Vol.2019-MPS-123 No.41
Vol.2019-BIO-58 No.41
2019/6/19

含意関係に注目した概念束分解手法の評価

石榑 隼人[1,a)]　武藤 敦子[1]　森山 甲一[1]　犬塚 信博[1]

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さまざま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．本研究では，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という観点から，概念束の分解手法を比較評価した．

Evaluation of Decomposition Methods for Concept Lattices Based on Implications

1. はじめに

データマイニングに関する手法の一つに形式概念分析 [2] 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など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 [3] やソースコードの分析 [4] などで利用されている．

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 [5], [6], [7] やコミュニティ検出 [8]，システムの権限設定 [9] で利用されている．概念束の分解手法も，様々な手法が提案されているが，利用可能とする前提を必要とすることが多く，常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に，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，概念束の分解は，簡素化と比較すると，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 [10] 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 [11] 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 [12] 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 [13] は，三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，様々な分解手法を理論的に評価・考察し，また実験による評価を示す．

2. 形式概念分析

本節では文献 [14] に従って，形式概念分析の基礎概念を準備する．また，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．

[1] 名古屋工業大学
Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466–8555, Japan
[a)] h.ishigure.488@nitech.jp

[*1] 本稿は，[1] を基礎として新たな考察と実験を加えたものである．

Vol.2019-MPS-123 No.41
Vol.2019-BIO-58 No.41
2019/6/19

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）		×	×

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，集合は全て有限集合だと仮定する．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，$(g, m) \in I$ である時，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．表中の $\times$ 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) の写像を定義する．

$$X \mapsto X^I = \{m \in M \mid \forall g \in X[gIm]\} \tag{1}$$

$$Y \mapsto Y^I = \{g \in G \mid \forall m \in Y[gIm]\} \tag{2}$$

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

定義 1 (形式概念)．形式文脈 (G, M, I) 及び対象の集合 $A \subseteq G$ と属性の集合 $B \subseteq M$ に対して，組 (A, B) が $A^I = B, B^I = A$ を共に満たす時，これを (G, M, I) の形式概念という．この時，A, B をそれぞれ外延，内包と呼ぶ．また，(G, M, I) の形式概念全ての集合を $\mathfrak{B}(G, M, I)$ と書く．

定義 2 (概念束)．形式文脈 (G, M, I) の二つの形式概念 $(A_i, B_i), (A_j, B_j)$ に対して，以下のように半順序が定義される．

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

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，下限は以下の様になる．

定理 1. 概念束 $(\mathfrak{B}(G, M, I), \leq)$ において，下限，上限は次式の様に定まる．

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

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

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．一つのノードは一つの形式概念を表す．属性 m のラベルは形式概念 $\left(m^I, m^{II}\right)$ に付与される．以降ではこの形式概念を $\mu_I m$ と書く．この形式概念は m を含む形式概念の中で，最上位の形式概念である．例えば，図 1 の概念束において，$\mu_I m_1 = (\{g_1, g_2\}, \{m_1\})$ 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．例えば，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．ここで，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にあるラベルの和集合が $B \subseteq M$ であることを，**図から内包 $B \subseteq M$ が読み取れる**と言うこととする．$\mathfrak{B}(G, M, I)$ に含まれる任意の内包は，図から読み取れる．

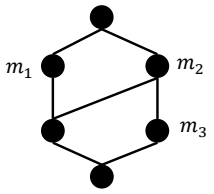

図 1　表 1 の概念束
Fig. 1　Concept Lattice of Table 1

2.2　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．

定義 3 (属性間の含意関係)．(G, M, I) を形式文脈とする．属性集合 $P, Q \subseteq M$ 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，$P \to Q$ を (G, M, I) の含意関係と呼ぶ．

定義より，$P \to Q$ は，任意の対象が P の全ての属性を持つならば，Q の全ての属性も持つことを表す．また，$P \to Q$ が成立することと，全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．例えば，表 1 の形式文脈では，$m_3 \to m_2$ という含意関係が成立する．

また，含意関係は図示された概念束から確認することもできる．概念束 $L = \mathfrak{B}(G, M, I)$ において成立する含意関係の集合を $\mathfrak{L}(L)$ とする．このとき，属性集合 $P, Q \subseteq M$ に対して，式 (3) が成立する．

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

すなわち P に含まれる属性のラベルがある形式概念全ての下限が，Q に含まれる属性のラベルがある形式概念全ての下限より下にある時，$P \to Q$ 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 $P \to Q$ が読み取れる**と言うこととする．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上限は最小

Vol.2019-MPS-123 No.41
Vol.2019-BIO-58 No.41
2019/6/19

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，$m_1, m_3 \to m_2$ が図から読み取れる．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 [15]，atlas decomposition [14]，属性集合の分割，subdirect decomposition [14] を説明し考察を補う．その後，これらの手法を比較する．これらの手法は概念束の分解を求めるのが比較的容易であるため，利用しやすい．ただし，属性集合の分割以外は，常に分解可能ではない．

分解手法には他に，vertical decomposition [16]，reverse doubling construction [17]，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．substitional 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．

定義 4 (horizontal decomposition)．概念束 L に対して，$\top, \bot$ を L の最大元，最小元とする．$L \setminus \{\top, \bot\}$ のハッセ図をグラフとみなした場合の連結成分を $X_1, X_2, \ldots, X_n$，$L_i = X_i \cup \{\top, \bot\}$ としたとき，$L_1, L_2, \ldots, L_n$ を L の horizontal decomposition と呼ぶ．

図 2 は horizontal decomposition の例である．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上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$L_1, L_2, \ldots, L_n$ は，定義より式 (4) の性質を満たす．

$$\bigcup_{i=1}^{n} L_i = L \tag{4}$$

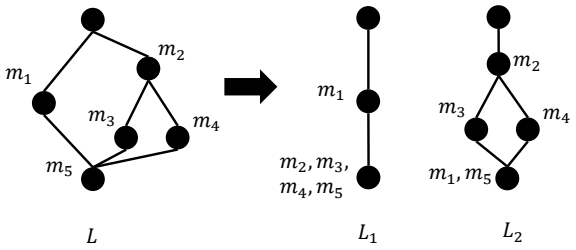

図 2 horizontal decomposition
Fig. 2 Horizontal Decomposition

表 2 表 1 の block relation の例
Table 2 A Block Relation of Table 1

	m_1	m_2	m_3
g_1	×	×	×
g_2	×	×	×
g_3		×	×
g_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．このとき，$J = \bigcup_{(A,B)\in L_i} A \times B$ と書くと，L_i は形式文脈 (G, M, J) の概念束でもある．したがって，分解後の束は概念束であるため，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 $\mu_J m = \mu_I m$ 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 $\bot$ のみである．したがって，この場合は $\mu_J m = \bot$ 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，概念束の構造をわ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 5 (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．

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

表 2 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) を一つ考えると，(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，必ず $\mathfrak{B}(G, M, I)$ において，二つの形式概念 $(A, A^I), (B^I, B)$ に挟まれた形式概念の集合となる．

定義 6 (atlas decomposition)．概念束 $L = \mathfrak{B}(G, M, I)$ に対して，(G, M, I) のある block relation J のブロックを $L_1, L_2, \ldots, L_n$ とするとき，これを L の atlas decomposition

Vol.2019-MPS-123 No.41
Vol.2019-BIO-58 No.41
2019/6/19

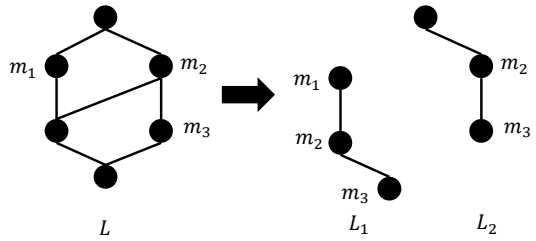

図 3　atlas decomposition
Fig. 3　atlas decomposition

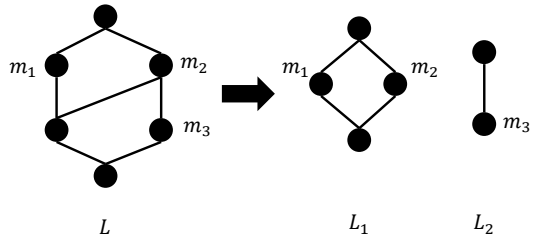

図 4　属性集合の分割
Fig. 4　Partition of the Set of Attributes

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 2 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 $L_1, L_2, \ldots, L_n$ は式 (5) を満たすことがわかる．

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

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 $M_1, M_2, \ldots, M_n$ に対して，$L_i = \mathfrak{B}(G, M_i, I \cap (G \times M_i))$ としたとき，$L_1, L_2, \ldots, L_n$ 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. (G, M, I) を形式文脈，$M = \bigcup_{i=1}^{n} M_i$ とする．このとき，任意の $i = 1, 2, \ldots, n$ に対して，$J = I \cap (G \times M_i)$ とおくと，次の写像は全射である．

$$\begin{aligned} \mathfrak{B}(G, M, I) &\to \mathfrak{B}(G, M_i, J) \\ (A, B) &\mapsto \left((B \cap M_i)^J, B \cap M_i\right) \end{aligned}$$

分解前の概念束の形式概念一つは，分解後の概念束それぞれに対して対応する形式概念を持つ．すなわち，分解前の形式概念は分解後の形式概念の組み合わせで表現できる．

図 4 は属性集合の分割の例である．ここでは，属性集合を $\{m_1, m_2\}, \{m_3\}$ の二つに分割している．

3.4　subdirect decomposition

ある形式文脈に対して，次の様に compatible subcontext が定義される．

定義 7 (compatible subcontext)．(G, M, I) を形式文脈，$(H, N, I \cap (H \times N))$ をその subcontext（即ち $H \subseteq G, N \subseteq M$）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 $\Pi_{H,N} : \mathfrak{B}(G, M, I) \to \mathfrak{B}(H, N, I \cap (H \times N))$ は全射である．

$$\Pi(A, B) := (A \cap H, B \cap N) \tag{6}$$

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 8 (complete congruence relation)．形式文脈 (G, M, I) の compatible subcontext$(H, N, I \cap (H \times N))$ に対して，次式の様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一つが，分解後の形式概念の組み合わせで表現される．

定義 9 (subdirect decomposition)．概念束 $L = \mathfrak{B}(G, M, I)$ に対して，$(H_i, N_i, I \cap (H_i \times N_i)), i = 1, 2, \ldots, n$ を (G, M, I) のある compatible subcontext とする．また，$L_i = \mathfrak{B}(H_i, N_i, I \cap (H_i \times N_i)), i = 1, 2, \ldots, n$ とする．このとき，各 compatible subcontext から導出される complete congruence relation $\Theta_i, i = 1, 2, \ldots, n$ について，以下の条件が成立するならば，$L_1, L_2, \ldots, L_n$ を L の subdirect decomposition と呼ぶ．

$$\bigcap_{i=1}^{n} \Theta_i = \{(x, x) \mid x \in L\}$$

図 5 は概念束上の二つの complete congruence relation を表す．図中で色が同じ形式概念同士は同値である．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，分解後の二つの概念束のどちらにも入ることがない．

Vol.2019-MPS-123 No.41
Vol.2019-BIO-58 No.41
2019/6/19

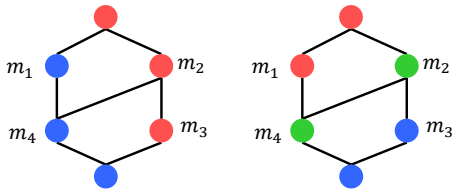

図 5　complete congruence relation
Fig. 5　Complete Congruence Relation

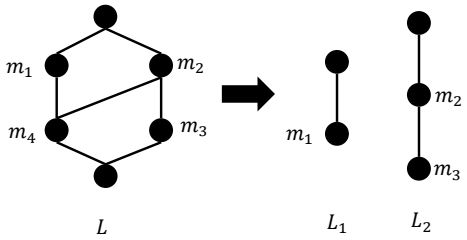

図 6　subdirect decomposition
Fig. 6　Subdirect Decomposition

4. 理論的評価

本節では，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，また直接読み取れない場合にどのような条件が必要かについて，分解手法の理論的な評価を行う．以下では，分解前の概念束を L，分解後の概念束を $L_1, L_2, \ldots, L_n$，L_i に含まれる属性の集合を M_i と書く．また，概念束 L に含まれる内包の集合を $\mathrm{in}(L)$ 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は，以下の関係があることが証明できる．

$$B \in \mathrm{in}(L_i) \Rightarrow B \in \mathrm{in}(L) \tag{7}$$

$$P \to Q \in \mathfrak{L}(L) \Rightarrow P \to Q \in \mathfrak{L}(L_i) \tag{8}$$

$$P \to Q \in \mathfrak{L}(L_i) \text{ and } \forall m \in P[\mu_J m \neq \bot] \Rightarrow P \to Q \in \mathfrak{L}(L) \tag{9}$$

$$P \to Q \in \mathfrak{L}(L_i) \text{ and } \exists m \in P[\mu_J m = \bot] \text{ and } \exists m \in P[\mu_J m \neq \bot, \top] \Rightarrow P \to Q \in \mathfrak{L}(L) \tag{10}$$

分解後の概念束全てを用いた場合，分解前の内包と分解後の内包の関係は式 (4) で表される．分解前の含意関係と分解後の含意関係の間の関係は，式 (11) 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構成できる．

$$\mathfrak{L}(L) = \bigcap_{i=1}^{n} \mathfrak{L}(L_i) \tag{11}$$

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，以下の関係が示せる．

$$B \in \mathrm{in}(L_i) \Rightarrow B \in \mathrm{in}(L) \tag{12}$$

$$P \to Q \in \mathfrak{L}(L) \text{ and } P \cup Q \subseteq M_i \Rightarrow P \to Q \in \mathfrak{L}(L_i) \tag{13}$$

$$P \to Q \in \mathfrak{L}(L_i) \text{ and } P \supseteq \{m \in M_i \mid \mu_J m = \top\} \Rightarrow P \to Q \in \mathfrak{L}(L) \tag{14}$$

また，分解後の概念束全てを用いた場合，式 (5) のように，元の概念束の内包が全て得られることがわかっている．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構成できる．

$$\mathfrak{L}(L) = \bigcap_{i=1}^{n} (\mathfrak{L}(L_i) \cup \{P \to Q \mid P \not\subseteq M_i\}) \tag{15}$$

4.3 属性集合の分割

属性集合の分割では，分解後の概念束一つに対して，以下が確認できる．

$$B \in \mathrm{in}(L_i) \Leftrightarrow \exists D \in \mathrm{in}(L)[D \cap M_i = B] \tag{16}$$

$$P \to Q \in \mathfrak{L}(L) \text{ and } P \cup Q \subseteq M_i \Leftrightarrow P \to Q \in \mathfrak{L}(L_i) \tag{17}$$

また，分解後の概念束全てを用いた場合では，これ以上の情報は得られない．

4.4 subdirect decomposition

subdirect decomposition では，属性集合の分割と同様に以下が確認できる．

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

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

分解後の概念束全てを用いた場合でも，属性集合の分割と同様に，L の内包と含意関係について，全ての情報は読み取れない．また，subdirect decomposition では，$\bigcup_{i=1}^{n} M_i = M$ が成立しない場合があるため，この場合には，$(\bigcup_{i=1}^{n} M_i) \setminus M$ に含まれる属性については全く知識が得られない．

5. 実験的評価

5.1 実験手法

本稿では，概念束の分解が可能かや分解後の概念束が元

表 3　実験に利用したデータ
Table 3　Data Used in Experiment

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	3205
Glass	163	44	4744
LED 7	326	24	7039

の性質をどれだけ保つかを，実験的に評価する．

実験は以下の様に行う．まずデータとして，[18] で公開されているデータを用いた．これらは全て離散化されているが，二値ではない．そのため，データ上の各属性値を持つかどうかを新たな属性とすることで，形式文脈に変換した．また，このように変換した形式文脈から冗長な対象と属性を全て取り除いた．冗長な対象と属性とは，それを取り除いても概念束が元と同型となるような対象と属性である．**表 3** は，実験で利用したデータである．

その後，作成した形式文脈から得られる概念束に対して，分解可能であるかを確認した．確認した手法は horizontal decomposition，atlas decomposition，属性集合の分割，subdirect decomposition である．属性集合の分割では，[12] で提案された手法で分割の仕方を定めた．この手法では階層的クラスタリングを利用するが，本稿では群平均法を用いた．

更に分割可能であった場合には，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うえで，分解後の概念束の性質について評価を行った．ここでは，分解後の概念束の図から読み取れる含意関係のうち，分解前でも成立することがわかる含意関係について評価した．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全て，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．

評価指標としては，information content[19] を利用した．information content は含意関係の集合に対して，それらによる制約の強さを評価する値である．値が 1 に近いほど含意関係による制約が強く，そのため含意関係が多くの情報を持つこととなる．実験では分解前でも成立する含意関係についてのみ評価するため，分解後の含意関係の information content は，分解前の含意関係の information content 以下となる．このため，元の概念束で成立する含意関係に対する information content の比を指標として用いる．

5.2　結果と考察

表 4 は各分解手法で分解を行った際に得られた information content の比である．表中の – は分解ができなかったことを表す．horizontal decomposition と subdirect decomposition では概念束の分割自体がまったくできなかった．atlas decomposition もあまり分割ができなかった．一方で，属性集合の分割では常に分割が可能だった．また，属性集合の分割は元の含意関係全ては保存する手法ではないが，実験結果から含意関係のほとんどを保っていることがわかる．

表 4　information content
Table 4　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	–

Ecoli と Page blocks では，二つの手法で分割が可能だった．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．属性集合の分割は分解後の形式概念の組み合わせで元の概念束をあらわすため，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，実験では必ずしもそのような結果とはならなかった．また属性集合の分割による分解後の概念束のサイズには偏りがあるが，これは [12] で用いられている階層的クラスタリングの性質によると考えられる．

以上から，属性集合の分割を除く手法では，概念束を分解すること自体が困難であると言える．しかし，horizontal decomposition はソフトウェア工学の分野での利用例があるため，特定の種類のデータでは，より利用しやすい可能性がある．

一方で属性集合の分割は，ほとんど必ず概念束の分解が可能である．また，含意関係についても，全てではないがほとんどを保存する．このため，属性集合の分割は，利用しやすい手法だと考えられる．しかし，分解後の概念束のサイズが小さくなりやすいと期待されるが，実験では必ずしもそのようにならなかった．また，この手法は分割の仕方や分割する数に結果が大きく依存するため，この点に関してより検討が必要である．

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．また，概念束の分解が可能であるかや分解後の概念束が元の含意関係をどれだけ保存するかを，実験的に評

価した．実験結果から属性集合の分割は分解がほとんど常に可能で，元の含意関係をほとんど保存することが分かった．

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挙げられる．

参考文献

[1] 石榑隼人，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).
[2] Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts, *Ordered Sets*, Springer, pp. 445–470 (1982).
[3] 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).
[4] 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, IEEE, pp. 476–484 (2013).
[5] Lindig, C. and Snelting, G.: Assessing Modular Structure of Legacy Code Based on Mathematical Concept Analysis, *Proceedings of the 19th International Conference on Software Engineering*, pp. 349–359 (1997).
[6] Bhatti, M. U., Anquetil, N., Huchard, M. and Ducasse, S.: A Catalog of Patterns for Concept Lattice Interpretation in Software Reengineering, *Proceedings of the 24th International Conference on Software Engineering & Knowledge Engineering*, pp. 118–123 (2012).
[7] 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).
[8] Rome, J. E. and Haralick, R. M.: Towards a Formal Concept Analysis Approach to Exploring Communities on the World Wide Web, *Formal Concept Analysis*, Springer, pp. 33–48 (2005).
[9] Ścibor Sobieski and Zieliński, B.: Modelling role hierarchy structure using the Formal Concept Analysis, *Annales Universitatis Mariae Curie-Sklodowska, sectio AI–Informatica*, Vol. 10, No. 2, pp. 143–159 (2010).
[10] Priss, U. and Old, L. J.: Data Weeding Techniques Applied to Roget's Thesaurus, *Knowledge Processing and Data Analysis*, Springer, pp. 150–163 (2011).
[11] 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).
[12] 深谷有吾，石榑隼人，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).
[13] Funk, P., Lewien, A. and Snelting, G.: Algorithtms for Concept Lattice Decomposition and their Application, Technical report, Technische Universität Braunschweig (1995).
[14] Ganter, B. and Wille, R.: *Formal Concept Analysis: Mathematical Foundations*, Springer (1999).
[15] Snelting, G.: Reengineering of Configurations Based on Mathematical Concept Analysis, *ACM Transactions on Software Engineering and Methodology*, Vol. 5, No. 2, pp. 146–189 (1996).
[16] Berry, A., Pogorelcnik, R. and Sigayret, A.: Vertical decomposition of a lattice using clique separators, *Proceedings of the 8th International Conference on concept Lattices and Their Applications*, pp. 15–29 (2011).
[17] 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).
[18] 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).
[19] Dias, S. M., Zárate, L. E., Song, M. A. J. and Vieira, N. J.: Indexes to Evaluate Reduced Concept Lattices, *Supplementary Proceedings of 14th International Conference on Formal Concept Analysis*, pp. 1–16 (2017).