while m in 1:n-1
            if rand(k) <= d
                count += 1
            end
            m += 1
        end
        c = 1:n-count
        if rand(c) <= 1
            answer[b] = 1
        end
    end
    b += 1
end
print(answer)
```

```
[1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

この answer が受験者の解答の一覧になっており、0 だと不正解、1 だと正解ということである。正解数の合計を見ると、

```
In [16]: sum(answer[1:end])
Out[16]: 92
```

92 となる。つまり、80%しか理解していない受験生が 92%正解することができている。これを何回か繰りしたい。そこで、上記で実装したものをさらに一般化する。

```
In [19]: function selection(a, n, d)
    k = 1:100
    answer = Vector{Int64}(a)
    answer[1:end] = 0

    b = 1
    while b in 1:a
        t = rand(k)
        if t <= d
            answer[b] = 1
        else
            count = 0
            m = 1
            while m in 1:n-1
                if rand(k) <= d
                    count += 1
                end
                m += 1
            end
            c = 1:n-count
            if rand(c) <= 1
                answer[b] = 1
            end
        end
        b += 1
    end
    return sum(answer[1:end])
end
Out[19]: selection (generic function with 1 method)
```

これを用いて、20 回ほど行うと、

In [3]:
```
q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 80)
    q += 1
end
r
```

Out[3]:
```
20-element Array{Int64,1}:
  92
  94
  96
  98
  94
  98
  96
  97
  97
  96
  96
  96
  94
 100
  96
  93
  94
  95
  97
  92
```

この平均は 95.5 となる。

In [4]:
```
mean(r)
```

Out[4]:
```
95.55
```

つまり、3 択問題の場合、80%しか理解していない受験生が、平均で 95.5%正解できてしまう。

では、受験生の理解度を変えてシミュレーションを行ってみる。いくつかの理解度で同じ処理を行う。

In [6]:
```
q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 0)
    q += 1
end
print(r)
mean(r)
```

```
[37, 46, 38, 30, 39, 31, 30, 27, 34, 31, 39, 25, 29, 41, 32, 34, 37, 40, 26, 36]
```

Out[6]:
```
34.1
```

In [7]:
```
q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 20)
    q += 1
end
print(r)
mean(r)
```

```
[48, 58, 50, 55, 52, 45, 37, 59, 53, 54, 56, 44, 51, 52, 57, 55, 52, 46, 50, 56]
```

Out[7]:
```
51.5
```

In [8]:
```
q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 40)
    q += 1
end
print(r)
mean(r)
```

```
[69, 70, 70, 72, 69, 75, 67, 71, 69, 65, 78, 74, 69, 71, 67, 71, 70, 63, 67, 74]
```

Out[8]:
```
70.05
```

In [9]:
```
q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 60)
    q += 1
end
print(r)
mean(r)
```

```
[85, 85, 85, 85, 89, 93, 84, 90, 89, 82, 82, 84, 88, 82, 91, 90, 89, 86, 88, 90]
```

Out[9]:
```
86.85
```

```
In [10]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 80)
    q += 1
end
print(r)
mean(r)

[95, 93, 97, 99, 95, 98, 95, 95, 96, 95, 98, 94, 96, 97, 99, 96, 98, 100, 98, 97]
Out[10]: 96.55
```

```
In [11]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 3, 90)
    q += 1
end
print(r)
mean(r)

[98, 97, 99, 99, 99, 99, 99, 100, 100, 100, 100, 98, 100, 100, 100, 97, 98, 98, 99, 100]
Out[11]: 99.0
```

従って、理解度の低い受験者ほど選択問題であるために正解率が高く出てしまい、逆に理解度が一定以上の受験者に関しては、差異が極めて小さくなってしまっている。

では、4 択問題で同じことを行うと、どうなるか検証してみる。結果は以下の通り。

```
In [19]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 4, 0)
    q += 1
end
print(r)
mean(r)

[25, 22, 22, 31, 25, 16, 28, 16, 35, 23, 24, 28, 36, 27, 22, 22, 23, 26, 33, 24]
Out[19]: 25.4
```

```
In [20]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 4, 20)
    q += 1
end
print(r)
mean(r)

[49, 48, 47, 49, 57, 38, 46, 45, 41, 53, 44, 44, 49, 50, 55, 44, 47, 50, 36, 37]
Out[20]: 46.45
```

```
In [21]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 4, 40)
    q += 1
end
print(r)
mean(r)

[61, 65, 69, 59, 67, 64, 67, 68, 60, 63, 60, 67, 61, 61, 63, 71, 71, 62, 67, 67]
Out[21]: 64.65
```

```
In [22]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 4, 60)
    q += 1
end
print(r)
mean(r)

[85, 88, 74, 77, 87, 80, 78, 85, 87, 81, 79, 80, 79, 81, 84, 76, 85, 74, 87, 82]
Out[22]: 81.45
```

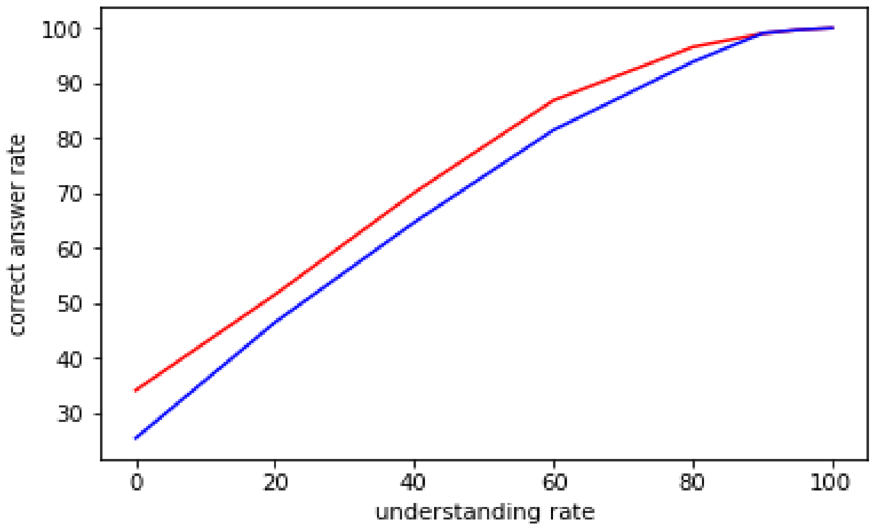
```
In [23]: q = 1
r = Vector{Int64}(20)
while q in 1:20
    r[q] = selection(100, 4, 80)
    q += 1
end
print(r)
mean(r)

[92, 98, 91, 90, 97, 91, 94, 96, 89, 94, 96, 98, 93, 93, 94, 94, 95, 97, 93, 92]
Out[23]: 93.85
```

```
In [24]: q = 1
         r = Vector{Int64}(20)
         while q in 1:20
             r[q] = selection(100, 4, 90)
             q += 1
         end
         print(r)
         mean(r)

[100, 100, 98, 98, 99, 100, 99, 100, 99, 99, 98, 100, 100, 98, 97, 99, 100, 100, 99, 98]
Out[24]: 99.05
```

グラフを描いて見比べてみると、以下のようになる（赤が 3 択、青が 4 択）。



従って、4 択問題にすることで、やや改善されるものの、依然として理解度の十分に高い受験者間での差異が小さくなってしまっている。

4. 考察

従って、受験者が正解選択肢を誤答選択肢と区別できないということを前提とした場合には、3 択よりも 4 択の方がよいことは導ける。但し、いずれの場合も選択肢問題は理解度が十分に高い受験者間では正解率の差異がなくなってしまう。これは、理解度が低い受験者層ほど「あてずっぽう」になる部分が大きくなるため、実際の実力に加わるランダムの＋αが大きくなるためである（理解度が 0%の受験者でも、3 択の場合にはおおよそ 1/3、4 択の場合にはおおよそ 1/4 程度は正解できてしまう）。一方で、理解度が十分に高い場合には、「あてずっぽう」になる部分が小さい上に、誤答選択肢をある程度減らせるため、「あてずっぽう」の正解率も高くなり、結果として理解度が 90%を超えるとほぼ結果は変わらなくなってしまう。

選択肢問題の目的が、「受験者の能力を正しく図る」ことであるとすると、理解度 d と、正解率 mean(r)が近くならなければならず、これだと妥当ではないという結論が得られる。これは「あてずっぽう」に起因する部分が大きいためである。一方で、「受験者の能力に応じて差別化が図れればよい」というだけであれば、理解度と正解率がある程度比例すればよいため、妥当性はあると言える。但し、理解度の十分に高い受験者層に関しては、ほとんど差異がなくなってしまうため、選択肢問題で能力による差別化を図ることは

できない。
以上のことを考えた際に、東京大学の入学試験の方式は合理的なものであることが分かる。まず、1 次選抜として、4 択問題の多いセンター試験を行う。この試験の目的は「足切り」であり、理解度の低い受験者を判別することが主たる目的である。従って、理解度が高い受験者の中からより理解度の高い受験者を判別するのには向いていないが、理解度の低い受験者を判別できる選択肢問題を用いるのは合理的である。そして、2 次試験では、残った理解度の十分に高い受験者の中で、受験者の理解度をより正しく判別する記述式問題が用いられるのも合理的である。

5. 追記

これを思いついたのは、単純に今年度ゼミでやったことを用いて、何かシミュレーションを行ってみたいということがスタートだったため、これを卒論に生かすとか、今後もさらにこれを深めようなどということは現状では特にないと思います。