

第7章

対応関係におけるグループ化とその連結

野田容助

はじめに

体系の異なる分類どうしを結び付けるには両者の対応関係を明らかにした対応関係コード表が必要であり、対応関係コード表を使用する場合には2つの分類がどのような対応関係にあるかを検討することが重要な問題になる。対応関係コード表の中で分類の核になる閉じた対応関係にある分類コードの集まりを「グループ」ということにする。グループは佐藤の*Fundamental Concept of Social/Regional Summary Data and Inference in Their Database* およびその要約である「要約データの基礎概念とデータベース内での推論—世界貿易統計データベースを例として—」によれば、2つの分類から「共通に導出可能な最も詳細な分類 (FCD:Finest Common Derivative)」に対応する分類である。すなわち、対応関係に少なくとも1つの共通した結合があればつないでいき、結合が無くなったところまでの構成要素でグループを決めるという方法で得られた対応関係の集まりである。

形式的にグループを作るとグループに属する分類コードの共通性が問題になるが、この点については野田・山本の「体系の異なる分類の対応関係と変換—グループ化および切断による商品分類の変換の試み—」による「切断」という方法でサブグループを作り、共通した特性を持たせるような方法がある。また、2つの分類体系のグループ化を拡張した複数の分類体系における対応関係の連結としては野田の「商品分類の改訂に伴う対応関係の連結」があり、この連結プロセスを貿易統計における商品分類体系の改訂に応用してより具体的な形で例示している。

本章は佐藤のFCD (Finest Common Derivatives) を基本としたグループ化の考え方、野田の対応関係の連結の方法を拡張して、複数存在する対応関係に対

しても対応関係を連結する考えとその方法を示すことを目的とする。また、PLIプログラムで作成されたグループ化のためのプログラムのclcvp6_p.pli、切断により得られるサブグループ化のためのプログラムclcvp7_p.pli、対応関係を連結するプログラムclcvp9.pliの解説とその使い方について紹介する。clcvp9.pliは複数ある対応関係を連結するより一般化されたプログラムであり、clcvp6.pliと組み合わせることにより効果的な連結が可能となる。

1. グループ化および切断によるサブグループ化

一般に分類はカテゴリーと呼ばれる抽象的な個別主体を要素とする集合で表すことができ、2つの分類間のカテゴリーの対応関係はこれらの分類間のカテゴリーを結びつける関数によって表された分類規則で表すことができる。このような分類規則によって定まる分類間の対応関係についての考え方は基本的には前述した佐藤によるデータの記述対象の分類方法に関するデータの導出およびデータ間の比較にもとづいている。

本章では分類間の対応関係のグループ化にあたっては佐藤のFCDの考え方をもとにした野田・山本のグループ化の方法を採用する。

1.1 対応関係のグループ化

佐藤によればデータベース内に分類階層が用意されており、その中に対応する分類AとBが定義されているとすると、この両分類から共通に導出可能な最も詳細な分類(FCD)を機械的に求めるメカニズムを考えることができる^(注1)。分類X,A,B,Gが同一分類階層内に存在するとして類別関数 f_a, f_b, p を、 $f_a : X \rightarrow A, f_b : X \rightarrow B, p : X \rightarrow G$ とする^(注2)。分類Xを基準とした分類AとBの対応関係は、 $x_i \in X, i=1 \dots N$ に対して、 $\{f_a(x_i), f_b(x_i)\} i=1 \dots N$ として求められたものの中から重複しているものを取り除いて得られる。また、分類Xを基準としたAとBの対応関係としてk回目の閉じた関係を $R^k(x)$ とする。佐藤は分類規準をAとしているが、本章では対応関係の連結を目的にしているので佐藤のメカニズムを変形して規準をXとおく。

$$(1) \quad R^k(x) = (f_b^{-1} f_b f_a^{-1} f_a)^k(x) \quad x \in X, k=1 \dots$$

この関係が $R^{k+1}(x) = R^k(x)$ となるとき、 $R^k(x)$ を A と B の (収斂した) 閉じた関係といい $R^*(x)$ で表し、 $R^*(x) = (f_b^{-1} f_b f_a^{-1} f_a)^k(x)$ as $k \rightarrow \infty$ とする。なお、 x のカテゴリーの数を n とするとき、 $R^*(x)$ はたかだか n で収斂する。さらに、分類 X を規準としているので、 X を R^* により m 個に分割した分類を X / R^* と表す。すなわち、 $R^*(x_i) \in X$, $i = 1 \dots m$ とするとき、 $R^*(x_i) \cap R^*(x_j) = \phi$, ($i \neq j$) に対して、

$$X / R^* = R^*(x_1) \cup \dots \cup R^*(x_m)$$

である。このとき、分類 G に対して、

$$(2) \quad R^*(x_i) \xrightarrow{p} g_i \in G, \quad i = 1 \dots m$$

とすることができれば、言い換えれば、 $X / R^* = \{p^{-1}(g) \mid g \in G\}$ であるなら、分類の規準を X としたとき、 G は A と B の FCD であり、

$$(3) \quad FCD(A, B : X) \xrightarrow{p} G$$

と表わすことにする (注3)。簡単のために p を省略して $FCD(A, B : X) \rightarrow G$ とする。この関係を利用することにより互いに体型の異なった分類をもつ2つの要約データ A と B を比較可能な同じ分類をもつ G に変換して結合することができる。本章ではこのようにして得られた分類 G をグループと呼ぶことにする。

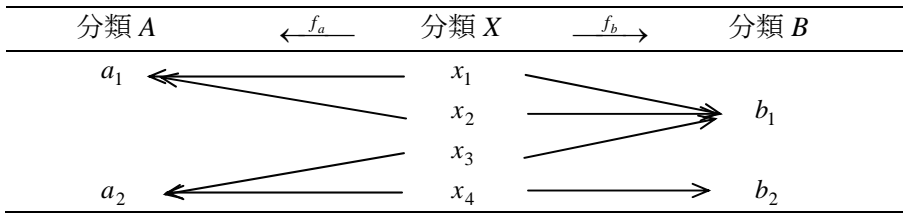
例として、次のような2つの分類 A と B があり、その対応関係は表1のようであるとす。これに、分類 X の最初の要素である x_1 から合成された類別関数の

(1) 式を使用して繰り返し計算をする。繰り返しが1回目の結果は次の通りである (注4)。

$$\begin{aligned} x_1 &\xrightarrow{f_a} a_1 \xrightarrow{f_a^{-1}} (x_1 \ x_2)' \xrightarrow{f_b} (b_1 \ b_1)' \\ &\Rightarrow b_1 \xrightarrow{f_b^{-1}} (x_1 \ x_2 \ x_3)' \end{aligned}$$

ここで、 \Rightarrow は重複する要素があるとき、その中の1つの要素のみを残して他を取り除き、要素間にダブリがないように並び変える演算を表す。これを式で表わして、 $(x_1 \ x_2 \ x_3)' = f_b^{-1} f_b f_a^{-1} f_a(x_1) = R(x_1)$ となる。1回目の関係については分類 X の x_1 から始まってもう一度分類 X に戻ったときには異なった結果 $(x_1 \ x_2 \ x_3)'$ を得たことになる。 $R^0(x_1) = x_1$ とすれば、 $R^1(x_1) \neq R^0(x_1)$ であり、この関係は閉じた状態にはなっていない。 $(x_1 \ x_2 \ x_3)'$ に対して (1) 式を利用

表1 分類Xを基準とした分類Aと分類Bの対応関係



(出所) 筆者作成

して2度目の繰り返しを実行する。

$$\begin{aligned}
 (x_1 \ x_2 \ x_3)' &\xrightarrow{f_a} (a_1 \ a_1 \ a_2) \Rightarrow (a_1 \ a_2)' \xrightarrow{f_a^{-1}} ((x_1 \ x_2)' \ (x_3 \ x_4)') \\
 &\xrightarrow{f_b} ((b_1 \ b_1)' \ (b_1 \ b_2)') \Rightarrow (b_1 \ b_2)' \xrightarrow{f_b^{-1}} ((x_1 \ x_2 \ x_3)' \ x_4)' \\
 &\Rightarrow (x_1 \ x_2 \ x_3 \ x_4)'
 \end{aligned}$$

これは、 $(x_1 \ x_2 \ x_3 \ x_4)' = R(x_1 \ x_2 \ x_3) = R^2(x_1)$ となるので、 $R^2(x_1) \neq R^1(x_1)$ であり、この関係は閉じた状態にはなっていない。 $(x_1 \ x_2 \ x_3 \ x_4)'$ に対して (1) 式を利用して3度目の繰り返しを実行する。

$$\begin{aligned}
 (x_1 \ x_2 \ x_3 \ x_4)' &\xrightarrow{f_a} (a_1 \ a_1 \ a_2 \ a_2) \Rightarrow (a_1 \ a_2)' \xrightarrow{f_a^{-1}} ((x_1 \ x_2)' \ (x_3 \ x_4)') \\
 &\xrightarrow{f_b} ((b_1 \ b_1)' \ (b_1 \ b_2)') \Rightarrow (b_1 \ b_2)' \xrightarrow{f_b^{-1}} ((x_1 \ x_2 \ x_3)' \ x_4)' \\
 &\Rightarrow (x_1 \ x_2 \ x_3 \ x_4)'
 \end{aligned}$$

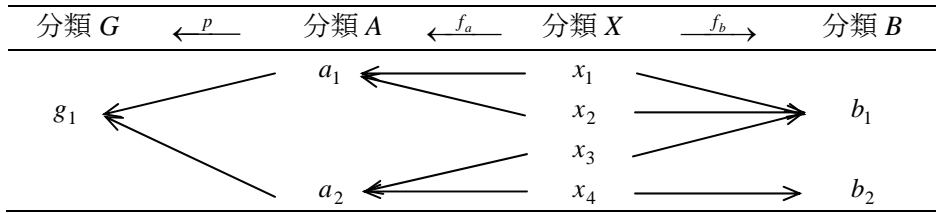
これは、 $(x_1 \ x_2 \ x_3 \ x_4)' = R(x_1 \ x_2 \ x_3 \ x_4)' = R^3(x_1)$ であり、 $R^3(x_1) = R^2(x_1)$ となる。すなわち、3回目は分類Xの $(x_1 \ x_2 \ x_3 \ x_4)'$ から始まってもう一度分類Xに戻ったときには同一の結果を得る。これ以上繰り返しても同じ結果しか得られないので、この状態で収束したことになり繰り返しの計算を中止する。ここで得られた $(x_1 \ x_2 \ x_3 \ x_4)'$ は閉じた関係にあるという。

次に、 $(x_1 \ x_2 \ x_3 \ x_4)'$ を新たな分類Gへ変換する。すなわち、分類Xから分類Gへの類別関数 p を (3) 式に従って、

$$(x_1 \ x_2 \ x_3 \ x_4)' \xrightarrow{p} g_1, \quad g_1 \in G$$

で定義する。これによってFCDの条件を満足するような分類Gと、類別関数 p が得られる。この分類GがグループGである。これらの結果をまとめると表2

表2 分類Aと分類Bとこの両分類から得られたFCDである分類Gの関係



(出所) 著者作成

が得られる。この分類GがグループGである。これらの結果をまとめると表2となる (注5)。

1.2 「切断」によるサブグループのモデル化

1つのグループからいくつかの対応関係を取り除くとグループがさらに2つ以上のグループに分かれるとき、この対応関係によってグループが「切断」されたといい、そのときに取り除いた対応関係を「切断の要素」ということにする (注6)。また、このとき得られたグループをもとのグループに対するサブグループという (注7)。 $X/R^* = R^*(x_1) \cup \dots \cup R^*(x_m)$ に対応してグループを、

$$G = G_1 \cup \dots \cup G_m, \quad G_i \cap G_j = \phi, \quad (i \neq j)$$

とする。グループの*i*番目である G_i の切断の要素の集まりを C_i で表す。このグループから切断の要素を取り除いた $G_i - C_i$ に対してグループ化をおこないサブグループを作る。 n_i 個のサブグループを $SG_i(j) \quad j=1 \dots n_i$ とするとグループ G_i は、

$$G_i = \{SG_i(1) \cup \dots \cup SG_i(n_i) \cup C_i\}, \quad SG_i \cap SG_j = \phi, \quad (i \neq j)$$

と分割される。

サブグループがグループから切断の要素を取り除いた対応関係コード表に対して再度グループ化をすることで得られるということは、切断の仕方によってサブグループの内容や個数が決まるということである。このことは切断というのは対応関係コード表のグループ化に対する1つのモデルであると考えることができる。切断をしない対応関係のモデルを対応関係の基本モデル、切断に

よりサブグループ化された対応関係を対応関係の切断モデルという。

1.3 対応関係の連結モデル

対応関係の連結については野田の「商品分類の改定にともなう対応関係の連結」に説明がある。この要点は分類の規準となる分類 X が存在しており、3種類の分類 A_1, A_2, A_3 があり、分類 X を規準として A_1 と A_2 、同じく分類 X を規準として A_2 と A_3 がそれぞれ対応関係にあるとき、この対応関係に共通して存在する分類 A_2 を規準として A_1, A_2, A_3 をグループ化という意味で連結できることである。

まず最初、分類 X を規準として分類 A_1 と A_2 の対応関係から得られたFCDを G_1 として、(4)式からつぎのように表すことができる。

$$(5) \quad FCD(A_1, A_2 : X) \rightarrow G_1$$

2番目の対応関係は分類 X を規準として分類 A_2 と A_3 から得られたFCDであり、これを G_2 とすれば、

$$(6) \quad FCD(A_2, A_3 : X) \rightarrow G_2$$

となる。分類 A_1, A_2, A_3 の対応関係は A_1 と A_2 の対応関係、 A_2 と A_3 の対応関係の両関係に存在する A_2 を規準とした連結から得られる。すなわち G_1 と G_2 を A_2 により関連させて G_1 と G_2 の対応関係を作成し、この対応関係のFCDを求めることで CG_2 が得られる。

分類 A_2 を規準とした G_1 と G_2 の対応関係はつぎのようにして求めることができる。(5)式において(2)式に相当する類別関数を $R_{G_1}^*(x)$ 、(3)式の p を p_{G_1} とすれば、(5)式は、

$$(7) \quad R_{G_1}^*(x) = \{x \in X \mid p_{G_1}(x) = g_1 \in G_1\}$$

を意味する。 A_2 を(6)式で分割すれば、

$$(8) \quad \begin{aligned} A_2 / R_{G_1}^*(x) &= \{v \in A_2 \mid f_b(x) = v, x \in R_{G_1}\} \\ &= \{v \in A_2 \mid p_{G_1} f_b^{-1}(v) = g_1 \in G_1\} \end{aligned}$$

となる。同じようにすれば(6)式から、

$$(9) \quad A_2 / R_{G_2}^*(x) = \{v \in A_2 \mid p_{G_2} f_b^{-1}(v) = g_2 \in G_2\}$$

が得られる。したがって、分類 A_2 を規準とした G_1 と G_2 の対応関係は (8) 式と (9) 式から、 $v \in A_2$ に対して、

$$(10) \quad \{p_{G_1} f_b^{-1}(v), p_{G_2} f_a^{-1}(v)\}$$

として得られたものの中から重複を取り除いたものとして表わすことができる。 $CG_1 = G_1$ とすれば、この関係は、

$$FCD(G_1, G_2 : A_2) = FCD(CG_1, G_2 : A_2) \rightarrow CG_2$$

となる。この過程において (10) 式で示されているように G_1 と G_2 を A_2 により関連させて G_1 と G_2 の対応関係を作成することが重要である。

同じようにすれば4種類の分類 A_1, A_2, A_3, A_4 の対応関係の連結は A_1, A_2, A_3 の対応関係、 A_3 と A_4 の対応関係のそれぞれを共通に存在する A_3 を規準として連結することで得られる。すでに得られている CG_2 に対して、

$$FCD(A_3, A_4 : X) \rightarrow G_3$$

を得る。 CG_2 と G_3 を A_3 により関連させて CG_2 と G_3 の対応関係を作成する。この対応関係のFCDは、

$$FCD(CG_2, G_3 : A_3) \rightarrow CG_3$$

となり、 CG_3 が得られる。

1.4 対応関係の一般化された連結モデル

この関係を拡張してグループの連結における一般形を漸化式として導くことができる。分類 A_1, A_2, \dots, A_{n+1} が $n+1$ 種類あり、同時に分類 X を規準として (A_k, A_{k+1}) $k=1 \dots n$ となる対応関係があるとする。これらの対応関係の連結を CG_n とするとき、この関係を漸化式としてまとめたのがつぎの処理過程である。

[1] 初期値を設定するために分類 X を規準として分類 A_1 と A_2 に対して $k=1$ とおいて、

$$FCD(A_1, A_2 : X) \rightarrow G_1$$

として G_1 を求め、 $CG_1 = G_1$ とする。

[2] $k=2 \dots n$ に対して、 A_1, A_2, \dots, A_k を連結した CG_{k-1} が得られているとき、分類 X を規準として A_k と A_{k+1} のFCDを求め、

$$(11) \quad FCD(A_k, A_{k+1} : X) \rightarrow G_k$$

として G_k を求める。(3) 式に対応してグループ化に必要となる CG_{k-1} 、 G_k と規準となる A_k が揃ったことになる。

[3] 連結の軸となる分類 A_k にもとづいて CG_{k-1} と G_k の対応関係を作成する。

[4] 分類 A_k を規準として CG_{k-1} と G_k の対応関係により FCD を求め、

$$(12) \quad FCD(CG_{k-1}, G_k : A_k) \rightarrow CG_k$$

として CG_k を求める。

[5] $k=n$ となるまで [2] から [4] までの処理過程を繰り返す。

[6] $k=n$ となったときに得られた CG_n が求める A_1, A_2, \dots, A_{n+1} の連結された分類である。

この処理過程においてステップ [3] における分類 A_k にもとづいた CG_{k-1} と G_k の対応関係の作成は重要である。 G_k については、(11) 式から $x \in X$ に対して、

$$R_{G_k}^*(x) \xrightarrow{p_{G_k}} g \in G_k$$

であり、これを書き直せば、 $R_{G_k}^*(x) = \{x \in X \mid p_{G_k}(x) = g \in G_k\}$ である。この式で A_k を分割すれば、

$$(13) \quad A_k / R_{G_k}^*(x) = \{v \in A_k \mid p_{G_k} f_a^{-1}(v) = g \in G_k\}$$

となる。 CG_{k-1} については、(12) 式から $v \in A_k$ に対して、

$$R_{CG_{k-1}}^*(v) \xrightarrow{p_{CG_{k-1}}} cg \in CG_{k-1}$$

であり、この式は A_k 自身であるので改めて A_k の分割として、

$$(14) \quad A_k / R_{CG_{k-1}}^*(x) = \{v \in A_k \mid p_{CG_{k-1}}(v) = cg \in CG_{k-1}\}$$

と表わすことにする (13) 式と (14) 式から分類 A_k を規準とした CG_{k-1} と G_k の対応関係は $v \in A_k$ に対して、

$$(15) \quad \{p_{CG_{k-1}}(v), p_{G_k} f_a^{-1}(v)\}$$

としてとして得られたものの中から重複を取り除いたものとして表わすことができる。

2. 対応関係におけるグループ化の方法

処理を簡単にするためにグループ化したい対応関係コード表に分類Aと分類Bごとに分類コードの一連番号を付ける。次に分類A、分類Bの順にこの一連番号を昇順に並べる。このようにして作られた対応関係コード表の例を表1に示す。対応関係のグループ化は次のようなプロセスを通しておこなう。

[1] 分類Aで同一番号を持っている分類コードは同一グループのメンバーである。同一グループに属している分類Bの番号のすべてをグループを識別する共通の識別記号で置き換える。この共通識別記号はグループ内に存在する最初の分類Bの一連番号で代替する。

[2] 分類Aに属するすべての分類コードについて [1] の処理を繰り返す。

[3] グループごとに決められた識別記号を基準に並べ替え、識別記号に一連番号を付ける。この識別番号、または、一連番号でまとめられたのがグループである。

プロセス [1] は分類Aの a_1 から処理を始めるため、(1) 式に相当する R を、

$$(16) \quad R'(a) = f_a^{-1} f_b f_b^{-1} f_a(a) \quad a \in A$$

とするとき、 a_1 に対応するBの要素の個数が m_1 であれば、

$$(17) \quad (b_1 \cdots b_{m_1})' = f_b f_a^{-1}(a_1)$$

と表され、この式において (b_1, \dots, b_{m_1}) を (b_1, \dots, b_1) に置換えることである。

プロセス [2] は (17) 式をAに属するすべての要素 $a_i, i=1 \cdots N$ に対して実施するこであり、 $a_i \in R^*(a_1)$ を満たすAのすべて要素を (a_1, \dots, a_n) とすれば、

$$(b_1 \cdots b_1)' = f_b f_a^{-1}(a_1 \cdots a_m)' = f_b f_a^{-1} R^*(a_1)$$

となっていることと等しい。AとBの対応関係がL個のグループに分かれるとき、分類Aは、 $R^*(a_k), k=1, 2, \dots, k_L$ によりL個に分割される。 $k_2 \cdots k_L$ はNを超えない自然数である。これが、 A/R^* である。Bについても同じことであり、 $f_b f_a^{-1} R^*(a_k), k=1, 2, \dots, k_L$ によりL個に分割され、 $B/(f_b f_a^{-1} R^*)$ となる。このプロセスを採用することにより (16) 式の繰り返し演算をせずに直接グループ化が可能となる。プロセス [3] は (3) 式に対応するが p の類別関数をAからG

への写像ではなく、 B から G へのそれと変更している。グループは一連番号で表すことにしているので類別関数 p によって分割された B に対して、

$$(b_1 \cdots b_L)' \xrightarrow{p} (1 \cdots L)'$$

とする。

2.1 グループ化の事例

表3にある対応関係コード表を具体例として実際にこの処理をおこなってみる。規準となる分類 X は左側の一連番号である。 X_A は分類 A のコード、 X_{A-Q} は分類 A のコードにおける昇順に並べられた一連番号、 X_B は分類 B のコード、 X_{B-Q} は分類 B のコードにおける昇順に並べられた一連番号を示す。最初の対応関係において分類 A の X_{A-Q} は連番1であり、これに分類 B の X_{B-Q} の連番2が対応している。分類 A の2番目は同じく連番1であり、それに分類 B の3が対応している。この2つの対応関係において分類 A が共通なので同一グループに属しており、

表3 分類 A と分類 B における対応関係コード表の例

番号	分類A	分類B	分類A	分類B	分類Bの変更								
	X_A	X_B	X_{A-Q}	X_{B-Q}	1	2	3	4	5	6	7	8	9
1	51561	51562	1	2	2	2	2	2	2	2	2	2	2
2	51561	51563	1	3	3	2	2	2	2	2	2	2	2
3	51561	51564	1	4	4	4	4	2	2	2	2	2	2
4	51562	51561	2	1	1	1	1	1	1	1	1	1	1
5	51562	51565	2	5	5	5	5	5	5	1	1	1	1
6	51562	51566	2	6	6	6	6	6	6	6	6	1	1
7	51563	51563	3	3	3	2	2	2	2	2	2	2	2
8	51564	51562	4	2	2	2	2	2	2	2	2	2	2
9	51564	51564	4	4	4	4	4	2	2	2	2	2	2
10	51565	51563	5	3	3	2	2	2	2	2	2	2	2
11	51565	51564	5	4	4	4	4	2	2	2	2	2	2
12	51566	51567	6	7	7	7	7	7	7	7	7	7	3
13	51567	51568	7	8	8	8	8	8	8	8	8	8	4
14	51568	51568	8	8	8	8	8	8	8	8	8	8	4
15	51569	51562	9	2	2	2	2	2	2	2	2	2	2

(出所) 著者作成

分類Bとの関わりで同じグループのメンバーになるためには同一の識別記号が必要になる。分類Aの連番1には既に分類Bの連番2が付けられているので連番の2を識別記号とみなして分類Bの連番3のすべてを連番2に置き換えることで分類Bに同一識別記号を与える。表の「分類Bの変更」の(1)にその結果が示されている。

分類AのX_{A-f}の3番目は同じく連番1であり、それに分類Bの4が対応しており、連番の4はこの同一グループのメンバーである。分類Aの連番1に属すグループには既に分類Bの連番2が付けられているので表の「分類Bの変更」の(2)のように分類Bの連番4のすべてを連番2に置き換え同一識別記号を与える。

分類Aの4番目は連番2であり、それに分類Bの1が対応している。分類Aに関してはその前の連番1とは異なったグループなのでそのままにしておく。分類Aの5番目は同じく連番2であり、それに分類Bの5が対応している。分類Aの連番2には既に分類Bの連番1が付けられているので分類Bの連番5のすべてを連番1に置き換える。表1の分類Bの(3)にその結果が示されている。

この処理を最後まで続ける。表1の分類Bの変更の(4)に最終結果が示されている。この項目の分類Bが同じ連番を持っていればそれは同一グループのメンバーになる。この例では結果として、連番の2を持つコードが9個でグループを構成し、連番の1を持つものが3個、連番の7が1個、連番の8が2個でそれぞれグループを構成し、4つのグループに分かれたことになる。最後に、この連番の1、2、7、8を改めて一連番号の1、2、3、4に付け替える。新しい一連番号がグループの一連番号に対応する。レコード番号で対応関係を表現すれば、4つのグループは $G_1=\{4,5,6\}$ 、 $G_2=\{1,2,3,7,8,9,10,11,15\}$ 、 $G_3=\{12\}$ 、 $G_4=\{13,14\}$ となる。また、この4つのグループにグループのタイプ分けをするとタイプ2、タイプ4、タイプ1、タイプ3にそれぞれ分類される。特にすべてが同じ番号の1になれば1つのグループしか存在しないことを表す。

2.2 プログラムClcVP6_P.pliおよびその使用法

基本モデルとしての分類Aと分類Bの対応関係をグループ化するためのPL/IによるプログラムがClcVP6_P.pliである。プログラムはINpD、@ClcVP6、OutDの3つのサブプログラムから構成されている。サブプログラムINpDにより対

応関係のデータを配列aに読み込む。a(*,1)は分類A、a(*,2)は分類B、に対応する配列、nは対応関係の個数である。グループ化をするサブプログラムが@ClcVP6(a,n,s,b,q)である。配列q(*,3)にグループとしての一連番号、配列s(*)はグループのタイプ、配列b(*,1)は分類Aの頻度、b(*,2)は分類Bの頻度である。q(*,1)は全体における昇順に並べられた分類Aに対する分類コードの一連番号、q(*,2)は全体における昇順に並べられた分類Bに対する分類コードの一連番号である。OutDにより対応関係の情報を出力する。

プログラムClcVP6_P.pliの入力データは分類Aの分類コードとして X_A (01-10)、分類Bの分類コードとして X_B (10-20) が長さ10の文字列で与えられることになっている。各分類コードの後にくる () の数字はカラムを表している。例えば、表4に示されているように入力データを与えたとする。空白であっても20カラムまではデータを入力する必要があるので注意すること。すなわち、51561.....51562....となるようにデータの長さとして20桁以上が必要である。.....は5桁の空白を表す。で表された空白は必要であるがデータの長さ

表4 プログラムClcVP6_P.pliに対する入力データの例

X_A	X_B
51561	51562
51561	51563
51561	51564
51562	51561
51562	51565
51562	51566
51563	51563
51564	51562
51564	51564
51565	51563
51565	51564
51566	51567
51567	51568
51568	51568
51569	51562

(出所) 著者作成

表5 グループ化プログラムの出力データの例

G_i	(j)	t	X_A	X_B	X_{A-f}	X_{B-f}	X_{A-Q}	X_{B-Q}
1	1	2	51562	51561	3	1	1	1
1	1	2	51562	51565	3	1	2	5
1	1	2	51562	51566	3	1	2	6
2	1	4b	51561	51562	2	2	1	2
2	1	4b	51561	51563	2	3	1	3
2	1	4b	51563	51563	1	3	3	3
2	1	4b	51564	51564	1	2	4	4
2	1	4b	51565	51563	2	3	5	3
2	1	4b	51565	51564	2	2	5	4
2	1	4b	51569	51562	1	2	9	2
3	1	1	51566	51567	1	1	6	7
4	1	3	51567	51568	1	2	7	8
4	1	3	51568	51568	1	2	8	8

(出所) 著者作成

は必ずしも固定長でなくともかまわない。

表4を入力としてClcVP6_P.exのプログラムを実行させると、その結果は表5のように示される。ここで、対応関係の基本モデルで示された各項目の記号とそれが示す内容は次のように表される。 $G_i(j)$: グループおよびサブグループのを表し、 i はグループの一連番号 (01-04)、 j はそのサブグループの一連番号 (05-08) である。基本モデルの対応関係ではサブグループは存在しないので、グループ化された j はすべて1となっている。 $Type$: サブグループの対応関係のタイプ (10-11) を表す^(注8)。 X_A : 分類Aの分類コード (13-22)、 X_B : 分類Bの分類コード (23-32) を表す。 X_{A-f} : 分類Aの分類コードの頻度 (33-36)、 X_{B-f} : 分類Bの分類コードの頻度 (37-41) を表す。 X_{A-Q} : 分類A内で分類コードを昇順に並べたときの一連番号 (42-46)、 X_{B-Q} : 分類B内で分類コードを昇順に並べたときの一連番号 (47-52) を表す。() 中の数字はカラム数を示す。

プログラムを実行するためのスクリプトファイルはつぎのようにして与えられる。

```

#*-----*
#|                ClcVP6_P                |
#*-----*
#! /bin/ksh
LANG=en_US

echo "[                ClcVP6_P                ]"
export DD_FI=clcvp6_p.inp
export DD_FO=clcvp6_p.out,'append(n)'
/u/noday/pli/link/clcvp6_p.x

```

プログラムClcVP6_P.pliについてはプログラム編を参照すること。サブプログラム@clcvp6.pliは本章の中で最も利用される頻度の多い外部サブプログラムであるが詳細については省略する。

3. 切断にもとづくサブグループ化の方法

グループに含まれている対応関係の中から切断の要素となる対応関係を取り除いて得られた対応関係は切断されたグループといわれ、切断されたグループをグループ化することによりこのグループは複数のサブグループによって分割される。切断モデルにはグループ内に切断の要素も含んでおり、その要素はサブグループの特別な状態として解釈される。

3.1 プログラムClcVP7_P.pli

切断にもとづくサブグループ化のためのPL/IプログラムがClcVP7_P.pliである。このプログラムはグループ化のためのプログラムClcVP6_P.plにより得られた結果を入力データとして使用する。この入力データに対して切断の要素を決めることにより対応関係をサブグループ化することが可能となる。プログラムClcVP7_P.pliは基本的にはINpD、@clcvp7、OutDのサブプログラムから構成される。サブプログラムINpDはそこにサブプログラムSLcFを含み、グループごとにサブグループ化の処理をおこない、切断の要素を配列ad、サブグループ化する対応関係のデータを配列aに読み込む。ad(*,1)およびa(*,1)は分類Aに対応し、ad(*,2)およびa(*,2)は分類Bに対応する。nは対応関係の個数、ncは切断の要素の個数である。切断の要素に従って対応関係をサブグループ化をする

サブプログラムが@ClcVP7 (n2,a,n,s,b,q,ad,nc,q2,cc) である。このサブプログラムのパラメターのうち入力情報はn2,a,n,adであり、残りは出力情報である。

配列q(*,3)にサブグループとしての一連番号、s(*)はサブグループのタイプ、b(*,1)は分類Aの頻度、b(*,2)は分類Bの頻度である。q(*,1)はグループ内における昇順に並べられた分類Aに対する分類コードの一連番号、q(*,2)はグループ内における昇順に並べられた分類Bに対する分類コードの一連番号である。Q2(*,1)は切断の要素である分類Aの分類コードが所属するサブグループの一連番号、q(*,2)は切断の要素である分類Bの分類コードが所属するサブグループの一連番号である。ccはグループ内の対応関係のタイプ別対応関係の数を表わし、配列のcc(1)からcc(5)に対して順に1,2,3,4a,4bが対応する。

最後に、対応関係の情報はOutDにより切断の要素とサブグループの対応関係をそれぞれ分けて出力するが、それらを同一ファイルへ出力するため、sortで並び替えることによりグループごとの並びに調整できる。

3.2 プログラムClcVP7_P.pliの使用法

プログラムClcVP7_P.pliの入力は、前述したようにプログラムClcVP6_P.pliの出力データを利用する。切断の要素はこの出力データにおけるサブグループの項目、すなわち、グループ $G_i(j)$ における j であり、サブグループの一連番号を表わしている項目の1を0に置き換えることで設定する。例えば、表3において上から5番目の対応関係を切断の要素にするには、

2 0 4b 51561 51563 2 3 1 3

と変更し、このデータを入力ファイルとして指定する。

出力は表6に示されている。グループ2のサブコードが0,1,2と変わり、サブグループ化されたことが確かめられる。また、出力のフォーマットは次のように示される。 i : グループ $G_i(j)$ における i でありグループの一連番号 (01-04)、 j : $G_i(j)$ の j でありサブグループの一連番号 (05-08)、*Type* : サブグループにおける対応関係のタイプ (10-11)、 X_A : 分類Aの分類コード (13-22)、 X_B : 分類Bの分類コード (23-32)、 X_{A-f} : 分類Aの分類コードの頻度 (33-36)、 X_{B-f} : 分類Bの分類コードの頻度 (37-41)、 X_{A-Q} : 分類Aの分類コードが属するグループ内の昇順に並べた一連番号 (42-46)、 X_{B-Q} : 分類Bの分類コードが属するグループ内の昇順に並べた一連番号 (47-52) である。()の中の数字

表6 サブグループ化プログラムの出力データの例

G_i	(j)	t	X_A	X_B	X_{A-f}	X_{B-f}	X_{A-Q}	X_{B-Q}
1	1	2	51562	51561	3	1	1	1
1	1	2	51562	51565	3	1	1	2
1	1	2	51562	51566	3	1	1	3
2	0	0	51561	51563	0	0	1	2
2	1	3	51561	51562	1	2	1	1
2	1	3	51569	51562	1	2	5	1
2	2	4a	51563	51563	1	2	2	2
2	2	4a	51564	51564	1	2	3	3
2	2	4a	51565	51563	2	2	4	2
2	2	4a	51565	51564	2	2	4	3
3	1	1	51566	51567	1	1	1	1
4	1	3	51567	51568	1	2	1	1
4	1	3	51568	51568	1	2	2	1

(出所) 著者作成

はカラム数を示す。ただし、切断の要素に対しては、 X_{A-Q} ：分類Aの分類コードが属するサブグループの一連番号、 X_{B-Q} ：分類Bの分類コードが属するサブグループの一連番号と解釈する。

プログラムを実行するためのスクリプトファイルはつぎのようにして与えられる。

```

#*-----*
#|                ClcVP7_P                |
#*-----*
#! /bin/ksh
LANG=en_US

echo "[                ClcVP7_P                ]"
export DD_Fl=clcvp6_p.out
export DD_F0=clcvp7_p.out, 'append(n)'
export DD_FN=clcvp7_p.n, 'append(n)'
/u/noday/pli_link/clcvp7_p.x

```

プログラムclcvp7_p.pliについてはプログラム編を参照すること。外部サブプログラム@clcvp7.pliについては省略する。

4. 対応関係を連結する方法

対応関係の連結は要約すると、もう既に連結されている分類 A_1, A_2, \dots, A_k とグループ化された A_k, A_{k+1} を A_k をもとに再度グループ化することであり、前述したグループの連結の過程はつぎのようになる。

(1) 分類 A_1, A_2, \dots, A_k の連結されたグループ CG_{k-1} が与えられており、 $FCD(A_k, A_{k+1} : X) \rightarrow G_k$ が得られたとする。このとき、連結の規準となる分類 A_k にもとづいて CG_{k-1} と G_k の対応関係を作成する。

(2) CG_{k-1} と G_k の対応関係により FCD を求め、 $FCD(CG_{k-1}, G_k : A_k) \rightarrow CG_k$ として CG_k を求める。

4.1 プログラムClcVP9.pli

対応関係の連結に必要とされるプログラムがClcVP9.pliである。このプログラムは連結されたグループ CG_{k-1} と連結の軸となる分類 A_k を持つ新たな対応関係のグループ G_k を読み込み、 CG_k を出力する機能を持つ。プログラムClcVP9.pliは基本的にはデータ入力のためのサブプログラムINpCGとINpG、分類 A_k を規準として CG_{k-1} と G_k の対応関係を作成し、その対応関係をグループ化するConGP、データを出力するOutDから構成されている。

ConGPではまず分類 A_k を規準として CG_{k-1} と G_k の対応関係を作成する。 $v \in A_k$ とするとき、 CG_{k-1} における CG_{k-1} と A_k の関係は、(15) 式から得られ、

$$(18) \quad \{p_{CG_{k-1}}(v), v\} = \{cg_{k-1}, v\}$$

であり、この関係はINpCGにより (18) 式の左辺がデータとして与えられる。同じく、 G_k における G_k と A_k の関係は、(15) 式から得られ、

$$(19) \quad \{v, p_{G_k} f_a^{-1}(v)\} = \{v, g_k\}$$

であり、この関係はINpGにより (19) 式の左辺がデータとして与えられる。また、(15) 式に示されているように分類 A_k を規準とした CG_{k-1} と G_k の対応関係は $v \in A_k$ に対して

$$(20) \quad \{p_{CG_{k-1}}(v), p_{G_k} f_a^{-1}(v)\}$$

として得られたものの中から重複を取り除いたものとして表わされる。しかし、(20)式は与えられていないので(19)式と(18)式から作成する必要がある。 v の個数が n であるとして v を昇順に並べ($v_1' \dots v_n'$)として配列 c に入れる。

(18)式の左辺の v が c の何番目にあるかを@BSchで探し p 番目であるとする。配列 $a(p,1,3)$ へ(17)式の左辺の cg_{k-1} を入れる。同じように(18)式の左辺の v が c の何番目にあるかを@BSchで探し p 番目であるとする。配列 $a(p,2,3)$ へ(18)式の左辺の g_k を入れる。このようにして作成された $a(*,*,3)$ が分類 A_k を基準とした CG_{k-1} と G_k の対応関係であり、前者が $a(*,1,3)$ 、後者が $a(*,2,3)$ に保存されている。 $a(*,*,3)$ を@ClcVP6によりグループ化することで両者が連結される。

OutDでは表7で示されるように連結された対応関係に対して、 k により A_k と A_{k+1} の対応関係を示し、その対応関係の中のグループおよびその対応関係を出力する。詳細はClcVP9.pliの使用法を参照すること。

4.2 プログラムClcVP9.pliの使用法

プログラムClcVP9.plの入力データは G_k と CG_{k-1} の2つからなる。 G_k は対応関係の基本モデルのプログラムClcVP6_P.pliまたは切断モデルClcVP7_P.pliの出力データを利用する。データフォーマットについてはプログラムClcVP6_P.pliおよびClcVP7_P.pliの使用法において既に説明されているのでそこを参照すること。この両モデルのフォーマットは X_a-Q と X_b-Q の項目において昇順に並べられた一連番号の付け方が異なるだけで形式的には同一である。また、フォーマットは同一であるが G_k を漸化式で表現するためデータフォーマットの項目の名称をつぎのように変更する。分類 A の分類コードを $X_A \rightarrow A_k$ 、分類 B の分類コードを $X_B \rightarrow A_{k+1}$ とそれぞれ変更する。同じようにして、分類コードの頻度を $X_{A-f} \rightarrow A_{k-f}$ 、 $X_{B-f} \rightarrow A_{k+1-f}$ 、分類コードの一連番号については $X_{A-Q} \rightarrow A_{k+1-Q}$ 、 $X_{A-Q} \rightarrow A_{k+1-Q}$ と変更する。 G_k において $k=2$ のときの入力データフォーマットの例が表7に示されている。

プログラムClcVP9.plの入力データのもう1つのデータである CG_{k-1} については次のように示される。分類 A_1, A_2, \dots, A_k の連結されたグループ CG_{k-1} は対にされる (A_m, A_{m+1}) , $m=1 \dots k-1$ の対応関係のグループで表現される。すなわち、煩雑さを避けるためにサブグループがなくすべて $j=1$ であるとするとき、

表7 G_k において $k=2$ のときのグループ化された対応関係

G_i	(j)	t	A_k	A_{k+1}	A_k-f	$A_{k+1}-f$	A_k-Q	$A_{k+1}-Q$
1	1	1	0251	0251	1	1	1	1
2	1	1	0252	0252	1	1	2	2
3	1	1	04211	0421	1	1	3	3
4	1	1	04212	0422	1	1	4	4
5	1	4a	08111	08111	1	2	5	5
5	1	4a	08119	08111	3	2	6	5
5	1	4a	08119	08119	3	1	6	6
5	1	4a	08119	08199	3	2	6	8
5	1	4a	08199	08195	2	1	7	7
5	1	4a	08199	08199	2	2	7	8
6	1	1	09808	09892	1	1	8	9
7	1	4a	29194	29196	1	2	9	11
7	1	4a	29199	29194	3	1	10	10
7	1	4a	29199	29196	3	2	10	11
7	1	4a	29199	29199	3	1	10	12

(出所) 筆者作成

(注) この表の A_k と表9の A_{k+1} が対応する

CG_{k-1} におけるグループ cg_i に対して対応関係 (A_m, A_{m+1}) のグループを、

$$G(m, h), \quad m=1 \cdots k, \quad h=1 \cdots n_m$$

とすれば、 m に対して対応関係が表8のように与えられる。したがって、出力のデータフォーマットは、 $i: CG_i(j)$ の i であり連結されたグループの一連番号 (01-04)、 $j: CG_i(j)$ の j であり連結されたサブグループの一連番号 (05-08)、 $C-Type$: 連結されたサブグループにおける対応関係のタイプ (10-11)、 k : 対応関係が分類 A_k と A_{k+1} にであることを示す (13-14)、 $h: G(k, h)(j)$ の h であり分類 A_k と A_{k+1} におけるグループの一連番号 (16-19)、 $j: G(k, h)(j)$ の j であり A_k と A_{k+1} におけるサブグループの一連番号 (21-23)、 $Type: A_k$ と A_{k+1} における対応関係のタイプ (25-26)、 A_k : 分類 A_k の分類コード (28-37)、 A_{k+1} : 分類 A_{k+1} の分類コード (38-47)、 A_k-f : 分類 A_{k+1} の分類コードの頻度 (48-52)、 $A_{k+1}-f$: 分類 A_{k+1} の分類コードの頻度 (53-57) である。 CG_2 の出力のデータフォーマットが表10に示されている。表10において影の付いているところが $k=1$ であり、 k

表8 プログラムClcVP9.pliの出力データ CG_k のパターン

連結されたグループ	m	対応関係のグループ	対応関係
cg_i	1	$G(1,1)$	$G(1,h)$, $h=1\cdots n_1$ に属する A_1 と A_2 の対応関係
	⋮	⋮	
	1	$G(1,n_1)$	
	⋮		
	⋮		
	k	$G(k,1)$	$G(k,h)$, $h=1\cdots n_k$ に属する A_k と A_{k+1} の対応関係
	⋮	⋮	
	k	$G(k,n_k)$	

(出所) 筆者作成

表9 CG_{k-1} において $k=2$ のときの連結された対応関係

CG_i	(j)	ct	k	$G(k,h)$	(j)	t	A_k	A_{k+1}	$A_k \cdot f$	$A_{k+1} \cdot f$
1	1	4a	01	1	1	4a	0250	0251	3	1
1	1	4a	01	1	1	4a	0250	0252	3	1
1	1	4a	01	1	1	4a	0250	09808	3	2
1	1	4a	01	1	1	4a	29159	29199	1	2
1	1	4a	01	1	1	4a	29199	09808	2	2
1	1	4a	01	1	1	4a	29199	29199	2	2
2	1	2	01	2	1	2	0421	04211	2	1
2	1	2	01	2	1	2	0421	04212	2	1
3	1	1	01	3	1	1	08111	08111	1	1
4	1	1	01	4	1	1	08119	08119	1	1
5	1	1	01	5	1	1	08199	08199	1	1
6	1	1	01	6	1	1	29194	29194	1	1

(出所) 筆者作成

わされる。影の付いていないところが $k=2$ であり、 k の項目は02で表示され、対応関係は (A_2, A_3) から得られるグループとして表わされる。

プログラムClcVP9.pliにおいて初期値として与えられる入力データは CG_1 と G_2 である。前者は $k=1$ のみからなる対応関係であり、その例は表9で示される。

表10 CG_{k-1} と G_k から作成された CG_k の連結された対応関係 ($k=2$)

CG_i	(j)	ct	k	$G(k,h)$	(j)	t	A_k	A_{k+1}	A_k	f	A_{k+1}	f
1	1	4a	01	1	1	4a	0250	0251	3	1		
1	1	4a	01	1	1	4a	0250	0252	3	1		
1	1	4a	01	1	1	4a	0250	09808	3	2		
1	1	4a	01	1	1	4a	29159	29199	1	2		
1	1	4a	01	1	1	4a	29199	09808	2	2		
1	1	4a	01	1	1	4a	29199	29199	2	2		
1	1	4a	01	6	1	1	29194	29194	1	1		
1	1	4a	02	1	1	1	0251	0251	1	1		
1	1	4a	02	2	1	1	0252	0252	1	1		
1	1	4a	02	6	1	1	09808	09892	1	1		
1	1	4a	02	7	1	4a	29194	29196	1	2		
1	1	4a	02	7	1	4a	29199	29194	3	1		
1	1	4a	02	7	1	4a	29199	29196	3	2		
1	1	4a	02	7	1	4a	29199	29199	3	1		
2	1	2	01	2	1	2	0421	04211	2	1		
2	1	2	01	2	1	2	0421	04212	2	1		
2	1	2	02	3	1	1	04211	0421	1	1		
2	1	2	02	4	1	1	04212	0422	1	1		
3	1	3	01	3	1	1	08111	08111	1	1		
3	1	3	01	4	1	1	08119	08119	1	1		
3	1	3	01	5	1	1	08199	08199	1	1		
3	1	3	02	5	1	4a	08111	08111	1	2		
3	1	3	02	5	1	4a	08119	08111	3	2		
3	1	3	02	5	1	4a	08119	08119	3	1		
3	1	3	02	5	1	4a	08119	08199	3	2		
3	1	3	02	5	1	4a	08199	08195	2	1		
3	1	3	02	5	1	4a	08199	08199	2	2		

(出所) 筆者作成

後者はもう既に紹介した表7である。この2つの対応関係の連結はそれぞれの表の中に影を付けた分類 A_2 をもとにして実行される。

おわりに

対応関係を連結する方法については繰返し説明しているように、もう既に連

結されている分類 A_1, A_2, \dots, A_k とグループ化された A_k, A_{k+1} を A_k をもとに再度グループ化することである。このプロセスは、(1) 初期値として $k=1$ を与えて $FCD(A_k, A_{k+1} : X) \rightarrow G_k$ を作成し、これを CG_k とおく、(2) 分類 A_1, A_2, \dots, A_k の連結されたグループ CG_{k-1} が与えられているので、

$$FCD(A_k, A_{k+1} : X) \rightarrow G_k$$

を求める、(3) 連結の規準となる分類 A_k にもとづいて CG_{k-1} と G_k の対応関係を作成する、(4) CG_{k-1} と G_k の対応関係により FCD を求め、

$$FCD(CG_{k-1}, G_k : A_k) \rightarrow CG_k$$

として CG_k を求める、ことから構成される。このプロセスにおいて (1) と (2) はプログラム ClcVP6_P.pli で実行され、(3) と (4) はプログラム ClcVP9.pli で実行される。

対応関係を次々と連結していくと場合によってはグループが大きくなりすぎてそのグループの特性が見えなくなってくることもある。特性を維持させるためには A_k, A_{k+1} をグループ化するさいに可能な限りグループの切断をおこなないサブグループを作っておく方がいい。切断の要素にもとづいてサブグループ化するのがプログラム ClcVP7_P.pli である。

したがって、対応関係を連結するにあたっては機械的にプログラムを実行するのではなく、対応関係の特性をできるだけ生かすようなサブグループ化を通して処理をしていく必要がある。

(注1) 佐藤による「要約データの基礎概念とデータベース内での推論—貿易統計データを例として—」によれば、分類 X と Y はデータベース内の分類階層内に定義されているものとする。この時、 X と Y の違いを識別できるような分類 B を同じ分類階層内に見出すことができるものと仮定する。 B から X への類別関数を f 、 B から Y への類別関数を g とする。この f と g より、 R^* が計算でき、これを元に FCD の条件を満足するような分類 Z と、 X から Z への類別関数 p を計算することができる。

(注2) 本章で使用している分類の X, A, B, G は佐藤の分類の B, X, Y, Z にそれぞれ対応する。また、本章で使用している類別関数の $f_{a/b}$ は佐藤の f, g にそれぞれ対応する。 p については同じように対応している。

(注3) $R^*(a_i) = p^{-1}(g_i)$, $i = 1 \dots m$ を利用すれば R^* によって分割された A は A/R^* で表

わされ、 $A/R^* = \{a \mid R^*(a_i) = p^{-1}(g_i), g_i \in G, i = \dots m\}$ から、

$$\begin{aligned} A/R^* &= \{R^*(a_i) \mid R^*(a_i)\} = p^{-1}(g_i), g_i \in G = \{p^{-1}(g_i) \mid g_i \in G\} \\ &= \{p^{-1}(g) \mid g \in G\} \end{aligned}$$

が導かれる。

(注4) 本来ならば、 f_a の逆写像は、 $f_a^{-1}(a_1) = \{x_i \in X \mid x_i = f_a(a_1), a_1 \in A\}$ と表されるが、 A, X, B の1対1の対応関係を明示的に示すため対応する要素をベクトルで表示して示すことにする。

(注5) このプロセスはAとBの間が直接に関係する対応関係の存在を想定していない。実際の場面ではある分類の対応関係は分類Xを経由せずに直接的なAとBの間に対応関係が存在している。そのため、上記のFCDはより簡単に示すことができる。すなわち、 $h = f_b f_a^{-1}$ とすることにより、 $h: A \rightarrow B$ 、 $p: A \rightarrow G$ となる。このとき、 $A/R^* = \{p^{-1}(g) \mid g \in G\}$ であるなら、 G はAとBのFCDである。ここで、 $R^*(a) = (h \cdot h^{-1})^n(a)$ as $n \rightarrow \infty$ と表わす。繰り返しが1回目の結果は次の通りである。

$$a_1 \xrightarrow{h} b_1 \xrightarrow{h^{-1}} (a_1 \ a_2)'$$

分類Aの a_1 から始まってもう一度分類Aに戻ったときには異なった結果 $(a_1, a_2)'$ を得る。そこで、2度目の繰り返しを実行する。

$$(a_1 \ a_2)' \xrightarrow{h} (b_1 \ (b_1 \ b_2)') \xrightarrow{h^{-1}} (a_1 \ (a_1 \ a_2)') \Rightarrow (a_1 \ a_2)'$$

$(a_1, a_2)'$ から始まって $(a_1, a_2)'$ で終わっているので、この状態で収束したことになるので、繰り返しの計算を中止する。ここで得られた $(a_1, a_2)'$ を新たな分類Gとして、上記と同じように分類Gへの類別関数pをで定義したFCDの条件を満足するような分類Gと、類別関数pが得られる。対応関係のグループをまとめることが可能であるということは、類別関数 $h \cdot h^{-1}$ が収束するまで繰り返しの処理をしてグループ化できるということである。

(注6) 実際の場面では分類の内容を個別に検討して比較的關係がなさそうだと判断される対応関係を切断するという方法を採用している。この方法を採用するには対象としている分類およびその対応関係についての専門的な知識が要求される。

(注7) ここでいうサブグループとは厳密な意味におけるグループ内の分割ではなくて互いに少数の共通の分類コードが含まれていても類似のものがまとまっているという意味でのゆるやかな対応関係の集まりをさす。したがって、サブグループは一意的には決まらない。

(注8) グループ化された対応関係は次のような4つの対応関係のタイプに分けることができる。対応関係が分類Aから分類Bのへ向かう方向を持っているとする。対応関係のタイプ1は分類Aと分類Bの個別分類コードが1対1に対応する関係である。このタイプではグループに含まれる対応する分類コードの個数は1個である。タイプ2は分類A

と分類 B が 1 対多の対応関係である。グループに含まれる対応する分類コードの個数は分類 B に含まれる分類コードの個数に等しい。タイプ 3 は分類 A と分類 B が多対 1 の対応関係であり、タイプ 2 とは逆にグループに含まれる分類コードの個数は分類 A に含まれる分類コードの個数に等しい。タイプ 4 は分類 A と分類 B が多対多の対応関係である。このタイプのグループに含まれる分類コードの個数について特に決まったパターンはない。タイプ 4 はさらにタイプ 4a と 4b とに分かれる。前者は配分ウエイトが代数方程式により一意的な解を持つタイプ、後者は一意的な解を持たないタイプである。

【参考文献】

[1] 佐藤英人「要約データの基礎概念とデータベース内での推論—世界貿易統計データベースを中心として—」（木下宗七、野田容助編『世界貿易データシステムの整備と利用』統計資料シリーズ No.67 アジア経済研究所 1995）

[2] 野田容助、山本泰子「体系の異なる分類の対応関係と変換—グループ化および切断による商品分類の変換の試み—」（木下宗七、野田容助編『世界貿易データシステムの整備と利用』統計資料シリーズ No.67 アジア経済研究所 1995）

[3] 野田容助「商品分類の改定に伴う対応関係の連結」（古河俊一、野田容助共著『標準国際貿易商品分類と産業分類の対応関係』統計資料シリーズ No.80 アジア経済研究所 1998）

[4] Hideto Sato, *Fundamental Concepts of Social/Regional Summary Data and Inference in Their Database*, Doctoral thesis, Tokyo University, 1983.

プログラム編

[1] プログラムClcVP6_P.pli

```
*process gonumber;
/*-----**
|                               ClcVP6_P                               |
**-----*/
ClcVP6_P: proc options(main);
dcl @TmCP entry      (char(15),fixed bin(31)) external,
    m                fixed bin(31) init(0);
    call INpD_P;
    call @TmCP(' #inp',m);
/*-----**
|                               main                               |
**-----*/
begin;
dcl (@ClcVP6 entry  (*,fixed bin(31),*,*,*)) external,
    a(m,2)          char(10),
    s(m)            char(2),
    (b(m,2),q(m,3),n) fixed bin(31);
    on attn stop;
    call INpD;
    call @ClcVP6(a,n,s,b,q);
    call OutD;
/*-----**
|                               INpD                               |
**-----*/
INpD: proc;
dcl f                file input stream;
    on endfile(f);
    open file(f) title('f');
    n=1;
    get file(f) edit(a(n,1),a(n,2))(col(1),(2)a(10));
    do while(^endfile(f));
        n=n+1;
        get file(f) edit(a(n,1),a(n,2))(col(1),(2)a(10));
    end;
    n=n-1;
    close file(f);
end INpD;
/*-----**
|                               OutD                               |
**-----*/
```

```

OutD:  proc;
dcl i          fixed bin(31),
    fo        file output stream;
open file(fo) output title(' fo');
do i=1 to n;
    put file(fo) edit(q(i,3), ' 1 ', s(i), a(i,1), a(i,2),
    b(i,1), b(i,2), q(i,1), q(i,2))
    (col(1), f(4), a(5), a(2), x(1), (2)a(10), (4)f(5));
end;
close file(fo);
end OutD;
/*-----**
|                end of main                |
**-----*/
end;
/*-----**
|                INpD_P                      |
**-----*/
INpD_P:  proc;
dcl a          char(1) based(p),
    p          pointer,
    f          file input record;
on endfile(f);
open file(f) title(' fi');
read file(f) set(p);
do while( ^endfile(f) );
    m=m+1;
    read file(f) set(p);
end;
close file(f);
end INpD_P;
end ClcVP6_P;

```

[2] プログラムClcVP7_P.pli

```

*process gonumber;
/*-----**
|                ClcVP7_P                    |
**-----*/
ClcVP7_P:  proc options(main);
dcl n2          fixed bin(31) init(0);
call INpD_P;
/*-----**
|                main                        |
**-----*/

```



```

SetCC: proc;
    gp2=gp;
    n,nc=0;
    a,ad=' ';
end SetCC;
/*-----**
|                               SLcF                               |
**-----*/
SLcF: proc(c, a2);
dcl a2(2)                char(10),
    c                    char(1);
    m=m+1;
    if c^='0' then do;
        n=n+1;
        a(n,*)=a2;
    end;
    else do;
        nc=nc+1;
        ad(nc,*)=a2;
    end;
end SLcF;
/*-----**
|                               OutD                               |
**-----*/
OutD: proc;
dcl i                    fixed bin(31);
    call @ClcVP7(n2, a, n, s, b, q, ad, nc, q2, cc);
    do i=1 to nc;
        put file(fo) edit(gp2, ' 0 0 ', ad(i,1), ad(i,2),
            ' 0 0', q2(i,1), q2(i,2))
            (col(1), f(4), a(8), (2)a(10), a(10), (2)f(5));
    end;
    do i=1 to n;
        put file(fo) edit(gp2, q(i,3), s(i), a(i,1), a(i,2),
            b(i,1), b(i,2), q(i,1), q(i,2))
            (col(1), (2)f(4), x(1), a(2), x(1), (2)a(10), (4)f(5));
    end;
    k=k+nc+n;
    if sum(cc)>0 then do;
        put file(fn) edit(gp2, (cc(i) do i=1 to 5))
            (col(1), f(4), x(5), (5)f(5));
    end;
end OutD;
/*-----**
|                               end of main                               |
**-----*/

```

```

end;
/*-----**
|                               INpD_P                               |
**-----*/
INpD_P: proc;
dcl 1 rec          based(p),
    2 a            char(1),
    p              pointer,
    f              file input record;
    on endfile(f);
    open file(f) title('f');
    read file(f) set(p);
    do while( ^endfile(f) );
        n2=n2+1;
        read file(f) set(p);
    end;
    close file(f);
end INpD_P;
end ClcVP7_P;

```

[3] プログラム ClcVP9.pli

```

*process gonumber;
/*-----**
|                               ClcVP9                               |
**-----*/
ClcVP9: proc options(main);

dcl @TmCP entry      (char(15), fixed bin(31)) external,
    m                fixed bin(31) init(0);
    call MaxM(m);
    call @TmCP('inp', m);

/*-----**
|                               main                               |
**-----*/
begin;
dcl (@ClcVP6 entry  (*, fixed bin(31), *, *, *),
    @BSch  entry    (char(*), *, fixed bin(31), fixed bin(31)),
    @MkTbl entry    (*, *, char(*), fixed bin(31))
    )
    external,
    (a(m, 2, 3), c(m)) char(10),
    d(m, 3)             char(5),
    s(m, 3)             char(2),
    (b(m, 2, 3), q(m, 3, 3), n(3),
    c2(m))              fixed bin(31),

```



```

        end;
    end;
end OutD;
/*-----**
|                               Ref_G                               |
**-----*/
Ref_G:  proc(e, i, k, s2);
dcl (e(m), j, i, k)    fixed bin(31),
    s2                char(2),
    p2                pic'(3)-9';
    e=0;
    do j=1 to n(3);
        if q(j, 3, 3)=i then do;
            p2=a(j, k, 3);
            e(p2)=1;
            s2=s(j, 3);
        end;
    end;
end Ref_G;
/*-----**
|                               end of main                          |
**-----*/
end;
/*-----**
|                               MaxM                                |
**-----*/
MaxM:  proc(m);
dcl (n(2), j, m)      fixed bin(31),
    p                 pic'9',
    f_nm(2)           char(2);
    n=0;
    do j=1 to 2;
        p=j;
        f_nm(j)='f' || p;
        call INpD_P(f_nm(j), n(j));
    end;
    m=max(n(1), n(2));
end MaxM;
/*-----**
|                               INpD_P                              |
**-----*/
INpD_P:  proc(f_nm, n);
dcl a                char(1) based(p),
    p                 pointer,
    n                 fixed bin(31) init(0),
    f_nm              char(2),

```

```
f          file input record;
on endfile(f);
open file(f) title(f_nm);
read file(f) set(p);
do while( ^endfile(f) );
    n=n+1;
    read file(f) set(p);
end;
close file(f);
end INpD_P;
end ClcVP9;
```