Vol.2019-MPS-122 No.5
2019/2/28

属性間の関連度を用いた分解による概念束の単純化

深谷 有吾[1]　石榑 隼人[1]　武藤 敦子[1]　森山 甲一[1]　犬塚 信博[1]

概要：形式概念分析では，概念束を可視化して分析を行うが，データ量が増大すると，複雑化して理解が困難になってしまう．そのために概念束の簡素化を必要とする．また簡素化とは別に，概念束分解の手法がいくつか提案されてきた．従来の分解手法では，分解できるデータに対して制約がある場合や，専門的な知識を必要とするため，実際のデータに適応しづらいという問題があった．本研究では，その問題を解決するために，分解手法の一つである Nested Line Diagram を改良した分解手法を提案する．本手法は，データの情報から関連性のある属性のグループに分けることで，概念束を分解する．また，それらを概念束の和と積を使い作成した概念束に本手法を適用して実験・評価を行う．実験では，互いに関連性のある属性が同じグループに所属され，互いに独立関係にある属性は別のグループに決まることが分かった．これにより実データにおいても，関連性のある属性をグループとした概念束に分解できると考えられる．また，本手法に扱うことのできる，6 つのクラスタリング手法の違いを明確にし，場合によってそれらの手法を変える必要があることを示唆した．

キーワード：形式概念分析，形式文脈，概念束分解，簡素化，階層型クラスタリング，相関係数，Nested Line Diagram

Simplification of Concept Lattice by Decomposition using Degree of Attribute Relevance

1. はじめに

形式概念分析 [1] は，数学的に定義された概念に基づきデータ分析を行う．概念の構造を表す概念束を可視化することで，理解を助けることができる．一方でデータの増大に従い概念束の構造の理解が難しくなってしまう [2]．そのため，概念束の**簡素化**を行う場合がある．簡素化手法の多くが，一部のデータを除去することで，概念束の構造の理解に容易にする．そのため，簡素化を行うと，分析が適切に行われていない可能性がある．また，データを除去せずに概念束の理解を容易にする手法として，**概念束分解**があるが，分解可能なデータに制約があったり，専門知識を必要とすることから，実際の分析にはあまり応用されていない．本研究では，実際の分析に応用できるための，新しい概念束の分解手法を提案する．提案手法では，データの情報から関連性のある属性のグループに分けることで，概念束を分解する．相関係数を用いて定義した関連度を使い，属性どうしの距離行列を作成，階層型クラスタリングを使って，属性のグループ分けを行う．これにより，分解手法のひとつである Nested Line Diagram を使うことで，専門的な知識を有していなくとも，概念束の分解を行うことができる．

2. 形式概念分析

形式概念分析では，対象の集合 G とそれが取り得る属性の集合 M，G と M の間の二項関係 $I \subseteq G \times M$ を扱う．$g \in G$ かつ $m \in M$ である g と m に対して $gIm \Leftrightarrow (g,m) \in I$ である時，「対象 g は属性 m を持つ」という．I は付随関係 (incidence relation) と呼ばれる．G，M，I の三つ組 $\mathbb{K} = (G, M, I)$ を**形式文脈**という．$\mathbb{K}$ は $|G|$ 行 $|M|$ 列の表で表される．この時 gIm であることを，g 行 m 列のセルに $\times$ を記入することで表す．表 1 は形式文脈の一例である．

[1] 名古屋工業大学
Nagoya Institute of Technology

表 1　形式文脈の例
Table 1　Example:Formal Context

名前	卵生（a）	言葉（b）	母乳（c）
ハト（1）	×		
ヒト（2）		×	×
カモノハシ（3）	×		×
ネコ（4）			×

また，データによっては複数の値を持つ属性を持っている場合がある．そのような属性を**多値属性**と呼び，形式文脈では複数の属性で表す．多値属性は複数の属性で表されているが，元は一つの値が決まっているため，それぞれの対象はそれらの属性のうち 1 つを持つことが多い．

$\mathbb{K}$ において，任意の $X \subseteq G$ と $Y \subseteq M$ に対して，式 (1), (2) で定義される写像を定義する．

$$X \mapsto X^I := \{m \in M \mid gIm \ for\ all\ g \in X\} \tag{1}$$

$$Y \mapsto Y^I := \{g \in G \mid gIm \ for\ all\ m \in Y\} \tag{2}$$

X^I は形式文脈 $\mathbb{K}$ において，X のすべての対象が共通して持つ属性の集合を表す．また Y^I は Y のすべての属性を持つ対象の集合を表す．誤解が生じない場合は，X^I および Y^I をそれぞれ，X', Y' と書く．このとき次のように**形式概念**が定義される．

形式文脈 $\mathbb{K} = (G, M, I)$ について，$A \subseteq G, B \subseteq M$ とする．組 (A, B) が $\mathbb{K}$ の形式概念であるとは，$A = B^I$ かつ $B = A^I$ であることを言う．この時，A を**外延**，B を**内包**と呼ぶ．

2 つの形式文脈 $(A_1, B_1), (A_2, B_2)$ に対して式 (3) のように順序を定める．

$$(A_1, B_1) \geq (A_2, B_2) \Leftrightarrow A_1 \supseteq A_2 \Leftrightarrow B_1 \subseteq B_2 \tag{3}$$

式 (3) が成り立つ時，(A_1, B_1) を上位概念，(A_2, B_2) を下位概念という．この順序により $\mathbb{K}$ の形式概念すべてからなる集合は束となる．これを**概念束**と呼ぶ．

表 1 は形式文脈の一例である．また，この形式文脈の概念束が**図 1**(左) であり，その略記が図 1(右) である．

また，形式文脈のサイズが大きくなると，概念束が複雑になり構造の理解が難しくなる．そのためにいくつかの概念束の簡素化手法が提案されてきた [3],[4]．また，別のアプローチとして，概念束の分解手法や別の可視化の手法が提案されてきた [5]．分解手法の一つである Nested Line Diagram では，データに制約はないため，実際の分析への応用がしやすいが，専門的な知識を必要とする．

3. 概念束の簡素化と分解

ここでは概念束の単純化手法の中で，概念束の簡素化と分解の代表的な手法について説明する．

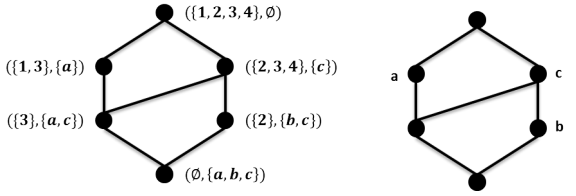


図 1　表 1 から得た概念束のハッセ図（左）とその略記（右）
Fig. 1　Concpet Lattice of Table1

3.1 安定度を用いた手法

Kuznetsov は，安定な形式概念を選択することで概念束を簡素化する手法を提案した [6]．形式概念が安定であるとは，その内包が多くの対象に依存しないことである．形式概念の安定度を測る指標は [6] で提案され，その後 [7] でその定義が修正された．形式文脈 $\mathbb{K} = (G, M, I)$ の形式概念 (A, B) の安定度指標 $\sigma(A, B)$ は，式 (4) のように定義される．

$$\sigma(A, B) = \frac{|\{C \subseteq A \mid C' = B\}|}{2^{|A|}} \tag{4}$$

安定な形式概念の内包は，形式文脈から少数の対象を除いても内包となる．また安定な形式概念の外延は，下位の形式概念の外延と離れている．安定度を用いた簡素化手法では，安定度指標がしきい値以上の形式概念の集合を構成するので，概念束の簡素化を行う．このように構成した形式概念の集合は必ずしも束とはならない．

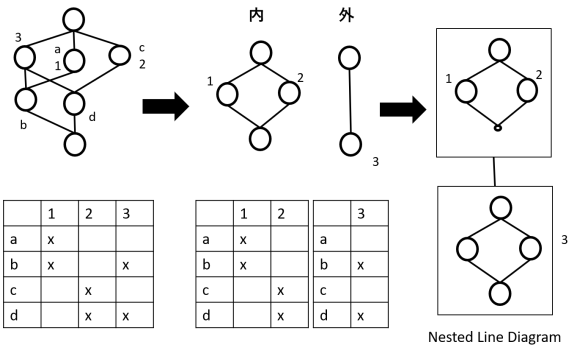
3.2 属性推定を用いた手法

石榑らが提案したこの手法は，類似した 2 つの形式概念の組から得られる関係を用いて，概念束の簡素化を行うことを目的にしている [8]．外延の差が小さい 2 つの異なる形式概念の組からは，特定の属性をもつ対象の多くが別の属性を持つという近似的含意関係が得られる．その近似的含意関係を用いた属性推定を行うことで概念束の簡素化を行う．形式文脈とその否定から関係を抽出する．その後抽出された関係を用いて属性推定を行い形式文脈を更新していく形で概念束の簡素化を行う．

しかし，この手法はノイズの影響を受けやすく，また抽出する近似的含意関係やその優先度に強く依存している．

3.3 Nested Line Diagram

Nested Line Diagram[9] は，形式文脈の属性集合を，排他的に 2 つに分けることで概念束を分解し，可視化を行う手法である．分けた形式文脈から 2 つの概念束を生成し，それぞれ内と外の概念束とする．内と外に分けた概念束は**図 2** のように可視化することで，概念束を整理して観察を行うことができる．また，元の概念束の概念と分解後の概念の対応は，内の概念束と外の概念束の組合せで決まる．元の概念束に対応する概念が存在しないものは，他の概念より小さく表示する．この手法は，属性集合を自由に分割



	1	2	3
a	x		
b	x		x
c		x	
d		x	x

	1	2
a	x	
b	x	
c		x
d		x

	3
a	
b	x
c	
d	x

図 2 Nested Line Diagram の例
Fig. 2 Example:Nested Line Diagram

することで概念束の分解を可能とするため，属性が 2 つ以上存在する形式文脈であれば，どんなデータであっても分解することができる．しかし，どのように属性集合を分割すべきか知るためには，属性ごとの特徴や他の属性の関連性など，専門知識を必要とする．

3.4 Subdirect Decomposition

Subdirect Decomposition[10] は，Nested Line Diagram と同じく，分解後の概念束の概念の組合せによって，元の概念束の概念と対応が決まる．分解を行う際には，まず形式文脈に以下の式 (5), (6) で定義される arrow relation を見つけ出す．それから式 (7) のような，arrow-closed subcontext となる形式文脈に分け，概念束を生成することで概念束の分解を行う．

この手法は，分解を行うことのできる形式文脈に制約がある．そのため，どんな概念束においても分解ができるわけではない．

$$g \swarrow m \iff \neg gIm \wedge (g' \subset h' \Longrightarrow gIh) \tag{5}$$

$$g \nearrow m \iff \neg gIm \wedge (m' \subset h' \Longrightarrow gIn) \tag{6}$$

$$(H, N, I \cap (H \times N)) \tag{7}$$
$$\forall h \in H[h \nearrow m \Longrightarrow m \in N]$$
$$\forall n \in N[g \swarrow n \Longrightarrow g \in H]$$

4. 関連度を用いた分解

本章では，提案手法について説明する．提案手法は，分解手法の Nested Line Diagram を応用した手法であり，以下の点で，従来の概念束の簡素化や分解手法より優れている．

- 代表的な簡素化手法のように，データの一部の情報を無視するようなことはない．
- ほとんどの分解手法のような制約がなく，どのようなデータにおいても適用することができる．
- 関連性の高い属性をグループとした分解を自動で行えるので，データに関する専門知識を必要としない．

提案手法では，関連性の高い属性を見つけるために**関連度**を定義する．関連度から距離行列を作成し，階層型クラスタリングを行う．この結果から，属性のグループ分けを行い，Nested Line Diagram を扱って分解を行う．

4.1 関連度

属性どうしの関係性を明らかにするために，関連度 Rel を定義する．属性 x と属性 y の標本相関係数を $r(x, y)$ とするとき，関連度 $Rel(x, y)$ は式 (8) のように定める．

$$Rel(x, y) = |r(x, y)| \tag{8}$$

これにより，属性 x と属性 y が独立関係であるほど 0 に近い値になり，正もしくは負の相関が強いと 1 に近い値になる．また，すべての対象が持っている，もしくは持っていない属性があると，分母が 0 になってしまうので関連度を求めることができない．よって，本手法はそのような属性が存在する形式文脈を取り扱うことができないが，そのような属性は分析を行う必要性がないため，特に問題はないと考えられる．

4.2 クラスタリング

提案手法では上記で定義した関連度から距離行列を作成し，階層型クラスタリングを行うことで分解を行う．属性 x と y における距離 $Dist(x, y)$ は以下の式 (9) のようにして決まり，関連度が高いほど距離が小さい値になる．

$$Dist(x, y) = 1 - Rel(x, y) \tag{9}$$

すべての属性の組み合わせにおいて距離を算出し，距離行列に表してクラスタリングを行う．本手法では以下の 6 つの階層型クラスタリング手法を扱い，実験では，それらの手法の違いを明らかにして，どの手法が適しているのかに関する考察を述べる．

- 最短距離法
- 最長距離法
- 重心法
- メディアン法
- 群平均法
- ウォード法

4.3 分解までの流れ

本節では，提案手法の全体的なアルゴリズムの流れについて説明する．

まず，すべての属性同士の組合せにおいての関連度を算出し，行列の形で表す．次に，分解を行いたい概念束の元となる形式文脈において，完全に排他的な関係となっている属性を見つけ，その関連度を最大の値である 1 に変換する．この操作は，多値属性となる属性らが同じグループになるように分解を行うためである．さらに，クラスタリン

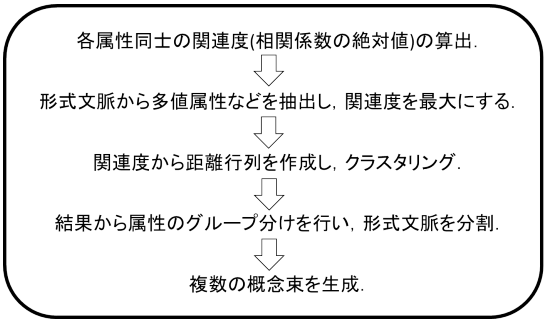


図 3 提案手法のアルゴリズムの流れ
Fig. 3 Alogrithm:Proposed Method

グを行うための距離行列を作成し，クラスタリングを行う．扱うクラスタリング手法は上記の 6 つの手法であり，分析の目的によって選ぶ必要がある．最後に，クラスタリングの結果から形式文脈を分割し，それぞれの概念束を生成することで概念束分解を行う．いくつの概念束に分解するかどうかは，分析者が自由に決めることができ，通常は分析者が理解しやすいサイズの概念束となるような個数とするのが好ましいと考える．

図 3 は上記のアルゴリズムの流れをまとめたものである．

5. 実験と結果

今回，実験は全部で 3 つ行った．1 つ目は完全にランダムで作成したデータに提案手法を適用し，分解した概念束の概念数が元の概念束の概念数とどう変化したか比較した．2 つ目は関連度が最大になる，もしくは最小となるような属性の組合せで構成された形式文脈に，一定の確率で反転するノイズを加えたデータにより，適切な分解が行われているか実験を行った．3 つ目では，概念束の和と積を用いて作成した概念束から提案手法で分解を行い，それらがどのような概念束に分解されるのか観察した．

5.1 ランダムデータでの実験

図 4, **図 5** は，ランダムデータによって分解後の 2 つの概念束における概念数の和，もしくは積を求めたグラフである．ランダムデータでは対象 50 個・属性 10 個の形式文脈を使用している．グラフの横軸はデータを作成する際の関係を結ぶ確率，縦軸が概念数の和・積の値の平均である．また，比較のため，分解前の概念束の概念数の平均もグラフに表している．

和の値は小さいほど分解後の概念束の構造の理解がしやすく，積の値が小さいほど元の概念束との非対応概念の数が少ない．和の値が元のデータの概念数より小さく，提案手法が概念束の構造の理解がしやすくなっていることが分かる．各クラスタリング手法を比較してみると，和の値と積の値はトレードオフの関係であると考えられる．これは分割した際に，属性数の偏りがしやすい，もしくは均等になりやすいクラスタリング手法の違いが出たと考えられる．本研究では，単純化を目的としているため，最長距離法・ウォード法・群平均法などを使用したほうが適していると考えられる．

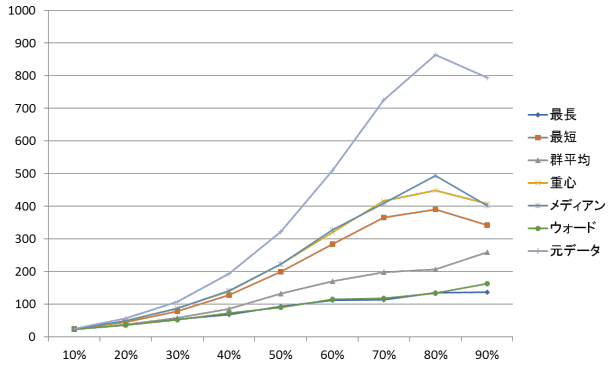


図 4 ランダムデータによる分解を行った概念数の和
Fig. 4 Sum of the Number of Concept

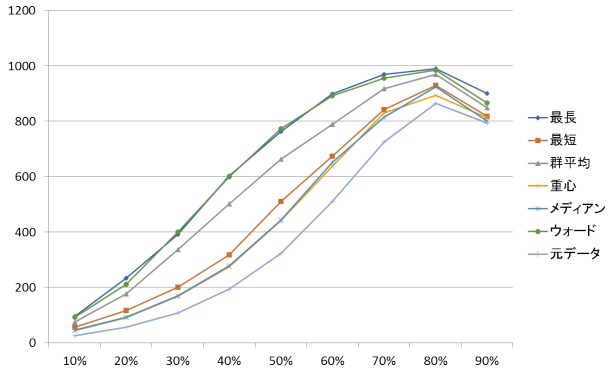


図 5 ランダムデータによる分解を行った概念数の積
Fig. 5 Product of the Number of Concept

5.2 ノイズを加えたデータでの実験

図 6 はノイズを変化させて 1000 回分解した場合の，適切な分解が行われていた割合のグラフである．横軸がノイズの生起確率，縦軸が割合である．テストデータは，対象 50 個・属性 8 個であり，属性 1〜4 と属性 5〜8 のように属性集合が分割されるように分解が行われるように設定した形式文脈である．この実験により，上記のような分解が行われるとき適切だと考え，どの程度の精度で分解が行われるか・クラスタリング手法ごとにどのような違いが出るか，検証する．

結果より，ウォード法・群平均法において，ノイズが 25 %あっても 60 %の割合で適切な分解がされており，高い精度で適切な分解が行われていることがわかる．逆に重心法・メディアン法においてかなり精度が低い結果になった．これは，前述の属性数の偏りが大きくなったために，ノイズの影響を強く出たと考えられる．これより，提案手法によって分解を行う際には，特にウォード法・群平均法が適していると考えられる．

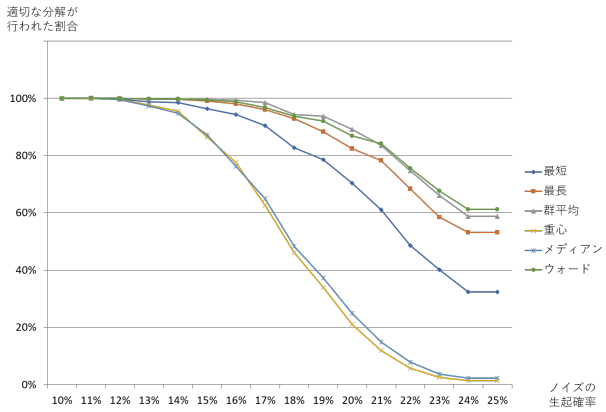


図 6 クラスタリング手法ごとの適切な分解の割合のグラフ
Fig. 6 Rate of Proper Decomposition

表 2 概念束 A と B の和で表される形式文脈
Table 2 FC:Sum of Concept Lattice A and B

	M_A	M_B
G_A	I_A	$\emptyset$
G_B	$\emptyset$	I_B

表 3 概念束 A と B の積で表される形式文脈
Table 3 FC:Product of Concept Lattice A and B

	M_A	M_B
G_A	I_A	$\times$
G_B	$\times$	I_B

5.3 概念束の和と積を扱ったデータでの実験

この実験では，概念束の和と積を扱った，複数のデータが入り混じったようなデータを使った実験を行う．ここでは，概念束の和は 2 つの概念束の Horizontal Sum，概念束の積は 2 つの概念束の Direct Product とする．これにより，提案手法が元の概念束を取り出すことができるか検証する．

概念束 A を構成する形式文脈 $\mathbb{K}_A = (G_A, M_A, I_A)$ と，概念束 B を構成する形式文脈 $\mathbb{K}_B = (G_B, M_B, I_B)$ の和 $\mathbb{K}_{A+B}$ と積 $\mathbb{K}_{A\times B}$ の形式文脈は以下の式 (10), (11) のようにして求める．ここで，$\mathbb{K}_A$ と $\mathbb{K}_B$ の対象集合・属性集合の共通部分はないものとする．

$$\mathbb{K}_{A+B} = (G_A \cup G_B, M_A \cup M_B, I_A \cup I_B) \tag{10}$$

$$\mathbb{K}_{A\times B} = (G_A \cup G_B, M_A \cup M_B, I_A \cup I_B \cup G_A \times M_B \cup G_B \times M_A) \tag{11}$$

つまり，概念束の和では**表 2** のような形式文脈，積は**表 3** のような形式文脈で生成される概念束となる．

図 7 は実験において，和と積を作成するための概念束とその形式文脈である．また，**表 4** は簡単な概念束の和，**表 5** は簡単な概念束の積によって作成されたデータで分解を行ったときの結果である．◯ は元の概念束が抽出された場合，△ は 2 つに分解ができるが元の概念束を抽出しない場合，× は 2 つに分解することが難しい (3 個以上に分解することが適切である) 場合である．

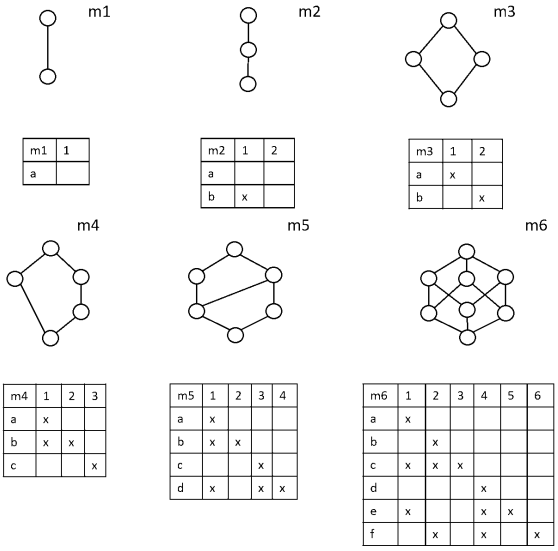


図 7 実験で扱うための概念束とその形式文脈 m1〜m6
Fig. 7 Concept Lattice used in Experiment

表 4 簡単な概念束の和での分解結果
Table 4 Decomposition used Sum of CL

	最長	最短	平均	重心	メディ	ウォー
m2+m2	×	×	×	×	×	×
m3+m3	×	×	×	×	×	×
m4+m4	×	×	△	△	△	×
m5+m5	×	◯	◯	◯	◯	△
m6+m6	×	◯	◯	△	△	◯
m3+m5	×	◯	△	△	△	△
m2+m6	×	◯	◯	◯	◯	△

表 5 簡単な概念束の積での分解結果
Table 5 Decomposition used Product of CL

	最長	最短	平均	重心	メディ	ウォー
m2×m2	◯	◯	◯	◯	◯	◯
m3×m3	×	◯	◯	◯	◯	◯
m4×m4	△	◯	◯	△	△	△
m5×m5	△	△	◯	△	△	◯
m6×m6	×	×	◯	△	×	◯
m3×m5	△	△	◯	△	△	◯
m1×m6	×	◯	◯	◯	△	△

簡単な概念束の和では，一部において，どのクラスタリング手法においても分解ができない場合があった．これは概念束のサイズが小さすぎるため，どの属性も独立になってしまい，関連度がすべて等しい値になったためだと考えられる．特に元の概念束を抽出する手法として最短距離法，次点で群平均法が挙げられた．

簡単な概念束の積では，和の場合よりも分解可能な場合が多かった．特に元の概念束を抽出する手法として群平均

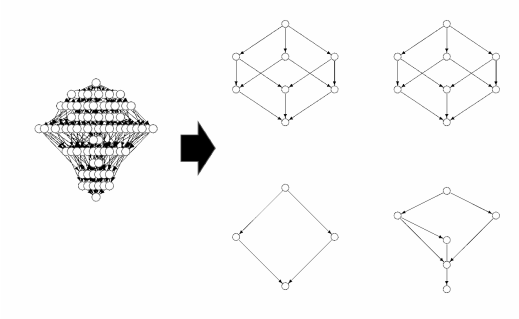


図 8 (m6×m6)+(m3×m5) の場合での 4 つに分解した概念束
Fig. 8 Decomposition (m6×m6)+(m3×m5)

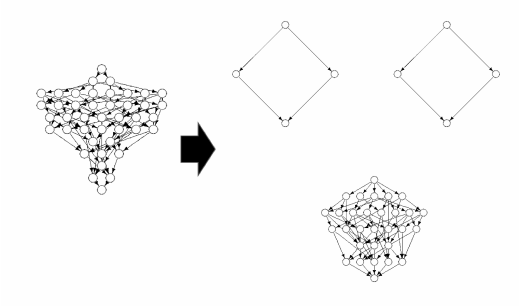


図 9 (m3+m5)×(m2+m6) の場合での 3 つに分解した概念束
Fig. 9 Decomposition (m3+m5)×(m2+m6)

法，次点でウォード法が挙げられた．

また，4 個の概念束の組合せから，より複雑な概念束を作成し，提案手法で分解を行った．

図 8 では，(m6×m6)+(m3×m5) で得られた概念束で 4 つの概念束に分解を行った場合の結果である．この場合では，全クラスタリング手法において同じ分解結果となり，すべての概念束が抽出された．また，2 つの概念束に分解した場合では，m6+m3 と m6+m5 に分解されてしまい，(m6×m6) と (m3×m5) のようには分解されなかった．これは，簡単な概念束の和と積の場合の結果のように，和よりも積のほうが分解がしやすいために，このような分解が行われたと考えられる．

図 9 は，(m3+m5)×(m2+m6) で得られた概念束をウォード法を用いて分解を行った結果である．この結果では，どのクラスタリング手法においても，元の概念束を抽出することができなかった．また，ウォード法においては，図のようにある程度均等に分解が行われていたが，他の手法では，極端に属性数の偏りがあるクラスタリング結果になり，サイズの大きい概念束が残ってしまう結果になった．これは，和と積で複雑な概念束を作成した場合，元の概念束のサイズの大きさに差があると，関係のない属性どうしで負の相関が大きくなってしまう場合がある．そのため，関連度が大きくなってしまい，別々の概念束の属性を一緒にした分解が行われたと考えられる．

6. おわりに

本研究では，構造の理解が難しい複雑な概念束を理解しやすくするために，Nested Line Diagram を改良した新しい分解手法を提案した．この手法は，簡素化手法のようなデータの情報欠損がなく，従来の分解のようなデータの制約や専門知識の必要性がないまま，概念束の単純化を行うことができる．実験では，提案手法が関連性の高い属性を取り出すことができ，概念束のサイズを大幅に小さくすることができることが分かった．提案手法内で扱われるクラスタリング手法ごとの違いを明確にし，群平均法が総合的に扱いやすい手法であることがわかった．しかし，和と積が複雑に入り混じったような概念束ではうまく分解が行われない場合もあった．

今後の課題として，実際のデータにおいて，関連性のある属性を抽出することができるのか，複雑な概念束を単純なもの変換できているか評価する必要がある．また，従来手法である概念束の簡素化や分解手法と，本研究で提案した手法と比較し，特徴や違いを明確にする必要がある．

参考文献

[1] Wille, R. : Restructuring lattice theory : An approach based on hierarchies of concepts, Ordered Sets, D. Reidel Publishing, pp.445-470(1982).
[2] Belohlavek, R and Macko, J.: Selecting Important Concepts Using Weights, ICFCA 2011: Formal Concept Analysis pp.65-80, (2011).
[3] Dias, SM and Vieira, NJ. : Concept lattices reduction: Definition, analysis and classification, Expert Systems with Applications 42 (20), pp.7084-7097,(2015).
[4] Kuznetsov, S. O. and Makhalova, T. P. : Concept interestingness measure: a comparative study, in Proceedings of CLA 2015, pp. 59-72 (2015)
[5] Priss, U and Old, L. J. : Data Weeding Techniques Applied to Roget's Thesaurus, KPP 2007, KONT 2007: Knowledge Processing and Data Analysis, pp.150-163, (2007)
[6] Kuznetsov, S. O. : On stability of a formal concept, Ann. Math. Artif. Intell. 49., pp.1–4, (2007)
[7] Kuznetsov, S. O. , Obiedkov, S and Roth, C. : Reducing the representation complexity of lattice-based taxonomies, Conceptual Structures : Knowledge Architectures for Smart Applications, pp.241–254, Springer, (2007)
[8] Ishigure, H. , Mutoh, A. , Matsui, T. , Inuzuka, N. : Concept Lattice Reduction Using Attribute Inference, GCCE2015, pp.108–111, (2015)
[9] Ganter, B and Wille, R. :Formal Concept Analysis: Mathematical Foundations, Springer Science & Business Media,(2012).
[10] Funk, P. , Lewien, A. , Snelting, G. , Braunschweig, T. :Algorithms for Concept Lattice Decomposition and their Application, Abteilung Softwaretechnologie., (1995)