

含意関係に注目した概念束分解手法の評価

石樽 隼人^{1,a)} 武藤 敦子¹ 森山 甲一¹ 犬塚 信博¹

受付日 2019年5月20日, 再受付日 2019年7月10日,
採録日 2019年8月26日

概要: 形式概念分析は概念を数学的に扱い, データ分析を行う手法である. 形式概念分析では, データから概念束と呼ばれる構造や含意関係と呼ばれる属性間の関係を用いて, データの分析を行う. しかしデータのサイズが増加すると, 概念束は急速に複雑になる. そのため, 概念束の分解を行う様々な手法が提案されているが, 分解手法の比較評価は十分ではない. 本稿では, 分解が可能かどうかや分解後の概念束が含意関係をどの程度保つか, 分解後の概念束から読み取れる含意関係が正しいかという観点から, 概念束の分解手法を比較評価した. また, 分解手法の1つである属性集合の分割に注目して, 元の情報を保ったまま概念束のサイズを小さくできるかに関して調査した.

キーワード: 形式概念分析, 概念束の分解, 含意関係

Evaluation of Decomposition Methods for Concept Lattices Based on Implications

HAYATO ISHIGURE^{1,a)} ATSUKO MUTOH¹ KOICHI MORIYAMA¹ NOBUHIRO INUZUKA¹

Received: May 20, 2019, Revised: July 10, 2019,
Accepted: August 26, 2019

Abstract: Formal concept analysis is a data analysis method using concepts mathematically. This method analyses data using a structure called a concept lattice and relations between attributes called implications. However, increase of data complicates a concept lattice very much. Because of this, various methods for concept lattice decomposition have been proposed but there is no sufficient comparative evaluation of them. We evaluated them from the perspective of decomposability, preservation of original implications and correctness of implications from decomposed lattices. We also confirmed whether a partition of the set of attributes, which is one of decomposition methods, is able to decrease the size of a concept lattice while preserving the original information.

Keywords: formal concept analysis, concept lattice decomposition, implication

1. はじめに

データマイニングに関する手法の1つに形式概念分析 [1] がある. これは束論の応用として, 概念やその構造を数理的に扱うことでデータ分析を行う手法である. 形式概念分析で扱われる概念は形式概念, 形式概念の構造は概念束と呼ばれる. 概念束を図示し, その構造を観察することで, データの構造や属性間の関係等を理解することができる. 形式概念分析は, 画像データベースの構造の可視化 [2] や

ソースコードの分析 [3] 等で利用されている.

一方でデータの増大に従い形式概念の数が急速に大きくなり, 概念束の構造が複雑になることが, 形式概念分析の問題として知られている. 概念束が大きく複雑になると, データの理解は困難となる. このため, 概念束の簡素化や分解が検討されてきた.

概念束の簡素化は, 大きく複雑な概念束を単純化し, 小さくする手法である. 一方で概念束の分解は, 大きく複雑な概念束を複数に分けることで扱いやすくする手法である.

概念束の分解はプログラムの分析 [4], [5], [6] やコミュニティ検出 [7], システムの権限設定 [8] で利用されている. 様々な分解手法が提案されているが, 利用可能とする前提

¹ 名古屋工業大学
Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan

^{a)} h.ishigure.488@nitech.jp

を必要とすることが多く、つねに分解可能とは限らない。他方、利点は数学的な優れた特徴である。多くの分解手法は代数的な性質を基に定義されているため、分解後の概念束と分解前の概念束との関係が強い。そのため、簡素化と比較すると、概念束の分解は適用後の概念束から適用前の概念束の情報が多く得られると考えられる。そこで本稿では、様々な分解手法について、それが利用可能となる条件と、利用可能である場合に得られる情報について明らかにし、またいくつかのデータを用いて実験的に評価を行った。

分解について比較・検討するいくつかの研究がある。Prissら [9] は、概念束の分解を含む多くの手法について分類を行った。Viaudら [10] は、束上の二項関係である complete congruence relation に基づく分解手法について、分解前の概念束と分解後の概念束の性質を比較した。深谷ら [11] は、概念束を図示する手法である nested line diagram に用いられる分解について評価を行った。これらの研究では、異なる分解手法間の比較は行われていない。また、Funkら [12] は、3種類の分解手法のアルゴリズムを紹介し、適用例を示したが、分解手法間の違いについては、計算量以外は明確に述べていない。

本稿は様々な分解手法を統一的に記述した後、理論的に評価・考察し、また実験による評価を示す。2章では形式概念分析の基礎概念を準備し、また図から読み取れる情報について明確化する。3章では本稿で評価する概念束の分解手法について説明する。そこで分解手法を統一的に述べるため、オリジナルな定義と等価な再定義を与える。4章では、分解後の概念束から分解前の概念束に関して得られる情報について、理論的な評価を行う。5章では、分解可能性と分解後に得られる情報について、いくつかのデータを用いて実験的に評価する。また、特に分解手法の1つである属性集合の分割に注目して、実験を加える。最後に、6章では結論を述べる。なお本稿は、文献 [13] を基礎として新たな考察と実験的評価を加えたものである。

2. 形式概念分析

2.1~2.3 節では文献 [14] に従って、形式概念分析の基礎概念を準備する。また、2.4 節において、図表現から形式概念の内包や含意関係を読み取ることができるということについて明確化する。

2.1 概念束

形式概念分析では、形式文脈と呼ばれるデータから形式概念と概念束を得ることを基礎として、データ分析を行う。以降では、集合はすべて有限集合であると仮定する。形式文脈 (G, M, I) は、対象集合 G 、属性集合 M 、 G と M の間の二項関係 $I \subseteq G \times M$ から構成される。対象 $g \in G$ と属性 $m \in M$ に対して、 $(g, m) \in I$ であるとき、 g は m を持つといい、 gIm と書く。表 1 は形式文脈の例である。表

表 1 形式文脈

Table 1 Formal context.

	卵生 (m_1)	母乳 (m_2)	言葉 (m_3)
ハト (g_1)	×		
カモノハシ (g_2)	×	×	
ネコ (g_3)		×	
ヒト (g_4)		×	×

中の × は対象が属性を持つことを表す。

ここで、対象の集合 $X \subseteq G$ と属性の集合 $Y \subseteq M$ に対して、式 (1)、(2) のとおり写像を定義する。

$$X \mapsto X^I := \{m \in M \mid \forall g \in X [gIm]\} \quad (1)$$

$$Y \mapsto Y^I := \{g \in G \mid \forall m \in Y [gIm]\} \quad (2)$$

集合のサイズが 1、すなわち $X = \{g\}$ や $Y = \{m\}$ のような場合には、 X^I, Y^I を g^I, m^I と書くことがある。また、 $(X^I)^I$ を X^{II} と書く。このとき、形式概念と概念束が以下のように定義される。

定義 1 (形式概念)。形式文脈 (G, M, I) および対象の集合 $A \subseteq G$ と属性の集合 $B \subseteq M$ に対して、組 (A, B) が $A^I = B, B^I = A$ をともに満たすとき、これを (G, M, I) の形式概念という。このとき、 A, B をそれぞれ外延、内包と呼ぶ。また、 (G, M, I) の形式概念すべての集合を $\mathfrak{B}(G, M, I)$ と書く。

定義 2 (概念束)。形式文脈 (G, M, I) の 2 つの形式概念 $(A_i, B_i), (A_j, B_j)$ に対して、以下の半順序を定義する。

$$(A_i, B_i) \leq (A_j, B_j) \iff A_i \subseteq A_j \iff B_i \supseteq B_j$$

この順序により $(\mathfrak{B}(G, M, I), \leq)$ は束をなす。これを概念束と呼ぶ。

以降は、 $\mathfrak{B}(G, M, I)$ と $(\mathfrak{B}(G, M, I), \leq)$ を区別しない。また、概念束において、上限、下限は以下ようになる。

定理 1 (文献 [14])。概念束 $\mathfrak{B}(G, M, I)$ において、下限、上限は次式のように定まる。

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)^{II} \right)$$

$$\bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)^{II}, \bigcap_{t \in T} B_t \right)$$

ここで、 T はすべての形式概念の集合につけたインデックスの集合の任意の部分集合である。

図 1 は表 1 で表される形式文脈の概念束を、ハッセ図で図示したものである。1 つのノードは 1 つの形式概念を表す。属性 m のラベルは形式概念 (m^I, m^{II}) に付与される。以降ではこの形式概念を $\mu_I m$ と書く。形式概念 $\mu_I m$ は m を含む形式概念の中で、最上位の形式概念である。たとえば、図 1 の概念束において、 $\mu_I m_3 = (\{g_4\}, \{m_2, m_3\})$ である。対象の数は属性の数に比べて膨大である場合が多

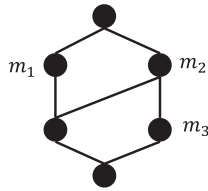


図 1 表 1 の概念束

Fig. 1 Concept lattice of Table 1.

表 2 reduced でない形式文脈
Table 2 Non-reduced context.

	m_1	m_2	m_3	m_4
g_1	×	×		×
g_2		×	×	×
g_3				×
g_4		×		×
g_5		×	×	×

表 3 Reduced context
Table 3 Reduced context.

	m_1	m_2	m_3
g_1	×	×	
g_2		×	×
g_3			

いため、対象のラベルは省略することが多い。このとき、形式概念の内包は、それ自身かそれより上位の形式概念に与えられたラベルで示される。たとえば、図 1 において、 m_3 のラベルが付いた形式概念の内包は、 $\{m_2, m_3\}$ である。また、最大元の内包は \emptyset である。

2.2 対象・属性の冗長性

概念束は与えられた形式文脈に対して一意に定義されるが、概念束を見やすくするため、概念束の構造を変化させない範囲で一部の対象や属性を取り除くことがある。取り除いても概念束の構造を変化させない対象や属性は冗長と呼ばれる。冗長な対象や属性を取り除いた形式文脈として、reduced context が知られている。

reduced context は形式文脈から冗長な対象、属性をすべて取り除いた形式文脈である。reduced context は概念束の構造を変化させない範囲で最小の形式文脈となる。

定義 3 (reduced context). 形式文脈 (G, M, I) について、その概念束 $\mathfrak{B}(G, M, I)$ が任意の対象 $g \in G$ に対して $\mathfrak{B}(G \setminus \{g\}, M, I \cap ((G \setminus \{g\}) \times M))$ と同型でなく、かつ任意の属性 $m \in M$ に対して $\mathfrak{B}(G, M \setminus \{m\}, I \cap (G \times (M \setminus \{m\})))$ と同型でないとき、 (G, M, I) を reduced context と呼ぶ。

表 2 は reduced でない形式文脈を、表 3 は表 2 に対応する reduced context を表す。reduced context では、まったく同じ属性を持つ対象は 1 つにまとめられている (g_2 と g_5)。また、すべての対象が持つ属性 (m_4) は取り除かれ

ている。ほかに、reduced でない形式文脈で、 g_4 が持つ属性の集合は g_1 と g_2 が持つ属性の集合の積集合で表されるが、 g_4 のような対象は reduced context では取り除かれる。

reduced context を作成するための手段としては、arrow relation [14] が知られている。

定義 4 (arrow relation). 形式文脈 (G, M, I) で、 G と M の間の二項関係 arrow relation \swarrow, \nearrow は次式で定義される。

$$g \swarrow m \Leftrightarrow \begin{cases} (g, m) \notin I \text{ and} \\ \forall h \in G [g' \subseteq h' \text{ and } g' \neq h' \Rightarrow hIm] \end{cases}$$

$$g \nearrow m \Leftrightarrow \begin{cases} (g, m) \notin I \text{ and} \\ \forall n \in M [m' \subseteq n' \text{ and } m' \neq n' \Rightarrow gIn] \end{cases}$$

形式文脈の reduced context は、まずまったく同じ属性を持つ対象、まったく同じ対象に持たれる属性を取り除き、その後、 $g \swarrow m$ である属性 m が存在しない対象 g と、 $g \nearrow m$ である対象 g が存在しない属性 m を除くことで得られる。

2.3 属性間の含意関係

形式概念分析では属性間の含意関係から、データに対しての考察を得ることが多い。属性の含意関係 $P \rightarrow Q$ は、任意の対象が P のすべての属性を持つならば、 Q のすべての属性も持つことを表す。これは次のように定義される。

定義 5 (属性間の含意関係). (G, M, I) を形式文脈とする。属性の集合 $P, Q \subseteq M$ が、任意の対象 $g \in G$ に対して $P \not\subseteq g^I$ または $Q \subseteq g^I$ を満たすとき、組 (P, Q) を (G, M, I) の含意関係と呼び、これを $P \rightarrow Q$ と書く。また、概念束 $L = \mathfrak{B}(G, M, I)$ について、 (G, M, I) の含意関係すべてからなる集合を $\mathfrak{L}(L)$ と書く。

含意関係は形式文脈から定義されるが、概念束からも得られる。実際、 $P \rightarrow Q$ が成立することと、すべての形式概念の内包が P を含むならば Q も含むことは同値である。

たとえば、表 1 の形式文脈では、含意関係 $\{m_3\} \rightarrow \{m_2\}$ が成立する (以下、 $\{ \}$ を省略し $m_3 \rightarrow m_2$ と書く)。

ある形式文脈で成立するすべての含意関係の集合 $\mathfrak{L}(L)$ は大きなサイズになるため、それと同等な部分集合である基底を扱うことが多い。すなわち、基底とは $\mathfrak{L}(L)$ のすべての含意関係を導出することができる $\mathfrak{L}(L)$ の部分集合である。基底の 1 つとして、Duquenne-Guigues 基底 [15] がある。これは基底のサイズが最小になることが保証され、よく利用される。本稿では評価に用いる。

2.4 ハッセ図からの読み取り

ここでは、概念束を表すハッセ図からどのような情報が読み取れるかを考える。ハッセ図には、属性に関するラベルのみが付与されていると仮定する。概念束の図中にあるノード n が存在し、 n とそれより上位のノードのラベルの

集合 $B \subseteq M$ は n の内包である。このことから、図から内包 $B \subseteq M$ を読み取れるということとする。したがって、 $\mathfrak{B}(G, M, I)$ に含まれる任意の内包を、図から読み取れる。

次に、図から含意関係を読み取るについて考える。含意関係に関しては、属性の集合 $P, Q \subseteq M$ に対して、式 (3) が成立する。

$$P \rightarrow Q \in \mathfrak{L}(L) \Leftrightarrow \bigwedge_{m \in P} \mu_I m \leq \bigwedge_{m \in Q} \mu_I m \quad (3)$$

すなわち P に含まれる属性のラベルがある形式概念すべての下限が、 Q に含まれる属性のラベルがある形式概念すべての下限より下にあるとき、 $P \rightarrow Q$ が成立する。これが成立することを、本稿では、図から含意関係 $P \rightarrow Q$ を読み取れるということとする。図 1 において、 m_1 のラベルが付いた形式概念と m_3 のラベルが付いた形式概念の下限は最小元であり、これは m_2 のラベルが付いた形式概念より下位であるため、 $m_1, m_3 \rightarrow m_2$ は図から読み取れる。形式文脈で成立する任意の含意関係は、図から読み取れる。

3. 概念束の分解

形式文脈のサイズが大きくなると、概念束は急速に大きく複雑になる。これに対処するため、概念束を分解する様々な手法がある。本稿では、horizontal decomposition [16], atlas decomposition [14], 属性集合の分割, subdirect decomposition [14] を説明し考察を補う。atlas decomposition と subdirect decomposition は本稿による再定義であり、属性集合の分割は、通常分解と見なしていない手法を本稿で分解として扱うものである。

分解手法にはほかに、vertical decomposition [17], reverse doubling construction [18], substitutional decomposition [14], subtensorial decomposition [14] がある。しかし、vertical decomposition はグラフ分解を応用した手法であり、代数的な性質を背景としていないため、分解前後の関係が不明確である。reverse doubling construction は subdirect decomposition と同じく complete congruence relation を用いているため、性質が類似している。substitutional decomposition と subtensorial decomposition は、与えられた概念束に対して分解を行う効率的な手法が知られておらず、実用的ではない。これらの理由により、以上の手法は本稿での評価の対象としない。

3.1 horizontal decomposition

horizontal decomposition は、概念束を横に並べるように分解する手法である。具体的には、概念束の最大元と最小元を取り除いて分割した後、それぞれに新たな最大元と最小元を加えることで、概念束の分解を行う。本稿では、新たに加える最大元と最小元は、分解前の概念束の最大元と最小元とする。

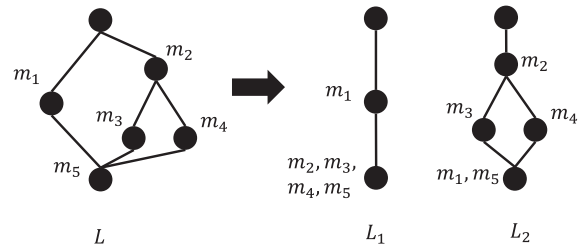


図 2 Horizontal decomposition
Fig. 2 Horizontal decomposition.

定義 6 (horizontal decomposition). 概念束 L に対して、 \top, \perp を L の最大元, 最小元とする。 $L \setminus \{\top, \perp\}$ のハッセ図をグラフと見なした場合の連結成分を X_1, X_2, \dots, X_n , $L_i = X_i \cup \{\top, \perp\}$ としたとき、 L_1, L_2, \dots, L_n のいずれもそのサイズが L のサイズより小さいならば、 L_1, L_2, \dots, L_n を L の horizontal decomposition と呼ぶ。

分解ではサイズが小さくなる必要があるため、 L から得られる連結成分が 2 つ以上、すなわち $n \geq 2$ である場合、そしてそのときに限り分解可能となる。分解不可能である場合に、事前に概念束に対して処理を行ったうえで分解を行う場合があるが、本稿では処理を行わない場合についてのみ評価を行う。

また、horizontal decomposition の性質について考察を行う。概念束 L の horizontal decomposition L_1, L_2, \dots, L_n は、定義より式 (4) の性質を満たす。

$$\bigcup_{i=1}^n L_i = L \quad (4)$$

また、分解後のそれぞれの束 L_i は元の概念束 L の完備部分束となる (束 L の部分集合 $S \subseteq L$ が完備部分束であるとは、任意の部分集合 $X \subseteq S$ に対して、 $\bigwedge X \in S$ かつ $\bigvee X \in S$ を満たすことをいう)。このとき、 $J_i = \bigcup_{(A,B) \in L_i} A \times B$ と書くと、分解後の束 L_i は形式文脈 (G, M, J_i) の概念束であり、したがって、図から内包と含意関係が読み取れる。

また、属性 $m \in M$ が $\mu_I m \in L_i$ である場合、明らかに $\mu_{J_i} m = \mu_I m$ である。 $\mu_I m \notin L_i$ である場合は、定義より m を含む L_i の形式概念は \perp のみである。したがって、この場合は $\mu_{J_i} m = \perp$ である。図 2 は horizontal decomposition の例を示す。この図では、 $\mu_{J_1} m_2 = \perp$ である。

3.2 atlas decomposition

atlas decomposition は、概念束の全体像と詳細を分けることで、概念束の構造を分かりやすくするための分解手法である。文献 [14] では束上の二項関係を用いて定義しているが、本稿ではそれと同等な block relation [14] を用いて再定義する。

定義 7 (block relation). 形式文脈 (G, M, I) の block relation $J \subseteq G \times M$ は以下の性質を満たす二項関係 J である。

表 4 表 1 の block relation の例
Table 4 A block relation of Table 1.

	m_1	m_2	m_3
g_1	×	×	×
g_2	×	×	×
g_3		×	×
g_4		×	×

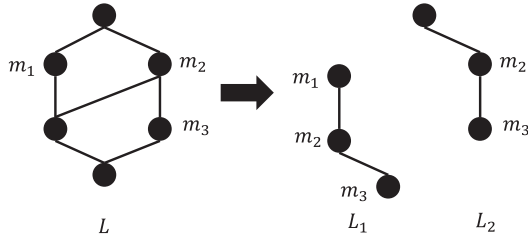


図 3 Atlas decomposition
Fig. 3 Atlas decomposition.

- $I \subseteq J$
- 任意の対象 $g \in G$ に対して g^J は (G, M, I) のある内包
- 任意の属性 $m \in M$ に対して m^J は (G, M, I) のある外延

また、 (G, M, I) と $(G, M, G \times M)$ を自明な block relation と呼ぶ。

表 4 は表 1 で表される形式文脈の block relation の例である。block relation J に対して (G, M, J) の形式概念 (A, B) を 1 つ考えると、 (A, B) の部分のみの束 $\mathfrak{B}(A, B, I \cap (A \times B))$ をブロックと呼ぶ。ブロックは、つねに $\mathfrak{B}(G, M, I)$ において、2 つの形式概念 $(A, A^I), (B^I, B)$ に挟まれた形式概念の集合となる。

定義 8 (atlas decomposition). 概念束 $L = \mathfrak{B}(G, M, I)$ に対して、 (G, M, I) のある自明でない block relation J のすべてのブロックを L_1, L_2, \dots, L_n とするとき、これを L の atlas decomposition と呼ぶ。また、 $\mathfrak{B}(G, M, J)$ を factor lattice と呼ぶ。

factor lattice は元の概念束の大まかな全体像を表し、ブロックは概念束の一部に関しての詳細を表す。図 3 は、表 4 の block relation による atlas decomposition を示す。

block relation の性質から、概念束 L の atlas decomposition L_1, L_2, \dots, L_n は式 (5) を満たすことが分かる。

$$\bigcup_{i=1}^n L_i = L \tag{5}$$

ある形式文脈に対して、block relation すべてからなる集合は閉包系となることが知られている。そのため、next closure algorithm [19] により block relation すべてを列挙できる。これによって自明でない block relation の有無を確認し、atlas decomposition の分解可能性を判定できる。

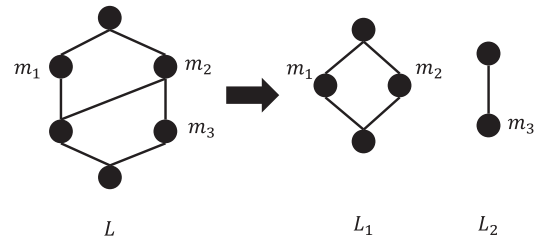


図 4 属性集合の分割
Fig. 4 Partition of the set of attributes.

3.3 属性集合の分割

概念束の図示の手法の一つである nested line diagram では、属性集合を複数に分割した後、それぞれ得られる形式文脈から概念束を求め、これらを組み合わせることで、概念束を図示する。本稿では、概念束 $L = \mathfrak{B}(G, M, I)$ と M の分割 M_1, M_2, \dots, M_n に対して、 $L_i = \mathfrak{B}(G, M_i, I \cap (G \times M_i))$ としたとき、 L_1, L_2, \dots, L_n のサイズが L より小さいならば、 L_1, L_2, \dots, L_n を L の分解として扱う。

属性集合の分割は次の性質を満たす。

定理 2 (文献 [14]). (G, M, I) を形式文脈、 M_1, M_2, \dots, M_n を M の分割とする。このとき、任意の $i = 1, 2, \dots, n$ に対して、 $J_i = I \cap (G \times M_i)$ とおくと、次の写像は全射である。

$$\begin{aligned} \mathfrak{B}(G, M, I) &\rightarrow \mathfrak{B}(G, M_i, J_i) \\ (A, B) &\mapsto ((B \cap M_i)^{J_i}, B \cap M_i) \end{aligned}$$

また、この写像について、元の概念束から分解後の概念束の直積への写像を考えると、これは単射となる。そのため、分解前の形式概念は分解後の形式概念の組合せで一意に表現できる。

図 4 は属性集合の分割の例である。ここでは、属性集合を $\{m_1, m_2\}, \{m_3\}$ の 2 つに分割している。

概念束が分解可能であることと概念束のサイズが 3 以上であることは同値である。属性集合 M を $|M|$ 個に分割したとき、分解後の概念束のサイズはすべて 2 以下である。したがって L のサイズが 3 以上ならば、 L は分解可能である。また、 L のサイズが 2 ならばある属性を持たない対象が存在するため、どのように分割しても分解後のいずれかの概念束のサイズが 2 となる。 L のサイズが 1 である場合は、明らかに分解不可能である。

属性集合の分割を行う際は、分析者が事前知識や直感に基づき分割を行うことが多いが、分割を自動で定める手法 [11] も提案されている。この手法では、属性間の距離を相関係数を用いて計算した後、階層的クラスタリングを行い、得られたクラスタを分割として利用する。

3.4 subdirect decomposition

ここでは subdirect decomposition を文献 [14] と異なり、compatible subcontext を用いて再定義する。ある形式文脈に対して、次のように compatible subcontext が定義さ

れる。

定義 9 (compatible subcontext). (G, M, I) を形式文脈, $(H, N, I \cap (H \times N))$ をその subcontext (すなわち $H \subseteq G, N \subseteq M$) とする. (G, M, I) の任意の形式概念 (A, B) に対して, $(A \cap H, B \cap N)$ が $(H, N, I \cap (H \times N))$ の形式概念であるとき, $(H, N, I \cap (H \times N))$ を compatible と呼ぶ.

compatible subcontext $(H, N, I \cap (H \times N))$ に対して, 式 (6) で定義される写像 $\Pi_{H,N} : \mathfrak{B}(G, M, I) \rightarrow \mathfrak{B}(H, N, I \cap (H \times N))$ は全射である.

$$\Pi_{H,N}(A, B) := (A \cap H, B \cap N) \quad (6)$$

また, compatible subcontext に対して, 概念束上の同値関係である complete congruence relation が定義される.

定義 10 (complete congruence relation). 形式文脈 (G, M, I) の compatible subcontext $(H, N, I \cap (H \times N))$ に対して, 次式のように定義される概念束 $\mathfrak{B}(G, M, I)$ 上の同値関係 θ を $(H, N, I \cap (H \times N))$ から導出される complete congruence relation と呼ぶ.

$$(A_1, B_1)\theta(A_2, B_2) \Leftrightarrow B_1 \cap N = B_2 \cap N$$

このとき, 概念束 $\mathfrak{B}(H, N, I \cap (H \times N))$ を factor lattice と呼ぶ.

complete congruence relation 上の同値類は, factor lattice に含まれる形式概念に対応する.

以上の定義を用いて, 概念束の subdirect decomposition が定義される. subdirect decomposition では, 属性集合の分割と同様に, 分解前の概念束の形式概念 1 つが, 分解後の形式概念の組合せで一意に表現される.

定義 11 (subdirect decomposition). 概念束 $L = \mathfrak{B}(G, M, I)$ に対して, $(H_i, N_i, I \cap (H_i \times N_i)), i = 1, 2, \dots, n$ を (G, M, I) の n 個の compatible subcontext とする. また, $L_i = \mathfrak{B}(H_i, N_i, I \cap (H_i \times N_i)), i = 1, 2, \dots, n$ とする. このとき, 各 compatible subcontext から導出される complete congruence relation $\theta_i, i = 1, 2, \dots, n$ について, 以下の条件が成立するならば, L_1, L_2, \dots, L_n を L の subdirect decomposition と呼ぶ.

$$\begin{cases} \bigcap_{i=1}^n \theta_i = \{(x, x) \mid x \in L\} \\ \forall i \in \{1, 2, \dots, n\} [\theta_i \neq \{(x, x) \mid x \in L\}] \end{cases}$$

図 5 は概念束上の 2 つの complete congruence relation を表す. 図中で色が同じ形式概念どうしは同値である. また, 図 6 は, 図 5 による subdirect decomposition を示す. subdirect decomposition では, 元の概念束に含まれる属性が, 分解後の概念束には現れない場合がある. この例では, 属性 m_4 は, 分解後の 2 つの概念束のどちらにも入ることがない.

subdirect decomposition の分解可能性の判定について

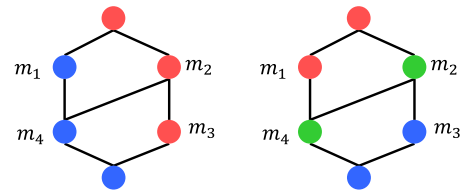


図 5 Complete congruence relation
Fig. 5 Complete congruence relations.

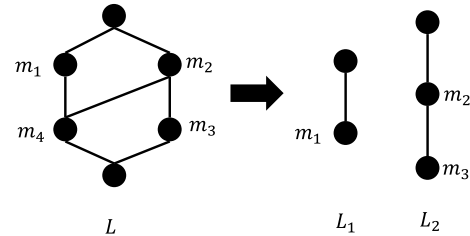


図 6 Subdirect decomposition
Fig. 6 Subdirect decomposition.

は arrow-closed subcontext による判定手法 [14] が知られている.

4. 理論的評価

本章では, 分解後の概念束の図から分解前の概念束の内包, 含意関係が読み取れるか, また条件を付けて読み取るべきものであるかについて, 分解手法の理論的な評価を行う. これは, 分解後の概念束を 1 つのみ用いた場合と分解後の概念束をすべて用いた場合について検討する.

以下, 式 (7) ~ (21) に性質をまとめる. 式 (21) は既知の結果である. 式 (7), (11), (13), (16), (18), (20) は定義や他の式等から容易に導出できるため簡単な付記のみとする. ほかは本稿で導いたものであり, 証明を与える.

以下では, 分解前の概念束を L , 分解後の概念束を L_1, L_2, \dots, L_n , L_i に含まれる属性の集合を M_i と書く. また, 概念束 L に含まれる内包, すなわち, 読み取ることのできる内包の集合を $\text{in}(L)$ と書く.

4.1 horizontal decomposition

分解後のある概念束 L_i と分解前の概念束について, 式 (7) が定義から導ける.

$$B \in \text{in}(L_i) \Rightarrow B \in \text{in}(L) \quad (7)$$

すなわち, L_i の図から読み取れる任意の内包は, 分解前の概念束の内包である.

含意関係については, 以下の関係 (8)~(10) が証明できる. 式 (8) は, 分解前の概念束の含意関係が必ず, 分解後の各概念束の図から読み取れることを表す. 式 (9) と式 (10) は, 分解後の概念束の図から読み取れる含意関係がある条件の下で正しいことを表す.

$$P \rightarrow Q \in \mathfrak{L}(L) \Rightarrow P \rightarrow Q \in \mathfrak{L}(L_i) \quad (8)$$

Proof. $P \rightarrow Q \in \mathfrak{L}(L)$ と仮定すると、任意の形式概念 $(A, B) \in L$ について、 $B \subseteq P \Rightarrow B \subseteq Q$ である。また、式 (4) より、任意の i について $L_i \subseteq L$ 。したがって、 L_i の任意の形式概念 (A, B) について、 $B \subseteq P \Rightarrow B \subseteq Q$ である。また、 $M_i = M$ より、 $P \cup Q \in M_i$ である。以上より、 $P \rightarrow Q \in \mathfrak{L}(L_i)$ が成立する。 \square

$P \rightarrow Q \in \mathfrak{L}(L_i)$ and

$$\forall m \in P [\mu_{J_i} m \neq \perp] \Rightarrow P \rightarrow Q \in \mathfrak{L}(L) \quad (9)$$

ここで、 $J_i = \bigcup_{(A,B) \in L_i} A \times B$ である。

Proof. 左辺が成立すると仮定する。このとき、任意の $m \in P$ について、 $\mu_I m = \mu_{J_i} m$ であり、かつ L_i は L の完備部分束であるため、 $\bigwedge_{m \in P} \mu_I m = \bigwedge_{m \in P} \mu_{J_i} m$ となる。また、任意の $m \in Q$ について、 $\mu_{J_i} m$ は $\mu_I m$ または \perp と一致するため、 $\bigwedge_{m \in P} \mu_{J_i} m$ は $\bigwedge_{m \in P} \mu_I m$ または \perp と一致する。すなわち、 $\bigwedge_{m \in Q} \mu_{J_i} m \leq \bigwedge_{m \in Q} \mu_I m$ である。したがって、以下が成立する。

$$\bigwedge_{m \in P} \mu_I m = \bigwedge_{m \in P} \mu_{J_i} m \leq \bigwedge_{m \in Q} \mu_{J_i} m \leq \bigwedge_{m \in Q} \mu_I m$$

これは $P \rightarrow Q \in \mathfrak{L}(L)$ と同値である。 \square

$P \rightarrow Q \in \mathfrak{L}(L_i)$ and $\exists m \in P [\mu_{J_i} m = \perp]$ and

$$\exists m \in P [\mu_{J_i} m \neq \perp, \top] \Rightarrow P \rightarrow Q \in \mathfrak{L}(L) \quad (10)$$

ただし、 J_i は式 (9) と同じである。

Proof. 左辺が成立すると仮定する。ここで、 $m \in P$ を $\mu_{J_i} m \neq \perp, \top$ を満たす属性、 $n \in P$ を $\mu_{J_i} n = \perp$ を満たす属性とする。このとき、 $\mu_I n = \perp$ または $\mu_I n \notin L_i$ が成立する。 $\mu_I n = \perp$ である場合、明らかに $\bigwedge_{m \in P} \mu_I m \leq \bigwedge_{m \in Q} \mu_I m$ となる。また、 $\mu_I n \notin L_i$ である場合、horizontal decomposition の定義より、 $\mu_I m \wedge \mu_I n = \perp$ である。ここから、 $\mu_I m \wedge \mu_I n = \perp$ である。したがって、どちらの場合でも $\mu_I m \wedge \mu_I n = \perp$ が成立する。すなわち、 $\bigwedge_{m \in P} \mu_I m = \perp$ である。ここから右辺が導かれる。 \square

分解後の概念束すべてを用いた場合、分解前の内包と分解後の内包の関係は以下が成立することが定義から分かる。

$$\bigcup_{i=1}^n \text{in}(L_i) = \text{in}(L) \quad (11)$$

この式から、分解後のすべての概念束の図から読み取れる内包を利用することで、分解前の概念束の内包が復元できることが分かる。

分解前の含意関係と分解後の含意関係の間の関係は、式 (12) で表されることを確認した。これにより、分解後の概念束のそれぞれの図から読み取れる含意関係を用いて、分解前の概念束の図から読み取れる含意関係を復元できる。

$$\mathfrak{L}(L) = \bigcap_{i=1}^n \mathfrak{L}(L_i) \quad (12)$$

Proof. $P \rightarrow Q$ が成立することと、 $P \rightarrow Q$ が任意の内包と矛盾しないことは同値である。すなわち、分解前の概念束に対して以下が成立する。

$$P \rightarrow Q \in \mathfrak{L}(L) \Leftrightarrow \forall B \in \text{in}(L) [B \supseteq P \Rightarrow B \supseteq Q]$$

分解後の概念束 L_i に対しては、 $P \rightarrow Q \in \mathfrak{L}(L_i) \Rightarrow P \cup Q \subseteq M_i$ であるため、以下が成立する。

$$P \rightarrow Q \in \mathfrak{L}(L_i)$$

$$\Leftrightarrow \forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q] \text{ and } P \cup Q \subseteq M_i$$

また、 $\text{in}(L) = \bigcup_{i=1}^n \text{in}(L_i)$ かつ $M_i = M$ であるため、次のようになる。

$$\begin{aligned} P \rightarrow Q \in \bigcap_{i=1}^n \mathfrak{L}(L_i) \\ \Leftrightarrow \forall i \in \{1, 2, \dots, n\} [\forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]] \\ \Leftrightarrow \forall B \in \bigcup_{i=1}^n \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q] \\ \Leftrightarrow \forall B \in \text{in}(L) [B \supseteq P \Rightarrow B \supseteq Q] \\ \Leftrightarrow P \rightarrow Q \in \mathfrak{L}(L) \end{aligned}$$

以上より題意は示された。 \square

4.2 atlas decomposition

atlas decomposition においては、分解後のある概念束 L_i と分解前の概念束には、関係 (13) が式 (5) から導ける。

$$B \in \text{in}(L_i) \Rightarrow B \in \text{in}(L) \quad (13)$$

これは、分解後の概念束の図から読み取れる内包が必ず正しいことを表す。

次に含意関係に関しては、以下の関係 (14)、(15) が証明できる。式 (14) は、元の概念束の含意関係が、そのすべての属性が分解後の概念束に含まれるならば、分解後の概念束の図から読み取れることを表す。式 (15) は、分解後の概念束の図から読み取れる含意関係がある条件の下で正しいことを表す。

$$P \rightarrow Q \in \mathfrak{L}(L) \text{ and } P \cup Q \subseteq M_i \Rightarrow P \rightarrow Q \in \mathfrak{L}(L_i) \quad (14)$$

Proof. 左辺を仮定する。任意の形式概念 $(A, B) \in L$ について、 $B \subseteq P \Rightarrow B \subseteq Q$ である。また、式 (5) より、任意の i について $L_i \subseteq L$ 。したがって、 L_i の任意の形式概念 (A, B) について、 $B \subseteq P \Rightarrow B \subseteq Q$ である。また、 $P \cup Q \in M_i$ である。以上より、 $P \rightarrow Q \in \mathfrak{L}(L_i)$ が成立する。 \square

$$P \rightarrow Q \in \mathfrak{L}(L_i) \text{ and } P \supseteq \{m \in M_i \mid \mu_J m = \top\}$$

$$\Rightarrow P \rightarrow Q \in \mathfrak{L}(L) \quad (15)$$

Proof. 左辺を仮定する. また, (H, M_i, J) を L_i に対応する形式文脈とする. 式 (3) より, $\bigwedge_{m \in P} \mu_{Jm} \leq \bigwedge_{m \in Q} \mu_{Jm}$ である. ここで, $P \supseteq \{m \in M_i \mid \mu_{Jm} = \top\}$ より, $\bigwedge_{m \in P} \mu_{Im} \leq \top$ である. また, $L_i \subseteq L$ であり, μ_{Im}, μ_{Jm} はそれぞれ L, L_i で m を含む最上位の形式概念であるため, $\mu_{Jm} \leq \mu_{Im}$ となる. したがって, $\perp \leq \bigwedge_{m \in P} \mu_{Jm} \leq \bigwedge_{m \in P} \mu_{Im} \leq \top$ であるため, $\bigwedge_{m \in P} \mu_{Jm} \in L_i$ である. しかし, $\bigwedge_{m \in P} \mu_{Jm}$ は L_i で P を含む最上位の形式概念である. よって, これと $\bigwedge_{m \in P} \mu_{Jm} \leq \bigwedge_{m \in P} \mu_{Im}$ より, $\bigwedge_{m \in P} \mu_{Jm} = \bigwedge_{m \in P} \mu_{Im}$ となる. 一方で, $\mu_{Jm} \leq \mu_{Im}$ より $\bigwedge_{m \in Q} \mu_{Jm} \leq \bigwedge_{m \in Q} \mu_{Im}$ である. 以上より, 次式が成立する.

$$\bigwedge_{m \in P} \mu_{Im} = \bigwedge_{m \in P} \mu_{Jm} \leq \bigwedge_{m \in Q} \mu_{Jm} \leq \bigwedge_{m \in Q} \mu_{Im}$$

これは右辺と同値である. □

また, 分解後の概念束すべてを用いた場合, 内包に関して式 (5) から次式 (16) が導ける.

$$\bigcup_{i=1}^n \text{in}(L_i) = \text{in}(L) \quad (16)$$

すなわち, 元の概念束の内包がすべて復元可能である. 本稿では, 元の概念束で成立する含意関係と分解後の各概念束で成立する含意関係の間に, 次式 (17) が成立することを新たに確認した. これにより, 分解後の概念束のそれぞれの図から読み取れる含意関係を用いて, 分解前の概念束の図から読み取れる含意関係を復元できる.

$$\mathfrak{L}(L) = \bigcap_{i=1}^n (\mathfrak{L}(L_i) \cup \{P \rightarrow Q \mid P \not\subseteq M_i\}) \quad (17)$$

Proof. 分解後の概念束 L_i に対して, 以下が成立する.

$$P \rightarrow Q \in \mathfrak{L}(L_i) \\ \Leftrightarrow \forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q] \text{ and } P \cup Q \subseteq M_i$$

ここで, $P, Q \subseteq M_i$ の場合, 明らかに以下が成立する.

$$P \rightarrow Q \in \mathfrak{L}(L_i) \Leftrightarrow \forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]$$

また, $P \not\subseteq M_i$ である場合, L_i の任意の内包 B に対して $B \subseteq M_i$ であるため, 以下が成立する.

$$\forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]$$

さらに $P \subseteq M_i, Q \not\subseteq M_i$ である場合, $M_i \in \text{in}(L_i)$ より以下が成立する.

$$\neg \forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]$$

以上より, 次式がつねに成立する.

$$P \rightarrow Q \in \mathfrak{L}(L_i) \text{ or } P \not\subseteq M_i \\ \Leftrightarrow \forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]$$

したがって, 次のようになる.

$$P \rightarrow Q \in \bigcap_{i=1}^n (\mathfrak{L}(L_i) \cup \{P \rightarrow Q \mid P \not\subseteq M_i\}) \\ \Leftrightarrow \forall i \in \{1, 2, \dots, n\} [P \rightarrow Q \in \mathfrak{L}(L_i) \text{ or } P \not\subseteq M_i] \\ \Leftrightarrow \forall i \in \{1, 2, \dots, n\} [\forall B \in \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q]] \\ \Leftrightarrow \forall B \in \bigcup_{i=1}^n \text{in}(L_i) [B \supseteq P \Rightarrow B \supseteq Q] \\ \Leftrightarrow \forall B \in \text{in}(L) [B \supseteq P \Rightarrow B \supseteq Q] \\ \Leftrightarrow P \rightarrow Q \in \mathfrak{L}(L)$$

以上より題意は示された. □

4.3 属性集合の分割

属性集合の分割では, これまで明確には性質について議論されていない. まず, 分解後の概念束1つに対して, 以下 (18) が定理 2 から導出できる.

$$B \in \text{in}(L_i) \Leftrightarrow \exists D \in \text{in}(L) [D \cap M_i = B] \quad (18)$$

これは, 分解後の概念束から読み取れる内包が分割された属性の範囲で正しく, かつ元の概念束の内包が同じ属性の範囲で分解後の概念束から読み取れることを表す. 次に含意関係に関しては, 次式 (19) が証明できる. これは, 分解後の概念束から読み取れる含意関係はすべて正しく, 元の概念束から読み取れる含意関係が分割された属性の範囲で, 分解後の概念束から読み取れることを表す.

$$P \rightarrow Q \in \mathfrak{L}(L) \text{ and } P \cup Q \subseteq M_i \Leftrightarrow P \rightarrow Q \in \mathfrak{L}(L_i) \quad (19)$$

Proof. まず, 左辺 \Rightarrow 右辺について考える. $J = I \cap G \times M_i$ とおくと, L_i に対応する形式文脈は (G, M_i, J) と書ける. 左辺を仮定すると, L の任意の内包 B について, $P \not\subseteq B$ または $Q \subseteq B$ である. また, $P \cup Q \subseteq M_i$ である. したがって, 式 (18) より, L_i の任意の内包についても, 同じことが成立する. すなわち, $P \rightarrow Q \in \mathfrak{L}(L)$ が成立する.

次に, 右辺 \Rightarrow 左辺について考える. 右辺を仮定すると, $P \rightarrow Q \in \mathfrak{L}(L_i)$ より, $P \cup Q \subseteq M_i$ である. また L_i の任意の内包 B について, $P \not\subseteq B$ または $Q \subseteq B$ である. このとき, 式 (18) より, L の任意の内包についても, 同じことが成立する. したがって, $P \rightarrow Q \in \mathfrak{L}(L)$ が成立する. □

また, 分解後の概念束すべてを用いた場合, 分割後の属性を横断する情報が分解後の概念束には存在しない. そのため, 分解後の概念束をすべて利用しても, それぞれの概念束から得られる以上の情報は得られず, 元の内包や含意関係を復元することはできない.

4.4 subdirect decomposition

subdirect decomposition について, 以下の式 (20) は

表 5 正しさと再現性

Table 5 Correctness and recall.

手法	内包		含意関係	
	正しさ	再現性	正しさ	再現性
horizontal	○	×	△	○
atlas	○	×	△	○
分割	△	△	○	×
subdirect	△	△	○	×

○は性質が保証されることを, △は条件付きで保証されることを, ×は保証されないことを表す.

式 (6) から得られる.

$$B \in \text{in}(L_i) \Leftrightarrow \exists D \in \text{in}(L)[D \cap M_i = B] \quad (20)$$

また式 (21) が成立することが知られている [10].

$$P \rightarrow Q \in \mathcal{L}(L) \text{ and } P \cup Q \subseteq M_i \Leftrightarrow P \rightarrow Q \in \mathcal{L}(L_i) \quad (21)$$

これらは, 属性集合の分割の性質と同等である.

subdirect decomposition に関しては, 新たに得られた成果は存在しない.

subdirect decomposition では, 元の属性のいずれかがどの分解後の概念束にも含まれない場合がある. 実際, 図 6 では分解後の概念束 L_1, L_2 のどちらにも属性 m_4 が含まれていない. このため, 分解後の概念束すべてを用いても, 元の内包と含意関係を復元することはできない.

4.5 理論的評価についてのまとめ

以上より, 各分解手法に対して, 分解後の概念束 1 つからどのような条件で含意関係が理解できるか, 分解後の概念束すべてを用いた場合にはどのように理解できるかについて明らかにした.

表 5 は, 各分解手法に対して, 内包および含意関係に関して, 分解後の概念束 1 つから読み取れるものが正しいか (正しさ), 分解後の概念束 1 つから分解前のものが必ず得られるか (再現性) を表す表である. horizontal decomposition と atlas decomposition では, 分解後に得られる内包は「正しさ」を有する. 一方で, 属性集合の分割と subdirect decomposition では, 分解後の概念束に含まれる属性の範囲でのみ, 分解後の内包が正しい. また, horizontal decomposition と atlas decomposition では, 分解後の概念束は内包の再現性を持たず, 属性集合の分割と subdirect decomposition は, 分解後に含まれる属性の範囲においてのみ再現性を持つ.

また, 含意関係について horizontal decomposition と atlas decomposition では, 分解後の含意関係は条件付きで正しい. 一方で, 属性集合の分割と subdirect decomposition では, 分解後の含意関係は必ず正しい. horizontal decomposition と atlas decomposition の条件の改良については,

表 6 一意性と復元可能性

Table 6 Uniqueness and reconstructability.

手法	分解の一意性	復元可能性
horizontal	○	○
atlas	×	○
分割	×	×
subdirect	×	×

○, × は表 5 と同じである.

研究の余地がある. また, horizontal decomposition と atlas decomposition では, 分解による含意関係の再現性を有する. 逆に, 属性集合の分割と subdirect decomposition は再現性を持たず, 分解後の概念束に含まれない属性に関する含意関係が得られない.

表 6 は, 各分解手法による分解が一意であるか, また分解後の概念束すべてから元のすべての内包・含意関係が復元可能かを表す表である. 分解の一意性は, 各分解手法の定義から分かる. horizontal decomposition は分解可能な場合には, 分解は 1 つに定まる. 他の手法は, 可能な分解が複数存在する可能性がある. 特に, 属性集合の分割は可能な分解の数が多し.

また, horizontal decomposition と atlas decomposition は, 分解後の概念束をすべて用いると, 元の内包・含意関係をすべて復元できる. 属性集合の分割と subdirect decomposition では, 一般に復元できない.

5. 実験的評価

5.1 実験 1

本稿では, 3.1~3.4 節で述べた 4 つの手法について, 概念束の分解可能性と分解後の概念束と元の性質の関係について, 実験的に評価する.

5.1.1 実験手法

(1) データ

実験ではデータとして, 文献 [20] で公開されているデータと Modula-2 プログラムに関する形式文脈 [4], [18] で例として利用された形式文脈, セキュリティに関する形式文脈 [8] を利用した. 文献 [20] の形式文脈はすべて離散化されているが, 二値ではない. そのため, one hot encoding によって形式文脈に変換した. また, すべての形式文脈は reduced context にしたうえで, 実験に用いた. reduced context にしても, 本稿で扱う分解手法の分解可能性は変化しない. 表 7 は, 実験で利用したデータを示す.

(2) 分解手法について

horizontal decomposition 以外の分解手法は, 分解が一意でないことがある. その場合分解後の概念束の数が可能な限り少なくなるように分解を行った.

また, 属性集合の分割では, 属性分割の方法を定める必要がある. 実験では, 文献 [11] で提案された手法を参考と

表 7 実験に利用したデータ
Table 7 Data used in experiments.

データ	対象数	属性数	形式概念数
Iris	42	17	164
Ecoli	105	32	813
Page blocks	72	43	996
Pima Indians	170	38	3,205
Glass	163	44	4,744
LED 7	326	24	7,039
Modula-2	12	10	15
[18]	9	10	23
セキュリティ	11	11	20

して階層的クラスタリングを利用し、そのときの距離として群平均法を用いた。

(3) 評価法

分解手法の評価には、分解後の含意関係の正しさと分解前の含意関係の再現性、また information content [21] を利用した。

分解後の概念束から読み取れる含意関係の正しさの評価では、分解後の各概念束 L_1, L_2, \dots, L_n から得られる Duquenne-Guigues 基底 $base_i$ の和集合 $\bigcup_{i=1}^n base_i$ に対して、次式のとおり含意関係が元の概念束 L でも成立する割合 $correct(\bigcup_{i=1}^n base_i)$ を評価した。

$$correct(\mathcal{J}) = \frac{|\mathcal{J} \cap \mathcal{L}(L)|}{|\mathcal{J}|} \quad (22)$$

ここで、分解後の概念束から得られる含意関係の集合を \mathcal{J} とする。この値が大きいほど、分解後の各概念束から得られる含意関係が正しいため、図から理解しやすいと考えられる。

分解前の概念束の図から読み取れる含意関係の再現性の評価では、分解前の概念束から得られる Duquenne-Guigues 基底 $base$ に対して、次式のとおりそれに含まれる含意関係が分解後のいずれかの概念束から得られる割合 $recall(base)$ を評価する。

$$recall(\mathcal{J}) = \frac{|\{P \rightarrow Q \in \mathcal{J} \mid \exists i [P \rightarrow Q \in \mathcal{L}(L_i)]\}|}{|\mathcal{J}|} \quad (23)$$

ここで、分解前の概念束から得られる含意関係の集合を \mathcal{J} とする。この値が大きいほど、分解前の含意関係が分解後の概念束の図から読み取れると考えられる。

information content による評価では、含意関係と整合する属性の集合を通して、分解後の概念束全体が維持する含意関係の程度を評価した。このとき、分解後の概念束全体から復元できる元の含意関係を利用した。具体的には、horizontal decomposition と atlas decomposition では元の概念束で成立する含意関係すべて、属性集合の分割と subdirect decomposition では分割後の各概念束の図から読み取れる含意関係の和集合となる。

表 8 含意関係の正しさ
Table 8 Correctness of implications.

データ	horizontal	atlas	分割	subdirect
Iris	—	—	1.000	—
Ecoli	—	0.998	1.000	—
Page blocks	—	0.996	1.000	—
Pima Indians	—	—	1.000	—
Glass	—	—	1.000	—
LED 7	—	—	1.000	—
Modula-2	0.167	0.000	1.000	—
[18]	—	0.654	1.000	1.000
セキュリティ	0.656	—	1.000	—

information content は含意関係の集合に対して、そのいずれかと整合しない属性の集合の割合を評価するものである。たとえば、 $m_1 \rightarrow m_2$ という含意関係に対して、 m_1 を含むが m_2 を含まない属性の集合は整合しない。結局 information content $cont$ は、次式で計算される。

$$cont(\mathcal{J}) = \frac{|\bigcup_{P \rightarrow Q \in \mathcal{J}} \{S \subseteq M \mid P \subseteq S \text{ and } Q \not\subseteq S\}|}{2^{|M|}} \quad (24)$$

$cont(\mathcal{J})$ が 0 に近いほど、 \mathcal{J} 内の含意関係の集合に整合する属性の集合が多く、1 に近いほど、整合するものは少なくなる。また、 \mathcal{J} の真部分集合 \mathcal{J}' をとると、 $cont(\mathcal{J}') \leq cont(\mathcal{J})$ となる。

このため、元の概念束で成立する含意関係の集合 $\mathcal{L}(L)$ に対する information content $cont(\mathcal{L}(L))$ と分解後も成立する含意関係の集合 \mathcal{J}_{dec} の information content $cont(\mathcal{J}_{dec})$ の比 $cont(\mathcal{J}_{dec})/cont(\mathcal{L}(L))$ を指標として用いる。この比が 0 に近いほど、分解後の含意関係と整合する属性の集合が元の含意関係に対して増加している。

5.1.2 結果と考察

表 8 は、分解後の概念束から読み取れる含意関係の正しさを示している。表中の — は分解ができなかったことを表す。属性集合の分割ではすべての場合に分解ができたが、それ以外の手法では多くの場合に分解ができなかった。また、属性集合の分割と subdirect decomposition では、表 5 の理論的評価と一致し、表 8 でも値が 1 となっている。一方で horizontal decomposition と atlas decomposition では、値が小さい場合があり、分解後の各概念束からの情報抽出には注意が必要であると考えられる。

表 9 は、分解後の概念束から読み取れる含意関係の再現性を示している。horizontal decomposition と atlas decomposition は、理論的評価と一致し、値が 1 となっている。一方で、属性集合の分割と subdirect decomposition の評価値は小さな値となっている場合がある。これらの手法では、元の含意関係が分解後の概念束からは分からない場合がある。

表 9 含意関係の再現性

Table 9 Recall of implications.

データ	horizontal	atlas	分割	subdirect
Iris	—	—	0.134	—
Ecoli	—	1.000	0.788	—
Page blocks	—	1.000	0.029	—
Pima Indians	—	—	0.148	—
Glass	—	—	0.900	—
LED 7	—	—	0.062	—
Modula-2	1.000	1.000	0.057	—
[18]	—	1.000	0.250	0.875
セキュリティ	1.000	—	0.200	—

表 10 information content による評価

Table 10 Evaluation by information content.

データ	horizontal	atlas	分割	subdirect
Iris	—	—	0.998	—
Ecoli	—	1.000	1.000	—
Page blocks	—	1.000	1.000	—
Pima Indians	—	—	1.000	—
Glass	—	—	1.000	—
LED 7	—	—	1.000	—
Modula-2	1.000	1.000	0.987	—
[18]	—	1.000	0.935	1.000
セキュリティ	1.000	—	0.985	—

表 10 は各分解手法で分解を行った際に得られた information content の比を示す。属性集合の分割以外の手法では、分解可能であった場合には含意関係と整合する属性の集合は増加していない。特に、horizontal decomposition と atlas decomposition では、含意関係を完全に維持するという理論的評価と一致する。また、属性集合の分割は元の含意関係すべてを保存はしていないが、含意関係と整合する属性の集合は増加していないことが分かる。

2つ以上の手法で分解が可能であり、元の概念束のサイズが比較的大きい Ecoli と Page blocks の場合について、分解後の概念束のサイズを調べた。Ecoli の場合では、atlas decomposition で分解された概念束のサイズは 781 と 32、属性集合の分割では 812 と 2 だった。また、Page blocks では、atlas decomposition で 659 と 337、属性集合の分割では 659 と 4 であった。atlas decomposition と属性集合の分割のどちらも、分解後の概念束は利用しやすいサイズとはなっていない。

以上から、属性集合の分割による手法以外では、利用できる場合が限られることが分かった。しかし、horizontal decomposition と atlas decomposition は、分解後の概念束から元のすべての情報が得られるため、分解可能であれば、利用を検討する余地がある。たとえば、horizontal decomposition は、属性集合が排他的な形で分割可能ならば分解しやすいため、プログラムの分析等で応用されてい

表 11 Ecoli データセットにおける分割数と information content

Table 11 Number of parts and information content.

分割数	概念束のサイズ	information content
1	813	1.000
2	812, 2	1.000
3	453, 6, 2	1.000
4	197, 7, 6, 2	1.000
5	89, 7, 7, 6, 2	1.000
6	56, 7, 7, 6, 6, 2	1.000
7	15, 10, 7, 7, 6, 6, 2	1.000
8	15, 10, 7, 7, 6, 4, 2, 2	1.000

る [4], [16]。また、これら 3つの手法は、属性集合の分割と異なり、対象と属性を対称的に扱うため、対象に対しても属性と同様の関係が成立することが期待される。そのため、対象と属性を同時に扱う場合には、これらの手法を検討すべきである。

一方で属性集合の分割は、実質的につねに概念束の分解が可能である。また、分解後の概念束からの含意関係の読み取りは容易である。このため、属性集合の分割は、利用しやすい手法であると考えられる。また、属性集合の分割は分解後の形式概念の組合せで元の概念束を表すため、horizontal decomposition や atlas decomposition より分解後の概念束のサイズが小さくなる可能性がある。しかし、実験では分解後の概念束のサイズは十分に小さくなっていない。実験では二分割の場合のみを扱ったが、分割数を増やすことでこの問題を解決できる可能性がある。これについて、次の実験 2 で調べる。

5.2 実験 2

5.2.1 実験手法

実験 1 において、属性集合の分割が分解可能であることが多く、扱いやすいことが分かったが、属性集合を二分割した場合には、分解後の概念束のサイズが十分に小さくなっていなかった。分割数を大きくすることで分解後の概念束のサイズは小さくなると考えられるが、一方で、分割数が大きい場合には元の含意関係が保存されなくなる恐れがある。そこで、本実験では、属性集合の分割において、分割数と information content がどのように変化するかを調べる。実験において、形式文脈として Ecoli データセットを用いる。

5.2.2 実験結果と考察

表 11 は分割数を 1 から 8 まで変化させた際の、分解後の概念束のサイズと information content の比である。多くの概念束に分解しても、整合しない属性の集合の減少はわずかであった (表 11 では評価値を四捨五入しているため、値の変化が見られない)。また、分割数が大きくなると、分解後の概念束それぞれのサイズは小さくなっている。分割数が 5 以上の場合には分解後の概念束のサイズはすべ

て100を下回り、扱いやすいサイズとなった。

実験結果から、属性集合の分割では分割数を大きくすることで、含意関係と整合しない属性集合を減少させずに、分解後の概念束のサイズを十分に小さくできると考えられる。

6. おわりに

本稿では概念束の様々な分解手法について、分解後の概念束の図から読み取れる情報について、理論的な評価を行った。理論的な評価から、各分解手法における分解前の内包・含意関係の読み取りと分解後の読み取りの関係を明らかにした。さらに、正しさ、再現性、復元可能性の観点で分解手法の比較を行った。また、概念束の分解可能性や分解後の概念束による元の含意関係を保存する割合、分解後の概念束から読み取れる含意関係の内正しいものの割合等を実験によって評価した。実験結果から属性集合の分割は分解可能であることが多く、元の含意関係と整合する属性の集合が増加していないことが分かった。さらに、この分解手法は分解後の概念束から読み取れる含意関係が必ず正しいため、他の分解手法より扱いやすいといえる。また、属性集合の分割について注目して実験した結果、分割数を大きくすることで、元の性質を保ったまま分解後の概念束のサイズを小さくできることが分かった。

今後の課題として、属性集合の分割に対して、適切な分割法や分解後の概念束の数の決定法に関する検討があげられる。

参考文献

- [1] Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts, *Ordered Sets*, Springer, pp.445-470 (1982).
- [2] 澤勢一史, 延原 肇: 大規模画像群のための形式概念分析に基づく束構造可視化システム, 知能と情報, Vol.21, No.1, pp.32-40 (2009).
- [3] Kazato, H., Hayashi, S., Oshima, T., Miyata, S., Hoshino, T. and Saeki, M.: Extracting and Visualizing Implementation Structure of Features, *Software Engineering Conference (APSEC), 2013 20th Asia-Pacific*, Vol.1, pp.476-484, IEEE (2013).
- [4] Lindig, C. and Snelting, G.: Assessing Modular Structure of Legacy Code Based on Mathematical Concept Analysis, *Proc. 19th International Conference on Software Engineering*, pp.349-359 (1997).
- [5] Bhatti, M.U., Anquetil, N., Huchard, M. and Ducasse, S.: A Catalog of Patterns for Concept Lattice Interpretation in Software Reengineering, *Proc. 24th International Conference on Software Engineering & Knowledge Engineering*, pp.118-123 (2012).
- [6] 齋藤邦彦, 栗田健士: 形式的概念分析によるグルーピングのCプログラム理解支援への適用, 彦根論叢, No.376, pp.59-78 (2009).
- [7] Rome, J.E. and Haralick, R.M.: Towards a Formal Concept Analysis Approach to Exploring Communities on the World Wide Web, *Formal Concept Analysis*, pp.33-48, Springer (2005).
- [8] Ścibor S. and Zieliński, B.: Modelling role hierarchy structure using the Formal Concept Analysis, *Annales Universitatis Mariae Curie-Skłodowska, Sectio AI-Informatica*, Vol.10, No.2, pp.143-159 (2010).
- [9] Priss, U. and Old, L.J.: Data Weeding Techniques Applied to Roget's Thesaurus, *Knowledge Processing and Data Analysis*, pp.150-163, Springer (2011).
- [10] Viaud, J.-F., Bertet, K., Missaoui, R. and Demko, C.: Using congruence relations to extract knowledge from concept lattices, *Discrete Applied Mathematics*, Vol.249, pp.135-150 (2018).
- [11] 深谷有吾, 石樽隼人, 武藤敦子, 森山甲一, 犬塚信博: 属性間の関連度を用いた分解による概念束の単純化, 情報処理学会研究報告, Vol.2019-MPS-122, No.5, pp.1-6 (2019).
- [12] Funk, P., Lewien, A. and Snelting, G.: Algorithms for Concept Lattice Decomposition and their Application, Technical Report, Technische Universität Braunschweig (1995).
- [13] 石樽隼人, 武藤敦子, 森山甲一, 犬塚信博: 属性情報の図示に基づく概念束分解手法の比較, 情報処理学会研究報告, Vol.2019-ICS-195, No.5, pp.1-7 (2019).
- [14] Ganter, B. and Wille, R.: *Formal Concept Analysis: Mathematical Foundations*, Springer (1999).
- [15] Guigues, J.L. and Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires, *Mathématiques et Sciences Humaines*, Vol.95, pp.5-18 (1986).
- [16] Snelting, G.: Reengineering of Configurations Based on Mathematical Concept Analysis, *ACM Trans. Software Engineering and Methodology*, Vol.5, No.2, pp.146-189 (1996).
- [17] Berry, A., Pogorelnik, R. and Sigayret, A.: Vertical decomposition of a lattice using clique separators, *Proc. 8th International Conference on Concept Lattices and Their Applications*, pp.15-29 (2011).
- [18] Viaud, J.-F., Bertet, K., Demko, C. and Missaoui, R.: The reverse doubling construction, *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, Vol.1, pp.350-357 (2015).
- [19] Ganter, B.: Two basic algorithms in concept analysis, Technical Report, TH Darmstadt (1984).
- [20] Coenen, F.: The LUCS-KDD Discretised/normalised ARM and CARM Data Library, Department of Computer Science, The University of Liverpool, UK (online), available from (http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS_KDD_DN/) (accessed 2019-04-30).
- [21] Dias, S.M., Zárate, L.E., Song, M.A.J. and Vieira, N.J.: Indexes to Evaluate Reduced Concept Lattices, *Supplementary Proc. 14th International Conference on Formal Concept Analysis*, pp.1-16 (2017).



石樽 隼人 (学生会員)

2015年名古屋工業大学工学部情報工学科卒業。2017年同大学大学院工学研究科情報工学専攻博士前期課程修了。現在、同専攻博士後期課程在学中。



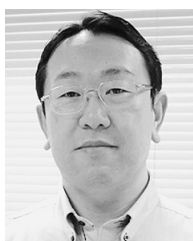
武藤 敦子 (正会員)

1998年名古屋工業大学工学部知能情報システム学科卒業。同年同大学文部科学技官，2004年同大学大学院工学研究科助手，2007年同助教，2016年同教授，現在に至る。博士(工学)。人工知能，人工生命，社会ネットワーク分析に関する研究に従事。IEEE シニア会員，人工知能学会，日本知能情報ファジィ学会，日本数理生物学会各会員。



森山 甲一

1998年東京工業大学工学部情報工学科卒業。2003年同大学大学院情報理工学研究科計算工学専攻博士課程修了。博士(工学)。同専攻助手，大阪大学産業科学研究所助手，助教，特任准教授を経て，現在，名古屋工業大学大学院工学研究科情報工学専攻准教授。人工知能，マルチエージェントシステム等の研究に従事。電子情報通信学会，人工知能学会各会員。



犬塚 信博 (正会員)

1987年名古屋工業大学工学部情報工学科卒業。1992年同大学大学院工学研究科博士後期課程修了。博士(工学)。同年同大学助手。2008年同大学工学研究科教授。現在に至る。人工知能，特に帰納学習，知識発見，社会ネットワーク分析の研究に従事。人工知能の教育への応用等に興味を持つ。人工知能学会，電子情報通信学会，AAAI，ACM 各会員。