

正誤データのためのニューラルテスト理論を用いた  
項目分析用ソフトウェア「neutet」の開発

橋本貴充

大学入試センター研究開発部試験評価解析研究部門

Department of Test Analysis and Evaluation, The National Center for University Entrance  
Examinations

正誤データのためのニューラルテスト理論を用いた

## 項目分析用ソフトウェア「neutet」の開発

橋本貴充

### 要約

ニューラルテスト理論 (Shojima, 2007a) を用いて項目参照プロファイルを計算するソフトウェア「neutet」を開発した。使用方法は次のとおりである: (1) 0-1 で表現されたデータファイルと、設定ファイルを用意する。 (2) neutet.exe をダブルクリックする。 (3) 設定ファイル名を入力する。以上の操作で、各項目の項目参照プロファイルと、各受験者の潜在ランクおよびランク・メンバーシップ・プロファイルを計算することができる。

キーワード: ニューラルテスト理論、正誤データ、ソフトウェア

## neutet — A Software for Computing Item Reference Profiles of Neural Test Theory for Binary Response Data

Takamitsu HASHIMOTO

### Abstract

"neutet" is a tool for calculation of item reference profiles using neural test theory (Shojima, 2007a). The usage is as follows: (1) Prepare a 0-1-data file and a setting file. (2) Double click neutet.exe. (3) Input a name of the setting file. Then you can obtain item reference profiles and examinees' latent ranks and rank membership profiles.

Keywords: neural test theory, binary response data, software

---

大学入試センター研究開発部試験評価解析研究部門

Department of Test Analysis and Evaluation, The National Center for University Entrance Examinations

## 1 はじめに

Shojima (2007a) のニューラルテスト理論を用いた、各項目の項目参照プロファイル、および各受験者の潜在ランクの計算を、Windows 上で容易に実行できるツール「neutet」を開発した。neutet は、最低限、入出力ファイル名さえ指定すれば標準的な分析を実行することが可能であるが、Shojima (2007a) の紹介する分析はひととおり実行可能な仕様になっている。本報告では、その使用方法および性能について説明する。

## 2 使用方法

### 2.1 概要

分析に必要なファイルは、データファイルと設定ファイルの2つである。データファイルには、正答を1、誤答を0で表し、1と0を隙間なく並べる。設定ファイルには、neutet.exeの動作の詳細を「変数 = 値」で設定する。

これら2つのファイルを用意した上で、neutet.exeをダブルクリックし、設定ファイル名を入力すれば、設定ファイルの内容に従った分析が行われる。コマンドライン上で設定ファイル名を引数として実行してもよい。

### 2.2 データファイルの書き方

データファイルには、正答を1、誤答を0で表し、1と0を隙間なく並べる。1つの行には1人分のデータを全て書き、同じ列には同じ項目の正誤が対応するようにする。例えば、5人の受験者が3項目の試験に解答したときのデータの例は、次のようになる。

データファイルの例 (1)

```
100
101
110
111
111
```

デフォルトの設定を利用する場合、空白を除く最初の列から行末までがデータとして読み込まれる。設定ファイルで databegin 変数 (p.4) や nitems 変数 (p.5) を設定すれば、任意の列から任意の列数のデータを読み込むことが可能である。例えば、次のような、6列目から12列目までの7列に正誤データが書かれたファイルからデータを読み込むとする。

データファイルの例 (2)

```
0001:1000000M10
0002:1111000M00
0003:1100000F00
0004:1111000F10
0005:1111111M00
```

この場合、設定ファイルに次のように書き込めば、6 列目から 7 列分をデータとして読み込むことができる。先頭の列は 1 列目として数える。

— 設定の例 —

```
databegin = 6  
nitems = 7
```

## 2.3 設定ファイルの書き方

設定ファイルにおいて最低限指定が必要な変数は `infile` (p.2) のみである。ただし、`outfile` (p.2) を指定しないと、出力先が標準出力になってしまうため、`neutet.exe` をダブルクリックで開く場合には、`outfile` の指定も必要である。

したがって、例えば、`neutet.exe` と同じディレクトリにある `data.dat` からデータを読み込み、項目参照プロファイルを `result.csv` に出力する場合、設定ファイルには次の 2 行を書いた設定ファイルを用意するだけでよい。

— 例 —

```
infile = data.dat  
outfile = result.csv
```

### 2.3.1 原則

設定ファイルは、1 行に 1 つの設定を「変数 = 値」の形で書く。行末にセミコロンはつけない。変数名は全て小文字である。設定ファイルにおける空白およびタブは無視される。

設定ファイルにおいて、`#`, `%`, `*` より後はコメントとして無視される。

### 2.3.2 ファイル名関係

相対パスによる指定は、`neutet.exe` からの相対パスと見なされる。ファイル名に空白を含む場合には、ダブルクォーテーションで囲む。

`infile` (必須) 指定が必要な唯一の変数である。データファイル名を指定する。

— 例 —

```
infile = "C:\Documents and Settings\hashimot\My Documents\data.dat"
```

`outfile` (初期値 `stdout`) 項目参照プロファイルを出力するファイル名を指定する。指定がない場合、または `outfile = stdout` と指定した場合、項目参照プロファイルは標準出力に出力される。例のように指定した場合、`irp.csv` というファイルに出力される。

— 例 —

```
outfile = irp.csv
```

**rankfile** 受験者個人の潜在ランクを出力するファイル名を指定する。rankfile = stdout と指定すると、標準出力に出力される。例のように指定した場合、rank.csv というファイルに出力される。指定が無い場合、受験者個人の潜在ランクは出力されない。

例

```
rankfile = rank.csv
```

**rmpfile** 受験者個人のランク・メンバーシップ・プロファイルを出力するファイル名を指定する。rmpfile = stdout と指定すると、標準出力に出力される。例のように指定した場合、rmp.csv というファイルに出力される。指定が無い場合、受験者個人の潜在ランクは出力されない。

例

```
rmpfile = rmp.csv
```

**dlim** (初期値 **comma**) outfile, rankfile, rmpfile で指定したファイルの区切り文字を指定する。指定可能な値は表 1 のとおりである。ただし、スペース区切りを指定しても、スペースが間に 1 つ入るだけで、縦位置が揃うわけではない。指定が無い場合、dlim = comma と指定したことと同じ、つまりカンマ区切りになる。

表 1: dlim 変数で指定可能な値

値	意味
comma	カンマ区切り
tab	タブ区切り
space	スペース区切り

例

```
dlim = tab
```

### 2.3.3 データファイルの内容

**idbegin** データファイルには受験者 ID を記入することも可能である。ここで指定した受験者 ID は、潜在ランクを出力する際に用いられるが、受験者 ID がない場合には潜在ランクのみが出力される。idbegin は受験者 ID が何列目から始まるか (先頭を 1 列目とする) を指定する。idbegin を省略し、idlength のみを指定した場合、idbegin = 1 として扱われる。idbegin も idlength も指定が無い場合、ID の列は無いものとして扱われる。

**idlength** 受験者 ID の長さを指定する。idlength を省略し、idbegin のみを指定した場合、idbegin で指定した列からスペースまたはタブまでを ID として読み込む。idlength を

指定した場合、スペースやタブも含めて、指定された文字数分の全ての文字が ID として読み込まれる。

例えば、次のようなデータファイルがあるとする。

データファイルの例

```
0001:1000000M10
0002:1111000M00
0003:1100000F00
0004:1111000F10
```

この 2 列目から 4 列目までの 3 列を受験者 ID として読み込みたい場合、idbegin と idlength の値をそれぞれ 2 および 3 にする。

設定の例

```
idbegin = 2
idlength = 3
```

**databegin** データが何列目から始まるか（先頭を 1 列目とする）を指定する。例えば、次のようなデータファイルがあり、databegin = 6 を指定した場合、左から 6 列目から、つまりコロンの右からデータを読み込み始める。

データファイルの例 (1)

```
0001:1000000
0002:1111000
0003:1100000
0004:1111000
0005:1111111
```

設定の例

```
databegin = 6
```

指定がない場合、受験者 ID の次の列（空白を除く）からデータを読み込む。

例えば、次のようなデータファイルがあり、idbegin = 1 と idlength = 5 を指定した場合、databegin を省略すると、データを 6 列目から読み込む。

データファイルの例 (2)

```
0001:1000000
0002:1111000
0003:1100000
0004:1111000
0005:1111111
```

次のようなデータファイルがあり、idbegin = 1 と idlength = 4 を指定した場合、databegin を省略すると、後の空白（この例では 2 列ある）を読み飛ばし、データを 7 列

目から読み込む。

データファイルの例 (3)

```
0001 1000000
0002 1111000
0003 1100000
0004 1111000
0005 1111111
```

次のようなデータファイルがあり、idbegin も idlength も指定しなかった場合、databegin を省略すると、空白（この例では5列ある）を読み飛ばし、データを6列目から読み込む。

データファイルの例 (4)

```
1000000
1111000
1100000
1111000
1111111
```

**nitems** 項目数、すなわちデータの長さを指定する。nitems を省略すると、行末までデータを読み込む。例えば、nitems = 7 と指定すれば、databegin で指定した列から7列分を、それぞれ順に第1項目の正誤、第2項目の正誤、……、第7項目の正誤として読み込む。

例

```
nitems = 7
```

### 2.3.4 計算のパラメタ

**nranks** (初期値 10) 潜在ランク数 (Shojima (2007a) の  $Q$ ) を指定する。例のように指定した場合、潜在ランク数を5として計算が行われる。指定が無い場合、潜在ランク数は10として計算が行われる。

例

```
nranks = 5
```

**method** (初期値 ml) 勝者ノードの決定方法を指定する。指定可能な値は表2のとおりである。例のように指定した場合、ユークリッド距離の2乗が最も小さくなるものが勝者ノードとして選ばれる。指定が無い場合、method = ml と指定したことと同じ、つまり最尤法が基準として用いられる。

なお、ここで ed2 を指定した場合、ランク・メンバーシップ・プロファイルには Shojima (2007b, p.6) の式 (30) で計算される値が出力され、ここで ml を指定した場合、ランク・

メンバーシップ・プロファイルには Shojima (2007b, p.6) の式 (25) で計算される値が出力される。

表 2: method 変数で指定可能な値

値	意味
ed2	ユークリッド距離の 2 乗が最も小さくなるもの
ml	対数尤度が最も大きくなるもの

例

```
method = ed2
```

**maxiter** (初期値 100) 学習のための反復計算の回数 (Shojima (2007a) の  $T$ ) を指定する。例のように指定した場合、1000 回反復される。指定が無い場合、`maxiter = 100` を指定したものとして、つまり反復回数は 100 回として扱われる。ただし、`crit` または `diffcrit` を指定した場合、基準に達した時点で (`maxiter` で指定した回数に満たなくても) 反復を終了する。

例

```
maxiter = 1000
```

**crit** 受験者の正誤反応と、それに対する第  $t$  サイクルの勝者ノードの参照ベクトルとの距離の 2 乗を、全受験者の分足し合わせたものを  $C^{(t)}$  (Shojima (2007a) の式 (8)) とすると、 $C^{(t)}$  が `crit` で指定した値を下回ったときに反復を終了する。例のように指定した場合、 $C^{(t)}$  が 400 を下回ると反復を終了する。指定しなければ、この基準での収束判定は行わない。

例

```
crit = 400
```

**diffcrit** 受験者の正誤反応と、それに対する第  $t$  サイクルの勝者ノードの参照ベクトルとの距離の 2 乗を、全受験者の分足し合わせたものを  $C^{(t)}$  とすると、 $|C^{(t)} - C^{(t-1)}|$  が `diffcrit` で指定した値を下回ったときに反復を終了する。例のように指定した場合、 $C^{(t)}$  が 1 つ前と比べて 0.1 未満しか変化しなかったとき、反復を終了する。指定しなければ、この基準での収束判定は行わない。

例

```
diffcrit = 0.1
```

**increase** (初期値 false) 項目参照プロファイルの単調増加制約をつけるかどうかを指定する。値は `true` または `false` で指定し、`true` なら単調増加制約をつけ、`false` なら単調増

加制約をつけない。指定が無ければ、`increase = false` と指定したことと同じになる。つまり、単調増加制約は無いものとして扱われる。

例

```
increase = true
```

`increase = true` とした場合、Shojima (2007a) の (L7) の方式で単調増加制約が課される。すなわち、各サイクルで学習がひととおり済んだ後、それぞれの項目における項目参照プロファイルの値を昇順にソートする。

`seed` Shojima (2007a) の (L2) において、入力順序による学習の効果をキャンセルするために、データの行ベクトルの並べ替えが行われているが、この変数を指定することにより、その乱数の種を指定することができる。値は自然数に限られる。指定がない場合、現在時刻を用いて種を生成する。

例

```
seed = 31416
```

`alpha1`, `sigma1`, `sigma0` 順に、Shojima (2007a) の  $\alpha_1$ ,  $\sigma_1$ ,  $\sigma_0$  を指定する。 $0 < \alpha_1 < 1$ ,  $0 < \sigma_0 < \sigma_1$  の実数である必要がある。

指定がない場合、`alpha1` は 0.1、`sigma1` は潜在ランク数、`sigma0` は 1 を指定したものとして扱われる。ただし、分数で指定することはできない。例えば `alpha1 = 0.33333` という指定は可能であるが、`alpha1 = 1/3` と指定することはできない。

例

```
alpha1 = 0.2  
sigma1 = 5.0  
sigma0 = 1.0
```

## 2.4 出力

### 2.4.1 項目参照プロファイル

outfile で指定されたファイルには、各行に項目、各列にノードを並べた、項目参照プロファイルが出力される。1 行目にはノード番号が、1 列目には項目番号がそれぞれ出力される。

タブ区切りで出力した、項目数 10、潜在ランク数 3 の場合の分析例は次のようになる。

例

item	1	2	3
1	0.577017	0.750439	0.894488
2	0.325599	0.549434	0.756819
3	0.17243	0.365815	0.595697
4	0.125497	0.286247	0.48978
5	0.0526553	0.113793	0.190001
6	0.0311758	0.0718639	0.121413
7	0.0359561	0.0709699	0.115834
8	0.0227798	0.0443316	0.0714382
9	0.0151587	0.0403715	0.0742907
10	0.00975085	0.0267531	0.0491738

Excel のグラフ描画機能などを利用すれば、項目参照プロファイル曲線を可視化することができる。また、項目参照プロファイルの値に重みを付けて合計すれば、テスト参照プロファイルを計算することも可能である。

### 2.4.2 受験者の潜在ランク

rankfile で指定されたファイルには、各受験者の潜在ランクが出力される。method に ed2 を指定した場合には、解答パターンとの距離が最も短い参照ベクトルのノード番号が潜在ランクとなり、method に ml を指定した場合には、最も所属確率の高いランク番号が潜在ランクとなる。

idbegin または idlength が指定されている場合には、ID に続いて、区切り文字と、潜在ランクが出力される。

潜在ランク分布は、Excel の FREQUENCY 関数などを用いて潜在ランクの度数分布表を作成することなどにより求めることが可能である。

### 2.4.3 受験者のランク・メンバーシップ・プロファイル

rmpfile で指定されたファイルには、各受験者のランク・メンバーシップ・プロファイルが出力される。1 行目にはノード番号が出力され、idbegin または idlength が指定されている場合には 1 列目に ID が出力される。method の指定により、この内容は変化する。method 変数で ed2 を指定した場合には Shojima (2007b, p.6) の式 (30) で計算される値が出力され、ml を指定した場合には Shojima (2007b, p.6) の式 (25) で計算される値が出

力される。Excel のグラフ描画機能などを利用すれば、Shojima (2007b, p.10) の Figure 3 のような図を描くことも可能である。

ID がない場合の、タブ区切りで出力した、受験者数 15、潜在ランク数 3 の場合の分析例は次のようになる。

例 ( method=ml の場合 )

1	2	3
0.892344	0.103751	0.00390488
0.728759	0.235412	0.0358287
0.820698	0.163366	0.015936
0.626536	0.284127	0.0893375
0.614594	0.316314	0.0690916
0.153135	0.373787	0.473078
0.466748	0.360247	0.173005
0.17165	0.374124	0.454226
0.0166575	0.201761	0.781582
0.626536	0.284127	0.0893375
0.0862931	0.320109	0.593598
0.136932	0.397867	0.465201
0.0438289	0.278521	0.67765
0.0192274	0.207913	0.77286
0.00446176	0.133895	0.861643

例 ( method=ed2 の場合 )

1	2	3
0.189643	0.251471	0.350118
0.104678	0.15193	0.23469
0.109805	0.164724	0.255933
0.0681653	0.0982959	0.162769
0.185726	0.204963	0.257903
0.211715	0.158102	0.132953
0.111171	0.131213	0.18485
0.171141	0.131842	0.121708
0.310543	0.218067	0.151157
0.0681653	0.0982959	0.162769
0.214146	0.164759	0.143789
0.20276	0.211067	0.252054
0.226964	0.15889	0.117831
0.285211	0.19214	0.124438
0.384459	0.290931	0.222999

### 3 プログラムについて

コンパイルは、Cygwin の GCC (version 3.4.4) を使い、-mno-cygwin オプションをつけて行った。ソースコードおよび Windows XP 用の実行ファイルは、以下の URL にて公開する。

<http://www.rd.dnc.ac.jp/~hashimot/neutet/>

ノートブック型 PC (OS: Windows XP Professional SP2, CPU: Intel Pentium M 1.6GHz, RAM: 496MB) 上で、項目数 33、受験者数約 2000 人のデータを用いて、項目参照プロファイルを 10 回計算した。各種パラメタは、勝者ノード決定方法 (method, p.5) と単調増加制約 (increase, p.6) 以外は初期値を用いた。その平均処理時間は表 3 のようになった。

表 3: 平均処理時間

勝者ノード決定方法	単調増加制約	平均処理時間 (秒)
ユークリッド距離	なし	78.1
ユークリッド距離	あり	101.6
最尤法	なし	123.1
最尤法	あり	144.6

### 引用文献

Shojima, K. (2007a). Neural test theory. 研究開発部リサーチノート RN-07-02.

Shojima, K. (2007b). Maximum likelihood estimation of latent rank under neural test model. 研究開発部リサーチノート RN-07-04.