

認知心理学会テクニカルレポート

自由分類課題によって得られた類似度データの分析

——MDSとDISTATIS——

井関龍太（大正大学）

一対比較法と自由分類課題

私からは主に課題の話とその課題を使って得られたデータをどうやって分析するかという話をさせていただきたいと思います。ちょうど先ほどの小杉先生のお話とは逆に、こういうデータを取ったんだけどどうしようといった観点で進んでいくと思います。申し遅れましたが、私の専門は認知心理学で、主に言語とか記憶の研究をしていますので、その文脈の中で多次元尺度構成法（multidimensional scaling, MDS）はもっとうまく使えるんじゃないかと思ったのが、そもそもの今回の企画のきっかけです。

マクロな観点から言いますと、いろいろな評価データを包括的に集めたいという需要は様々な文脈においてあると思います。最終的にMDSが出てくるような文脈だと、感性評価あるいは官能評価です。いろんな刺激に対してどんな感覚からどんな印象を抱くのかとか、どれがお互いに似ているか、あるいは、温かいとか冷たいといった印象を持っているかなどを知りたいということがあります。もっと応用的な文脈で言うと製品評価がありますね。パッケージやデザインの評価はよく行いますし、実際の商品を作った場合にも——さっきもお茶とか水の例が出ましたけども——お茶の中でもこれは苦味が強いねとか、これは甘味があっていいねとか、他社製品と競合するもの、あるいは新しいものとはどんな商品だろうといったときに、ここで扱うようなデータを集めるということが起こると思います。

基礎的な分野でも、同じようなデータが役に立つ場合があります。たとえば、実験刺激の選定といった場合です。目的にもよりますが、たとえば、たくさんの似たような刺激がある中から全体を代表するような刺激を選ぼうとすることがあります。実験の刺激として似たようなものばかり使ってもしかたがないので、刺激候補全体をうまく代表するものを選びたいということがあります。

これらの状況を表すとすると、基本的に求めているのは類似度データの収集と評価といえると思います。そうして類似度データをたくさん集めた後に類似度空間を構成するわけですね。分析をした後にこの類似度空間の中の項目群を利用します。あるいは軸の評価次元を解釈することができると思います。この項目や軸は類似度空間のマッピングを直接利用したり、刺激を選んだりということに使えるはずですが、さらに、この空間の中での評価次元の解釈まで進めれば、実はこれ自体が心理学の研究に、すなわち、心的過程の解明に役立つと思います。このような類似度データの収集と評価の古典的、典型的なやり方

としては、一対比較法があります。こういった心理測定の問題というのは、古典的には心理学で感覚の評価として行われていたと思います。一対比較法といえば心理学の実験計画法の授業の中でときどき出てくることだと思います。改めて言うまでもないかもしれませんが、これはその名の通り、項目のすべての対についての評価を行う方法です。A, B, Cの3つの項目が評価する対象であったとしましょう。たとえば、3種類のお茶だと思ってもらってもいいかもしれません。A, B, Cの3種類があるとする、AとBを比べたときには僕はAのお茶のほうが好きだなと評価し、今度はBとCを比べてCのほうがいい、CとAだとAのほうがいいというふうにしてすべての組み合わせについて評価することになります。今の例だと三つしか対象がないので簡単にすむのですが、このやり方で4つ、5つと項目が増えていくとどうなるかということは皆さんも想像がつくと思います。4つだとそれぞれから2つずつを組み合わせるわけだから、対の数は6になります。5つだと10になる、というふうにどんどん増えていくわけで、そうなってしまうと当然評価にかかる時間がどんどん長くなってしまいます。すると評価をしている参加者の方にも疲労とか飽きの影響が出てきます。このやり方で10個の項目を組み合わせたとすると45対できてしまうので、40個くらいを過ぎるころにはかなり疲れているし、今挙げた例のように、評価対象がお茶だとすると、お腹たぶたぶになっていますので、この一対比較法はそういう意味では実験者、参加者ともに忍耐が強いられる方法ではないでしょうか。そして、そのことは集めたデータの質と量にも潜在的に影響しているのではないかと私は考えています。

これに対して、同じようなデータを得られるもっと簡便な方法があります。それが自由分類課題（free sorting task）です。これも結構古くから使われている方法です。これは何かと言いますと、単純に項目を似ているもの同士で分ける課題です。好きなだけの数のグループを好きなように作ってくださいと教示します。ここにA, B, C, Dの4つの箱があります。それぞれの箱が評価する対象やそれを描いたカードだと思ってください。これから、このA, B, C, Dを似ていると思うもの同士に分けてみることにしましょう。まずAはまだ他と比べてないのでそのまま置いておくこととして、ではBはどこに行くかと言うとBは何となくDに似ているなと思ったので、この2つをくっつけましょう。次に、Cはどっちかと言うとAのほうに近そうだなと思ったのでAのほうに移動します。最後のEもどちらかと言うとCに近そうだなと思ったとしますね。こうすると、それぞれ

A, C, E と B と D という二つのグループができるわけです。
Figure 1 のようなやり方です。

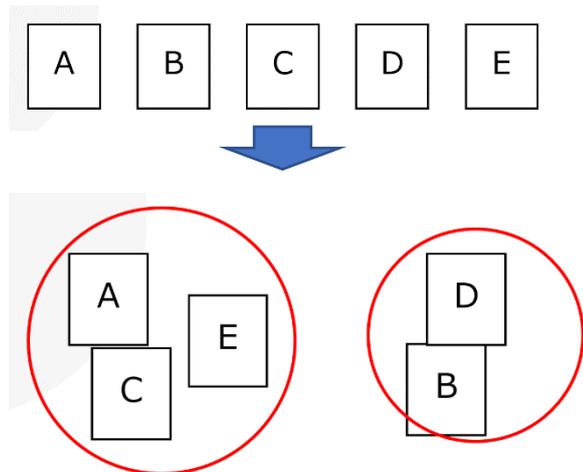


Figure 1. 自由分類の例

いまやってみたように、できるだけ制約を与えずに何でもい
いからとりあえず近いとか似ていると思ったものでまとめてく
ださいといったふうにするのが自由分類課題の一般的なやり方
です。ただ、このやり方だと、全部の項目がそれぞれ違うもの
だと思う、だから一個ずつ全部別々のグループで終わりという
人もいるので、グループの項目数は必ず 2 個以上にしてくださ
いといった制約を課すこともローカルルールとしてよくありま
す。

こうやって得られたデータは、先ほどの一対比較とはぜんぜ
ん違うじゃないかと思われるかもしれませんが、これを
数字で表してみると、実は一対比較と同じ構造のデータになっ
ていることがわかります。では、項目の組み合わせによって作
られたグループをこの 4×4 の表に配置していくことにします
(Figure 2)。どう配置するかと言うと、この表の行と列が交差
するところに、二つの項目が同じカテゴリーに分類されていれ
ば 1, そうでなければ 0 というように数値を当てはめていきま
す。まず対角線ですが、これは AA のようにお互い同じものの組
み合わせなので全部 1 です。自分自身だから違うグループに配
置されるはずがないわけですね。次に埋まってないところ……A
と B を考えてみます。先ほどの例ですと A と B は違うグルー
プに属していたので、AB は 0 ですね。本当は下三角行列だけ
で十分ですが、例なので一応全部埋めることにしましょう。とは
いえ、AB も BA も両方とも 0 になります。一方、A と C は同
じカテゴリーに先ほど分類しましたので、AC は 1 ですね。同じ
要領で全部埋めていくと下の表のようになります。そうします
と、これは A について B と何らかの点で似ているかどうかを尋
ねたときに「いいえ」、C と似ているかどうかを尋ねたときに「はい」、
D については「……」というふうに、全部の組み合わせについ

て類似度の評価を求めたのと同じ構造のデータであることがわ
かります。

	A	B	C	D	E
A	1	0	1	0	1
B	0	1	0	1	0
C	1	0	1	0	1
D	0	1	0	1	0
E	1	0	1	0	1

Figure 2. 自由分類課題のデータ構造

先ほどは項目をばつと並べて二つのカテゴリーにただけだ
ったんですが、それだけで 10 対のすべての組み合わせについ
てのデータが得られたわけです。このように、自由分類課題を使
うと手軽に大量のデータを手にすることができます。項目が 10
個あったとしても、たいていの場合は 3 つか 4 つぐらいのカテ
ゴリーになるので、それほど時間もかかりませんし、参加者と
してもそんなに負担感はありません。それでいて、できあが
たものは 45 対についても評価データになっているわけですね。

ちょっと補足的に、この手続きのバリエーションについてこ
こで説明しておきましょう。やり方としては、今説明したよ
うに自由に分けてもらうだけでも十分ですが、分けた後に参加
者にそれぞれのグループについて説明を求めることもよく行わ
れます。それぞれのグループを順に見ていって、これはどうい
う仲間ですかというのを説明してもらうわけですね。つまり、そ
れぞれのカテゴリーが何によって分けられたかをそこでインタ
ビューしておく、これはそれ自体として質的評価、参加者によ
る主観的な評価として使うこともできますし、この後に出て
くるような分析の手がかりとして使うこともできます。それか
ら、先ほどの説明ではどのぐらい似ているか、仲間だと思ふも
の同士で分けてくださいと教示したんですが、全部で三つのカ
テゴリーに分けてくださいとか、抽象的な類似度以外の基準と
して、たとえば、強さが近いと思うもの同士に分けてくださ
いなど、何らかの傾向が似ているものについて分けるように方
向づけを与えて分類させることもできます。そうすると、通常
の似ているというだけの場合とはちょっと違った形で分けてら
うことができます。最後に、分類が終わって、たとえば、3 カテ
ゴリーぐらいになった後でこれを併合して一個になるまでまと
めてもらうということもできます。三つのカテゴリーができた
後の、この 1, 2, 3 のカテゴリーのうちどちらかというとなら
ば 1 と 2 は 3 よりも近いと判断したとします。そうすると、1 と 2
を合体させたカテゴリーができるというような手続きを繰り返す
ことによって階層的に分類をさせるというやり方もできます。

他のところで説明するタイミングがないので、ここで自由分
類課題の限界と留意点についても説明しておきます。まず、項

目数が多すぎると実施が困難になるということがあります。先ほど10個ぐらいの項目があったとしてもすぐできますよと言ったのですが、適正な範囲は多分9から20ぐらいだろうとChollet et al. (2014) は述べています。実際問題として確かに項目が30も40もあるとすべてを適切に並べることは不可能になってくると思います。次に、この自由分類課題は、カードにして何かを分類させるとか、あるいはお茶とか水のように手軽に出せるようなものは現物そのものを分類させるとか、かなりいろんな使い方ができるのですが、対象として時間的に変化が起こるような対象には使いにくいということが言えます。時間的な変化にはいくつかバリエーションあるのですが、たとえば、刺激の受容に時間かかる場合がそのひとつです。ニオイなどがあるかもしれない。匂いを嗅いでからそれを感じるまでに時間がかかります。あとは動画刺激などですね。動画だと短いものでもどうしても視聴するのに何秒かかかります。5秒の動画でも10項目を視聴するには断続的に50秒かかります。また、こういった刺激の場合、記憶への依存が大きくなります。順番に受容していき、最初の方のほうがよくわからなくなっているという可能性があるわけです。もちろんもう1回受容しなおして何度でもやり直せるのなら構わないわけですが、このあと指摘する二つのいずれかの問題が起こる可能性があります。2つ目の場合として、対象そのものが時間に伴って変化する場合があります。これは主に食べ物などに当てはまります。料理だとだんだん冷めてしまい、時間が経つとおいしくなくなってしまうとか、味や風味などが時間に伴って変化する場合には、この方法はあまり適切でないといえるでしょう。最後に、参加者の方が変化する場合があります。これは、満腹になるなど、だんだん課題が進むにつれて参加者のほうの状態が変わってしまう場合です。こういう場合には、この課題で評価することはあまり適切でない可能性があります。したがって、基本的には視覚刺激、あとはものにもよりますが触覚刺激や聴覚刺激が使いやすいかなと思います。

類似度データをどう分析するか

今、簡単に自由分類課題について紹介して、この課題を使うと手軽に大量の情報が得られることがわかりました。簡単にすべての対の比較データが得られたと思って喜んでしまうわけですが、これをどう分析するかということが次の問題です。この発表ではどうするかはもうおわかりですね。手元にあるデータは、同じカテゴリーに分類されたものは1、違ったら0というふうになっているわけです。これを複数の参加者に対して複数の参加者を通して平均すると0から1までの連続量になります。1のときが最も多くの参加者が似ていると判断した対で、0のときは誰も似ていると判断しなかった対、中間の値はその中間の判断になるわけですね。そうすると、これは類似度行列のデータ構造になっています。類似度構造の分析方法としては、クラス

ター分析もありますが、MDSを使うと項目の相互の位置関係を単純な次元で表現できるはずですが、ではMDSをやってみましょう。

ここで具体的な研究を取り上げましょう。一対比較を使った研究ですが、程度量表現の評価を行った織田揮準先生の研究が教育心理学研究に掲載されています(織田, 1970)。「非常に」「やや」「ほとんど」といった程度量の表現の類似度を調べたものです。なぜそういうものを調べたかといいますと、科学者によるコミュニケーションの手段として非常に重要だからということが序論で述べられています。「非常に」や「ほとんど」といった程度量は、リッカート尺度を使ったような質問紙の表現としてよく使われていると思います。また、実験の教示でも「少し待ってください」とか「できるだけ間違えないように」など言ったりします。どういう表現だと参加者に余計な待ちをさせないですむかとか、求めていることが正しく伝わるかというのは割と直感に任せているところがあります。これに対して定量的な評価を行うことは意味があることではないでしょうか。また、臨床家による情報伝達では、どのぐらいある病気とか療法に危険度や効果があるのかを表現するときに程度量が関係してくるかもしれません。織田(1970)では、そうしたバックグラウンドのもとに程度量についての様々な調査を行った研究が報告されています。具体的な手続きとしては、ここにあるように二つずつ、一対比較の研究なので二つずつ提示して、下線部の表現についてどっちがより大きいことを表すか、大きいという意味を表すかを選択してもらいました。二択で「かなり」とか「非常に」の場合は大きい意味のほうを選んでもらい、逆に「少しも」とか「全然」といった表現の場合はそうでないほう、つまり、大きくないという意味のほうを選んでもらっています。

これと同じことを自由分類課題を使ってやってみようというのがここからの主旨です。最初にお配りしている紙を出してください。ここまで来てこの紙の意味がわかったと思うのですが、今からこの「非常に」とか書かれたほうの紙を切り分けてください。まず紙を縦に二つに折って切り離してください。折ったところでその後それぞれを五つずつ分けてください。はい、10枚のカードができたかと思います。裏に数字が書いてありますがこれはあとで分類するための数字なのであまり気にしないでください。

※以下の10種類の表現を刺激として用いた

- 非常に
- とても
- かなり
- わりに
- やや
- 少し
- どちらかといえば

- あまり
- ほとんど
- 少しも

このようにカードができたところでこれから分類課題を実際にやってみたいと思います。それぞれに「非常に」とか「やや」とか書いてあります。質問紙のリッカート尺度の「とてもあてはまる」とか「ややあてはまる」とかいった尺度の値の上に置く形容詞だと思ってみてください。そのうえで何らかの意味で似ているもの同士でこれらをグループ分けしてみてください。分け方には制約はありませんが、一つのグループには項目が二つ以上になるように分けてください。ネガティブな意味のもの、「ない」「でない」という意味が正しく続くと思われるものについてはそのような意味で解釈をして分類してください。山はいくつになっても構いません。できた方は回答フォームを使ってどんな回答をしたかを送ってください。送っていただいたら、その後集計してお返ししたいと思います。なので、ここで小休憩を入れたいと思います。

MDSの実行

データをアップロードしましたので資料の URL からアクセスしてください（※当日配布資料にはデータダウンロード用の QR コードと短縮 URL を印刷してあった）。zip フォルダでアップロードしてありますのでパソコンをお持ちの方はパソコンから短縮 URL で開いたら分類データというフォルダの中に CSV ファイルがいっぱい並んでいると思います。

CSV ファイルの中身はたとえば Figure 3 のような感じです。何のラベルもつけていませんが、各行と各列が「とても」「わりに」「非常に」といった、一つひとつの程度量表現を表しています。2つの表現が同じグループに分類された場合は「1」、そうでない場合は「0」になります。対角成分は同じ表現同士の組み合わせになるのでつねに「1」になります。

フォルダの中には、「convmatrix.R」というファイルもあります。ここで、R を起動してください。起動したらツールバーの「ファイル」をクリックして「R のソースコードを読み込み」を選んでください。ファイル選択ウィンドウが開くので、いま

ダウンロードした「convmatrix.R」のファイルを読み込んでください。

次に、今回使用する DistatisR というライブラリを起動します。これはデフォルトではインストールされていないライブラリなので、まだインストールしていない方は「パッケージ」から「パッケージのインストール」を選び、適当なミラーサイトから DistatisR をインストールしてください。インストール出来たら、コンソールに以下のコードを入力してください。

1. ライブラリのインストールと読み込み

```
install.packages(DistatisR)
library(DistatisR)
```

何もエラー等の出力がなければ読み込み成功です。次に、以下のコードをコピーしてください。

2. 距離行列の作成

```
DistanceCube <- convmatrix("C:hogehoge")# csv
ファイルを格納したフォルダパスを入力すること
```

注意していただきたいのは、実行する前に「C:hogehoge」のところをご自身の PC のいまダウンロードしたデータファイルがあるところのパスに書き換えてください。パスを区切る際にはバックslash (¥) ではなく slash (/) を使ってください。Mac では、デフォルトのダウンロードフォルダであれば、「~/Users/hogehoge/Downloads」のようになります。パスを適切なものに書き換えたら実行すると、先ほど読み込んだ「convmatrix.R」に書かれた関数を使ってデータを読み込みます。この関数は特に難しいことをしているわけではありません。いま作成した DistanceCube という変数の中身を見るとわかりますが、これは複数の CSV ファイルをひとつずつ開いては読み込み、最後に合体させてひとつの三次元配列にするものです。配列として読み込むのはちょっと面倒なのですが、配列型の変数でないで後で実行する分析に困るのでそうしています。ここでそういうコードの解説をしていると時間がかかってしまいますので、あらかじめ一連の作業をまとめたコードを配布し

1	1	0	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0	0	0
0	0	0	0	1	0	1	1	0	0
0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	1	1	0	0
0	0	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	1

Figure 3. 分析に用いるデータの例 (一名分)

ました。

実際に読み込んだ `DistanceCube` を見てみましょう。先ほど PC にダウンロードしていただいた CSV ファイルと比べてみてください。同じデータが読み込まれているはずですが、16 人分ありますので 10 以降は表示では省略されているかもしれません。

さて、この後のコードは実は分析のためには必須ではありません。必須ではないのですが、これをしないと出力を見たときに非常にわかりにくくなるのでここで各行と列に名前をつけましょう。一般的に使えるコードとしては以下ようになります。

```
dimnames(DistanceCube) <- list(c("hoge1",
  "hoge2", ...), c("hoge1", "hoge2", ...),
  c("taro", "hanako", ...))
```

`list` 型で、一つ目と二つ目の要素としては名前のベクトルを与えています。行と列で同じ要素が同じ順序で並びます。これは、ここで使用しているデータの構造がそのようになっているからです (CSV ファイルの中身を思い出してください)。三つ目の要素に「taro」、 「hanako」といった名前が見えます。これは参加者の別を表しています。

では、今回のデータに合わせて書き直した具体的なコードを示します。程度量表現のリストを何度も書くのは面倒なので、いったん `namelist` という変数に格納しています。そのあとで、参加者の別を「p1」から「p16」で表しています (あらかじめ参加人数がわからなかったため、`DistanceCube` の要素数から人数を読み取る書き方にしています)。

```
namelist <- c("とても", "非常に", "わりに", "かなり",
  "少し", "やや", "あまり", "どちらかといえば", "少しも", "ほとんど")
dimnames(DistanceCube) <- list(namelist,
  namelist, paste0("p",
  1:(length(DistanceCube)/(nrow(DistanceCube)
  * ncol(DistanceCube))))
```

このコードを実行した後で、再度 `DistanceCube` の中身を見てみるとどのように名前が付けられたのか確認できます。

では、ここから実際の分析をやっていきます。まず、今までもお話したようなふつうの MDS です。以下のコードをまとめて実行してください。

```
# 3. 全評価者の距離行列を足して全体データで分析する (第一の分析)
TotalDistance <- apply(DistanceCube, c(1, 2),
  sum)
```

```
# 3. 1 TotalDistance をメトリック MDS で分析
mdsRes <- mmds(TotalDistance)

# 3. 2 結果をプロットする
PlotMDS <- prettyPlot(mdsRes$FactorScore,
  display_names = TRUE,
  display_points = TRUE,
  contributionCircles = TRUE,
  contributions =
  mdsRes$Contributions)
```

ひとつめのコードは、先ほど作った $10 \times 10 \times 16$ の配列を二次元の 10×10 の配列にまとめる作業をしています。単純に 16 人分の二次元行列を足し合わせています。この全員分を足した行列を使って MDS を行います。単純に `mmds` という関数を適用するだけです。このコードでは `mdsRes` という変数の中に結果を格納しました。その次に、これをプロットするコードを 3.2 のところに書いてあります。

実行すると、以下のような図が出力されます (Figure 4)。

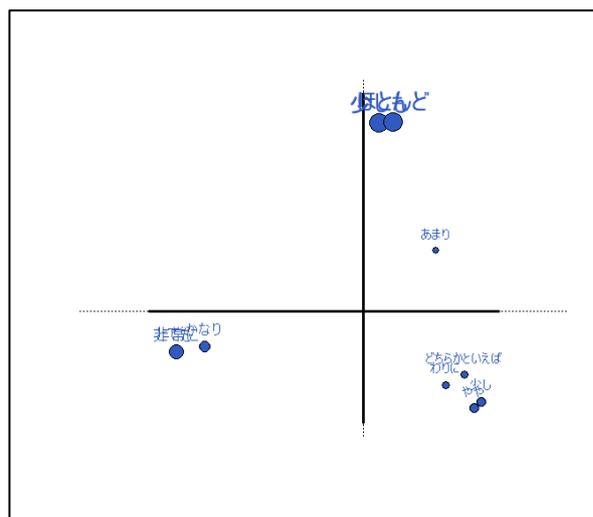


Figure 4. DistatisR による MDS の実行例
(二次元解のプロット)

ちょっとこの `DistatisR` パッケージの欠点というか、使いにくいところはラベルがかぶって見づらいということですね。日本語で表示する必要がないというのであれば、最初に名前をつける場所を省略すればラベルが数値になります。数値になるとここまで邪魔ではないので、あとでどれが何の項目だったのか照合していただくのがいいと思います。ただぱっと結果を見るためには、ラベルがないと何の結果なのか見づらいのであえて日本語のラベルがつくようにしています。

では、順番が前後しますがこの `mmds` について説明をしましょう。`mmds` は古典的なメトリック MDS を行う関数です。小杉先生からもお話があったので必要ないかもしれませんが、類似度行列でなくて距離行列を入れないといけないというのが注意しないといけないところですね。これについては、先ほど読み込みのための関数を使ったところで 1 と 0 を逆転させてあり、類似度行列を距離行列に置き換えています。あと描画の際に使った `prettyPlot` にはオプションとして `contributionCircles` というものがあり、これを `TRUE` にしておくことで寄与率の高い項目ほど大きく表示します。出力されたグラフを見たときに日本語のラベルの大きさが違って何だこれかと思ったかもしれませんが、それはこのオプションが原因です。オプション指定でここを `FALSE` にしておくことで全部同じ大きさで表示されるようになります。そうすると寄与率の情報は表示されなくなりますが、配置は見やすくなると思います。R には MDS を実行するための複数の関数があります。なので、メトリック MDS を実行するためにわざわざこの `mmds` を使う必要は特にないのですがせっかく同じ `DistatisR` パッケージに入っているのだから使ってみました。試してみたところ、`cmdscale` とまったく同じと思われる結果が得られます (`cmdscale` の結果に基づく Figure 5 を回転させてみてください)。違うのは `prettyPlot` というパッケージを使った図に対応しているところかなと思います。このパッケージが寄与率を表示する仕事をしています。

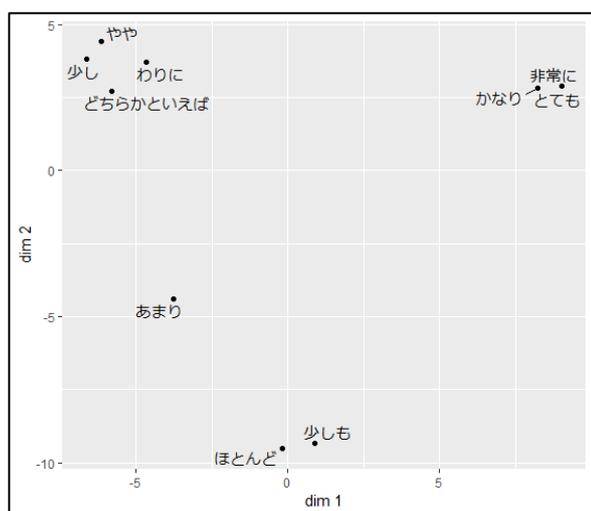


Figure 5. `cmdscale` に基づく結果

いまごく簡単に一対比較をできたわけですが、織田先生の行った研究の結果と比べてみましょう。織田 (1970) の Table 3 と Fig. 2 が今回のデータにおよそ対応する内容になっています。実際には、この研究はたくさんのいろんな程度表現を評価していて、今回はそのうちのごく一部、織田先生が現実の程度量表現用語と呼んでいるものの一部です。オリジナルでは 18 項目あって、それをさらに大学生、中学 2 年、小学 6 年、小学 4 年生

に評価してもらっているというかなり巨大なデータですね。Table 3 には 5 とか 4 とかの値が並んでいますが、どちらが勝ったか——つまりどちらのほうが大きさが強いと評価されたかが数値で表されています。これをもう少しまとめたものが Fig. 2 の程度量表現の尺度値です。ここでは、ギルフォードの『精神測定法』のケース V に基づいて尺度値を算出したとあります。ここから言えるのは、近い位置にある表現同士が比較的近い量を表すということです。上の高い数値のものが量が非常に高く、下の低いものは低く、また、点々が囲まれているものはお互いに比較的よく似ていると判断されているのがわかります。いまこの場にいるのは成人だけなので、大学生と比べてみるのがよいのではないかと思います。この大学生のデータで見ると「すごく」とか「非常に」とか「たいへん」とか、この辺りはよく似ていて、離れたところに「やや」「少し」「多少」というちょっと曖昧な感じの表現があって、下のほうに「あまり」「どちらともいえない」「ほとんど」みたいなかなり極小な感じの表現が並んでいます。実際に計算してみた MDS の結果というのはどうだったでしょうか。図の左下の領域には「非常に」とか「とても」とか「かなり」といった多い量を表す表現が集まっています。右上には「ほとんど」と「少しも」があり、これらは強い打ち消しの意味の表現ですね。そして、右下に「やや」「少し」「どちらかといえば」「わりに」が集まっています。これらは方向性に関係なく、量を少しだけ増やしたり減らしたりする、修飾するような表現ですね。最後に、「あまり」だけが孤立しています。これは「あまり～でない」と解釈していただきたい表現ですが、その通りに解釈していただけていたとすれば、他とはちょっと違う意味だったので孤立しているのかもしれませんが。「やや」他のグループは肯定でも否定でも意味が取れるのに対し、「あまり」は否定でしか意味が取れませんか、減らす量も「やや」のグループよりも多いか少ないと感じられたのかもかもしれません。表現同士の距離については、織田 (1970) も今回行った課題でもよく似ていますが、「あまり」の孤立などはやや違ったところですね。また、一次元ではなく、二次元に配置したことによって、大きく 3 つのグループができるということがよくわかる結果になったと思います。

DISTATIS の実行

さて、いま行った MDS の結果は個人差というものは無視しています。と言うよりも考慮できません。一方、われわれが実際に行った課題では、同じ項目に対して複数の参加者が評価をしました。そうすると類似性の評価において「かなり」と「とても」が似ているということについてはほとんどの人は一致しているけども、「やや」と「少しも」とは違うということについては、そうだと思う人とそうでもない人がいるかもしれません。この点を見ていこうというのがここからの分析です。

そこで、`DISTATIS` という方法を使ってみようと思います。

今回の発表の内容は、Abdi という人たちによる自由分類課題のデータの分析法を解説した論文に基づいています (Chollet et al., 2014)。DISTATISはこのAbdiのグループが開発した方法です。個人差を扱うということでは、小杉先生にご紹介いただいたINDSCALと目的は似ていますが、計算のプロセスやアプローチが若干違っていると思います。このDISTATISという方法では、同じ評価対象に対する複数の回答者による距離行列を分析するのですが、複数ある距離行列の間の類似度を評価して、最もよく元の行列を表現できるような合成行列を作成してから元の行列を合成行列に投射する——平均を取ると言ったら大雑把すぎるんですけども、発想としてはそういうような感じだと思います。数学的なところはくわしくは解説しませんが、まずはやってみましょう。

手順としては、`distatis` という関数に先ほど準備したデータを入力すれば簡単にできます。気をつけないといけないのは、データの型を三次元行列にしないと絶対に分析できないというところです。`list` や `matrix` ではなく `array` でないといけないということですね。Chollet et al. (2014) の付録に分析のためのコードが書いてあるのですが、パッケージがバージョンアップしたために付録のままのコードだと動かないところがあります。その部分を修正したコードを以下に示します。また、オリジナルの説明とやり方が違うところとして、4種類の結果を個別に出力するのではなく、ひとつの関数でまとめて出力するようにしています。実行していただくと一度にたくさんの図が出てくるのでびっくりしないでください。では、早速やってみましょう。

```
# 4. DISTATIS を使って分析する (第二の分析)
testDistatis <- distatis (DistanceCube)

# 4. 1 ブートストラップによる推論
BootF <-
  BootFactorScores (testDistatis$res4Splus$PartialF,
                    niter = 1000)
F_fullBoot <- BootFromCompromise (DistanceCube,
                                   niter = 1000)

# 4. 2 グラフを作る
LeF <- testDistatis$res4Splus$F
PartialFS <- testDistatis$res4Splus$PartialF

# 4種類のプロットを一度に描く
# 1. 因子得点からのブートストラップ信頼区間つきの観測値
# 2. 部分因子得点からのブートストラップ信頼区間つきの観測値
# 3. 観測値とブートストラップ信頼区間を楕円で表現
```

```
# 4. Rv 行列からの因子得点
```

```
PlotOfAll <- GraphDistatisAll (LeF, PartialFS,
                               BootF, testDistatis$res4Cmat$G)
```

4の次の行のコードが実際の分析の実行で、その後は出力を得るためのステップがいくつかあります。これも本当は細かく分けなくてもできると思いますが、一行ですませようとするとかえって複雑になりそうなので分けています。4.1のところにあるのはそのような中間ステップで特にここで何かを出力することはありません。最後の `PlotOfAll` という変数に代入しているところで一気に出力が出ます。一瞬画面が光ったように見えるかもしれませんが、四つの図が一度に出力されるためです。では、これらの図がそれぞれなんなのかということと、その読み方を説明していきたいと思います。

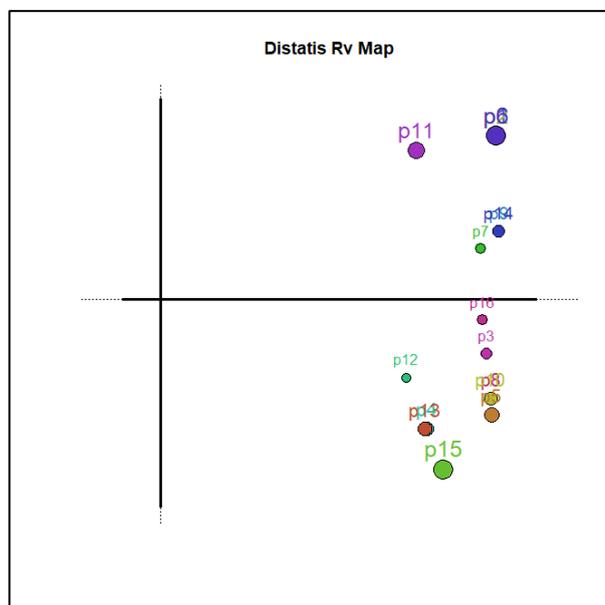


Figure 6. Distatis Rv Map

まず、Distatis Rv Map という図について見てみましょう (Figure 6)。これは因子得点をもとに評価者ごとのばらつきをマップしたものです。それぞれの円の脇に、たとえば、p1のように添え字があります。これはコードの中で評価者の違いを表すために p1 番から順番に番号を振った結果です。円のひとつひとつが評価者を表しています。分類の仕方が似ている人ほど近い位置にプロットされることになります。たとえば、p15 番と p11 番の人たちがそれぞれ極の位置にいて、ちょっと違う判断をしていることがわかります。また一人だけ孤立しているとか、島ができていない感じでもないの、全体としてはあまり大きな違いはないといえるでしょう。もっと離れていてひとりだけぼつんといたり、大きく 2 つに人々が分かれているようなことがあれば、他とは違う評価基準を持った個人やグループがいると

ということになるでしょう。明らかにこの人は外れ値だといったデータが見つかったら除外して再分析することも考えられますね。

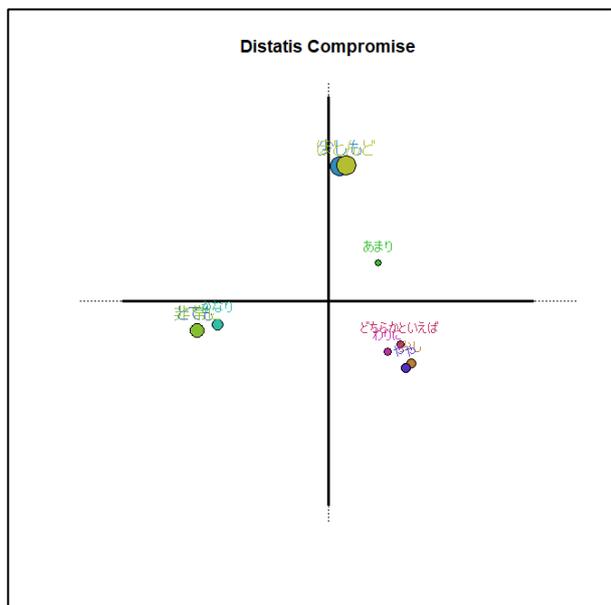


Figure 7. Distatis Compromise

次に、Distatis Compromise を見ましょう (Figure 7)。複数の距離行列を反映するように合成して作った行列を投射したもので、MDS に対応する結果です。先ほどの MDS と大体同じような布置になっていると思います。「とても」と「かなり」と「非常に」が重なっていて、右下にやんわりした評価が集まっています、残り 2 つのまとまりと「あまり」があるので、まあだいたい同じ配置といっているのではないのでしょうか。MDS のプロットと同じように解釈できる図で、DISTATIS の主な分析結果と見ていいかもしれません。これがつねに MDS と同じ結果になるのかということですが、個人差を反映した結果、MDS とはずれるということがときどき起こります。

今度は Distatis Bootstrap を見ましょう (Figure 8)。これは Compromise と基本的に同じものですが、それぞれの項目の因子得点に bootstrap 信頼区間が加わっています。著者らはこれ 95% 信頼楕円 (confidence ellipsoid) と呼んでおり、実際に楕円の形をしています。中心から円で囲む形で信頼区間を描いているので、角度によって距離が異なるために楕円状になるということです。これを見ると項目の布置にどのくらいばらつきがあるのかを知る手がかりになります。どういう使い方ができるかというと、二つの項目の信頼楕円に重なっていないときはその二つの項目は評価者によって有意に異なるものと見なされているといった評価ができます。逆に、分かれているように思えても、信頼楕円が重なっている場合は、本当に分かれているのかどうかかわからないとか、ばらつきが大きいために違いがあるかどうか

かはっきりしないといった見方ができます。今回得られたデータではかなりきれいに分かれていますね。これまで見てきた通り、だいたい 3 グループに分かれていて「あまり」は「少し」「ほとんど」群と「どちらかといえば」他の群の両方に引っ張られているのが見て取れると思います。

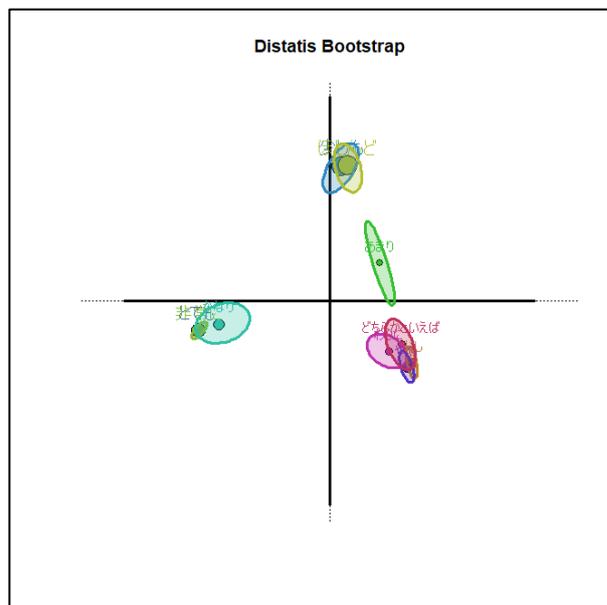


Figure 8. Distatis Bootstrap

最後に今一番見づらい図ですが、Distatis Partial Colored By Observations というものがあります (Figure 9)。この図はどうなっているかと言いますと、これまで見てきたような項目の因子得点のマップに、各評価者の行列から見た部分因子得点を加えたものです。

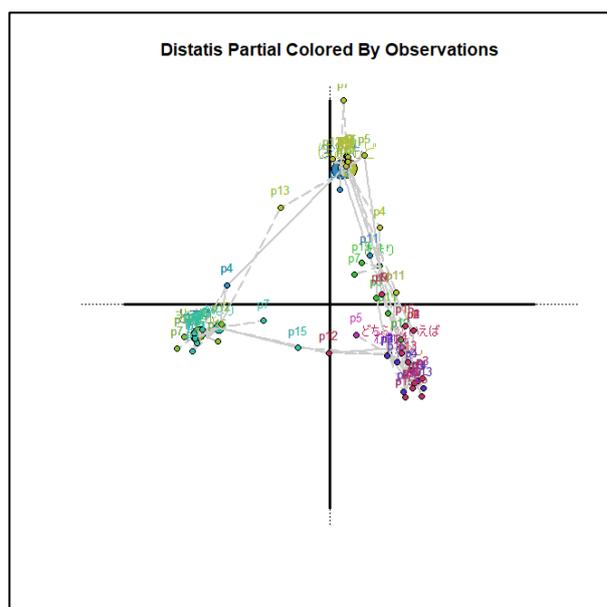


Figure 9. Distatis Partial Colored By Observations

各項目を表す円を中心に、各評価者を表す小さな円が衛星のように配置されており、項目から評価者へ線分が引かれています。ある項目を中心として見たときに、評価基準が離れている評価者ほど、互いに離れて配置されています。たとえば、「かなり」についての評価はp7番の人は他の人から大きく離れていることがわかります。評価する対象が10個あり、かつ評価した人が16人いるので、今回の出力はたいへんわかりづらいものになっています。完全に個人ごとにばらばらにするのではなく、何らかの属性に基づいて評価者をグループ化し、3~5程度のグループにすればもう少しわかりやすくなると思います。図そのものをもうちょっと見やすくする工夫があればなあというのは私も気になっているところですが、一通りこんな感じで見てきました。

方法論的考察

自由分類課題を使ってデータを集めて類似度行列を MDS なり DISTATIS で分析できること、あるいは、小杉先生にご紹介いただいたような他の MDS のバリエーションで分析ができるということが何となく伝わっていればと思います。実はここまで来てようやく本題という気が私はしています。いまこの課題を使って分析ができるということを経験していただいたわけですが、では、こういう方法を使って研究していくことは本当に心理学にとって役に立つのだろうか、おぼえて使っていけるのだろうかということをお自身も考えているところです。

まず自由分類課題自体が本当に有用なものなのかということがあると思います。最初に問題になるのは、一対比較もそうですが、これらの手続きにおいてどんなデータが得られているのかということ。明らかにこの自由分類課題のほうが参加者の負担が軽いのです。そうすると一個の項目にかかる時間とか費やす労力も小さくなるわけです。たとえば、この「とても」と「わりに」を比べてどのくらい以ているかを評価するという観点からすると、一対比較で出されたときほどには個々の項目を丁寧に見ていないわけです。そうするとこれは本当に妥当なのかという問題があるわけです。これをどうしたら克服できるのかということですが、もしかしたら先ほど見た信頼楕円のようなものを使えば、ばらつきの大きさからすると問題にするほど大きくはないといった評価はできるのかもしれませんが。

そして、ここまで考えて気がつくのですが、一対比較と同じならいいのかということが次に疑問になってくるのではないのでしょうか。一対比較の場合だと必ず二つずつの項目を見比べて評価します。それに対して、自由分類課題では三つ四つがいつぱい入ってくることになります。そうすると気が散るという見方もできますが、逆に同時の方が意味の評価のためにはよいという可能性もあります。いま評価しているペアだけでなく、第三、第四の項目も考慮した上で、A との関係でみれば B はまだ C に近い方だとか、いやここはこのぐらいというようにその

第三の項目との距離みたいなものも実は考えたほうがいい、そのほうが自然だという見方もできるかもしれません。そうすると一対比較に近いからいいという主張が成り立つのであれば、逆に一対比較のほうが自由分類にもっと近くないといけないという言い方もできるかもしれません。そこまで行くと、一対比較か自由分類かという話ではなくて、本当は外的基準がないとダメなんだという話になってくるのではないのでしょうか。

次に分析法の方ですね。MDS や DISTATIS による分析が本当に有用かという問題ですね。まず解釈の妥当性の問題というのはどうしても出てくると思います。特に、軸とかお互いの項目の関係について解釈するとき、理論的な裏付けなしに出力だけを見てしまうと、何でもお話が作れてしまう危険性は皆さんも感じられたのではないのでしょうか。こういった分析をしていて綺麗にマッピングができると非常にありがたいのです。解釈できないようなものが出てくることもあるのですが、できる場合は非常に解釈を助けます。しかし、それなら綺麗に描けない場合とか、何だかわからない、解釈できない場合はどうするんだということが気になるわけです。うまくいったものはすんなり報告して、うまくいかなかったものは何となく放棄されるとしたら、これはいわゆる結果の選択的な報告に当たるのではないのでしょうか。あるいは別の考え方として、選択的に報告するのはよくないから何だろうと全部報告するということになって、何だかわからないものを出されてもそれはそれで困るわけですね。解釈に役に立たないのなら出すべきでないのではないかも思っていて、これについては私もどうするのが正解なのかかわからないでいます。こういった問題はおそらく探索的研究につきものなのだろうなと思います。つまり、分類課題をしたり、MDS を使ってマッピングして探索的に調べようというときには探索的研究としてどうあるのがよいのでしょうか。

一方で、こういった手続きの利点として挙げられそうなのは、分類するとき言語的なバイアスを避けられるということかなと思います。だいたい心理学で探索的研究と言うと、何らかの質的分析とか因子分析をすることが多いのではないのでしょうか。質的分析では、データを取らせてもらった人の発言とか、研究者のアイデアからカテゴリーを作ってこれに当てはまるような行動や発言は何だろうと分類したりすることがよく行われます。そして因子分析も数量的に評価はしますが、そもそもどういった尺度項目を使うかを定める段階とか、あるいは、項目を分類して因子を見つける段階でも言語に頼っている部分は少ないわけですね。そうすると言語ラベルに引きつけて反応とかコードを分類してしまう可能性が残ってしまいます。それに対して、一対比較や自由分類では、明示的なラベルなしに分類が可能であるとは言えると思います。ただ、もちろん、解釈の時点ではどうしても言語的に表現をすることになってしまうのですが。最終的な解釈の妥当性を担保するためには、理論的にはこうだとか、参加者はこう言っていたというような外的基準が必要に

なると思うのですが、これらはアドホック的になりがちです。そこで、今度は逆に外的基準だけではなくて、内的な整合性の高さから今回の布置がうまくはまっているねみたいなことが言えると役に立つのではないかと考えています。今回はぼつちりデータを表現できていてこれはいいみたいなことが言えると、研究内での一貫性が主張できることになり、これは研究者の主観や言語によるバイアスのある程度免れたものになるのではないのでしょうか。

今回の研究会をきっかけに皆さんにも MDS や DISTATIS を使った研究を実践していただき、またその成果をお知らせいただければうれしいです。そうすると、この方法にこんな可能性あったのかとわかって私も非常に勉強になります。では、そろそろ時間となりましたのでおしまいとさせていただきます。今日はどうもありがとうございました。

文献

- Abdi, H., Valentin, D., Chollet, S., & Chrea, C. (2007). Analyzing assessors and products in sorting tasks: DISTATIS, theory and applications. *Food Quality and Preference*, **18**, 627-640.
- Abdi, H., Valentin, D., O'Toole, A. J., & Edelman, B. (2005). DISTATIS: The analysis of multiple distance matrices. *Proceedings of the IEEE Computer Society: International Conference on Computer Vision and Pattern Recognition (San Diego, CA, USA)*. pp. 42-47.
- Chollet, S., Valentin, D., & Abdi, H. (2014). Free sorting task. In P. V. Tomasco & G. Ares (Eds.), *Novel techniques in sensory characterization and consumer profiling* (pp. 207-227). Boca Raton: Taylor and Francis.
- 織田揮準 (1970). 日本語の程度量表現用語に関する研究 教育心理学研究, **18**, 38-48.
- Valentin, D., Chollet, S., Nestrud, M., & Abdi, H. (2018). Projective mapping & sorting tasks. In S. E. Kemp, J. Hort & T. Hollowood (Eds.), *Descriptive analysis in sensory evaluation* (pp. 535-559). Hoboken, NJ: John Wiley & Sons.

付記

本稿の内容は、日本認知心理学会・研究法研究部会 第1回研究会の発表内容に基づくものです。

付録：データ読み込み用の関数

```
# 指定したディレクトリ内のすべての csv ファイルを結合して
# 多次元配列として読み込む関数
# 同じディレクトリ内に関係のない（書式の異なる） csv ファ
```

```
イルがあると干渉するので注意
# datapath としては""でくくったフォルダパスを指定すること
convmatrix <- function(datpath) {
  filenames <- list.files(datpath) # ディレクトリ
  # 内のファイルをリストアップ
  datfiles <- filenames[grep("^.*¥¥.csv$",
  filenames)] # 拡張子.csvのファイル名のみ取り出す
  resultdat <- replicate(length(datfiles),
  list()) # 結果格納用の空のリスト

  # データを読み込み、距離行列に変換
  for(i in 1:length(datfiles)) {
    resultdat[[i]] <- 1-
    read.csv(paste(datpath, datfiles[i], sep =
    "/"), header = FALSE)
  }
  isize <- nrow(resultdat[[1]])
  resultmat <- array(unlist(resultdat), dim =
  c(isize, isize, length(datfiles)))

  # 出力
  return(resultmat) # コンソールに出力
}
```