ユニットのクラスタリングによる BOK エリア自動抽出

Units Clustering for BOK Areas Creation

矢吹 太朗

エリア・ユニット・トピックという階層構造を持つ知識体系（Body of Knowledge, BOK）の構築において，ユニットのクラスタリングによってエリアを見出す手法を提案する．この手法には，BOK と関連が深いと思われる授業からユニットを収集するだけで BOK を構築できるという利点がある．本稿では，計算機科学のための BOK である Computing Curricula 2001 Computer Science 掲載の授業構成例から BOK を再現することを試み，提案手法の有効性を議論する．

1 Body of Knowledge

知識体系（Body of Knowledge, BOK）は特定の分野を構成する概念をまとめたものである．代表的な BOK の 1 つに，Computing Curricula 2001 Computer Science（CC2001）に収録された，計算機科学の BOK，CS Body of Knowledge（CSBOK）がある[1]．

1.1 BOK 木

CSBOK は木構造で表現できる BOK である．本稿では，BOK を表現する木を BOK 木と呼び，BOK 木で表現される BOK のみを扱うが，木構造で表現されない BOK もある．たとえば，プロジェクトマネジメントのための BOK である PMBOK は木構造ではない[2]．

CC2001 では，ルートから数えて最初の階層（第 1 階層）をエリア，第 2 階層をユニット，第 3 階層をトピックと呼んでおり，本稿でもこの用語を用いる．

CSBOK には以下の 14 個のエリアがある．

1. Discrete Structures
2. Programming Fundamentals
3. Algorithms and Complexity
4. Architecture and Organization
5. Operating Systems
6. Net-Centric Computing
7. Programming Languages
8. Human-Computer Interaction
9. Graphics and Visual Computing
10. Intelligent Systems
11. Information Management
12. Social and Professional Issues
13. Software Engineering
14. Computational Science and Numerical Methods

1.2 WikiBOK

新生学問分野（例：社会情報学）の BOK を

Yabuki Taro 千葉工業大学社会システム科学部プロジェクトマネジメント学科

作成することは一般に困難だと考えられている．なぜなら，CSBOK を構築した IEEE Computer Society と ACM の統合委員会のメンバのような，その分野全体に精通する有識者がいないからである．そこで，必ずしも有識者ではない多くの参加者の集合知で BOK を構築することが試みられている．WikiBOK である[3]．WikiBOK は，その場で簡単に編集できるウェブページである Wiki を使って，不特定多数の集団で BOK 木を構築しようという試みである．

WikiBOK は有望な試みではあるが，以下のような問題が指摘されてきた．

- エリア・ユニット・トピックの使い分けが難しい．エリアとは何か，ユニットとは何か，トピックとは何かということについて，参加者がそれぞれ独自の考えを持っていて，それらを統一しにくい．
- 個人が BOK 木全体を見渡すのが難しい．参加者にとって，BOK 木の大部分は自分にとって専門外の領域であり，そういう中で自分が貢献する領域を位置づけにくい．

これらの問題点を踏まえ，参加者が自分の専門分野について作業するだけで BOK 木構築に貢献できる方法が求められており，本稿ではそのような方法を提案する．

2 BOK とカリキュラム

本節では，大学の教育カリキュラムと BOK の関係について述べてから，カリキュラムをもとに BOK 木を構築する方法を提案する．

2.1 BOK からカリキュラムへ

CC2001 では，CSBOK をもとにした複数のカリキュラムが提案されている．そのうちの 1 つのカリキュラムの一部を図 1 に示す．ここで重要なのは，BOK のエリアやユニットが，そのまま授業になるわけではないことである．カリキュラムは，知識を教授するために BOK を再構成したものであり，BOK を分割してできるものではない．

2.2 提案手法：カリキュラムから BOK へ

5 個のユニットからなる BOK を，4 つの授業でカバーするようなカリキュラム（表 1）を例に，提案手法を説明する．

	CS111 T. Intro to Programming	CS112 T. Data Abstraction	CS115. Discrete Structures	CS210 T. Algorithm Analysis	CS220 T. Computer Architecture	CS225 T. Operating Systems	CS230 T. Net-centric Computing	CS260 T. Artificial Intelligence	CS270 T. Databases	CS280 T. Social and Prof Issues	CS290 T. Software Development	CS490. Capstone Project	Total	Extra hours
DS1. Functions, relations, and sets			6										6	
DS2. Basic logic			10										10	
DS3. Proof techniques			9	3									12	
DS4. Basics of counting			5										5	
DS5. Graphs and trees		2		4									6	+2
DS6. Discrete probability			6										6	
PF1. Fundamental programming constructs	9												9	
PF2. Algorithms and problem-solving	3			3									6	
PF3. Fundamental data structures	6	6		3									15	+1
PF4. Recursion		5											5	

図 1 CSBOK をもとにしたカリキュラムの一部[1]．行が CSBOK のユニットに，列が授業に対応している．数字はユニットに割り当てられる時間である．

2.2.1　ユニットの抽出

まず，授業の構成要素を列挙し，ユニットとする．ユニットにかける時間もわかるとよい（わからない場合はすべて「1」でもよい）．この時点でユニットの順番はばらばらである．

表の行をユニットの特徴ベクトルとみなし，ユニット間の相違度を計算する．たとえば，非共通授業の割合を相違度とすれば，ユニット間の相違度は表 2 のようになる．

2.2.2　ユニットのクラスタリング

ユニット間の相違度が定まると，ユニットをクラスタリングできるようになる．クラスタリング手法には，階層的クラスタリングと非階層的クラスタリングがあるが，予備実験において，非階層的クラスタリングの結果が悪かったため，本稿では階層的クラスタリングの結果のみを示す．

階層的クラスタリングは次のような手順で行われる．

1．要素やクラスタ間の相違度を測定する．
2．相違度が小さいもの同士をクラスタにする．
3．クラスタに属していない要素があれば 1 に戻る．

表 1 のユニットに対して階層的クラスタリングを行った結果は，図 2 のようになる．このデンドログラムを水平に切ることによって，5 個のユニットを 3 あるいは 5 個のクラスタに分けることができる．クラスタを BOK 木のエリアと見なせば，カリキュラムから BOK 木のエリアとユニットが生成できたことになる．

以上をまとめると，本提案手法の実行手順は以下のようになる．

1．授業の構成要素をユニットとして抽出する
2．授業とユニットの関係を行列で表す
3．ユニット同士の相違度を計算する（相違度が小さいことは，同じような授業で教えられていることを意味する）
4．ユニットをクラスタリングする
5．クラスタをエリアと見なし，名前を付ける

表 1　ユニットと授業の例

		授業			
		CS111I	CS115	CS210T	CS220T
ユニット	DS2		10		
	PF1	9			
	DS3		9	3	
	PF2	3		3	
	AR7				3

表 2　ユニット間の相違度（非共通授業の割合）

	DS2	PF1	DS3	PF2
PF1	4/4			
DS3	3/4	4/4		
PF2	4/4	3/4	3/4	
AR7	4/4	4/4	4/4	4/4

2.3　提案手法の利点

提案手法には 2 つの利点がある．

第 1 に，エリア・トピック・ユニットの使い分けに悩む必要がなくなる．エリアはユニットのクラスタであり，ユニットは授業の構成要素であり，トピックはユニットの構成

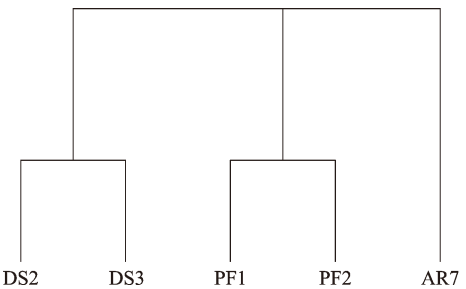

図 2　表 1 のユニットに対して階層的クラスタリングを行った結果

要素である．1つのカリキュラムをもとにBOKを構築するため，ユニットの粒度はそのカリキュラムに合ったものになる．

第2に，BOK全体を見る必要がなくなる．構築しようとするBOKの全体像はわからなくても，自分が担当している授業のことはわかっているはずであり，それを適切なユニットに分割することは比較的容易だと思われる．

例として，「プロジェクトマネジメント」をBOKの構成要素とすることを考える．筆者の所属するプロジェクトマネジメント学科をBOKで表現しようとすれば，プロジェクトマネジメントはBOK木のルートになるだろう．しかし，多くの教育機関では，プロジェクトマネジメントは半期（90分の授業を15回分）だけで教えられるため，1つのエリア程度の重要度になるだろう（ユニットに分解されて複数のエリアに配分されるということもあり得る）．このように，プロジェクトマネジメントの相対的な重要度は，教えられる環境によって大きく異なる．そのため，重要度についての基準なしにBOKを構築しようとすると，プロジェクトマネジメントをエリア・トピック・ユニットのどのレベルに配置するかを議論しなければならなくなる．しかし，プロジェクトマネジメントを専門とする参加者が少ない場合には，BOKの中でそれがどの程度の重要度になるかを判断するのは難しい．これに対して，本提案手法では，ユニットとして採用すべきものがカリキュラムからある程度決まってしまうため，任意の項目がどのレベルになるのかを，参加者が議論する必要はほとんど無い．

3 検 証

提案手法の有効性を検証するために，CC2001に掲載されている2つのカリキュラムからCSBOKを再現することを試みる．利用するカリキュラムは以下の2つである．

カリキュラム1 手続き的手法を学んだ後で伝統的なトピックについて学ぶ

カリキュラム2 オブジェクト指向を学んだ後で計算機科学を圧縮した形で学ぶ

ここではCC2001において基本的だと見なされている63個のユニット（コアユニット）のみを用いる．それに対応するCSBOKのエリアは13個であるため，以下の実験ではクラスタ数を13に固定する．BOKのエリア数が多すぎるのも少なすぎるのも好ましくないが，このように固定できるのは，この実験のように正解がわかっている場合のみである．

3.1 クラスタリングにおける選択肢

クラスタリングを実行する際には，要素（ユニット）間の相違度とクラスタの連結方法に，以下で述べるような選択肢がある．

相違度の指標には以下のようなものがある．

- ユークリッド距離
- 平方ユークリッド距離
- マンハッタン距離
- チェビシェフ距離
- キャンベラ距離
- 余弦距離
- 相関距離
- ブレイ・カーティス距離

クラスタの連結方法には以下のようなものがある．

- 最短距離法
- 群平均法
- 最長距離法
- 重み付き群平均法
- クラスタの重心からの距離
- クラスタのメジアンからの距離
- Wardの最小分散法

3.2　正解との比較指標

クラスタリング結果と正解との比較指標としては，次のようなものを考える．

再現率　クラスタ内の要素の全ペアで，正解の中にあるペアのうち，再現されたものの割合

精度　クラスタ内の要素の全ペアで，見いだされたペアのうち，正解の中にあるものの割合

Ｆ値　再現率と精度の調和平均

3.3　結果（カリキュラム１）

相違度の指標やクラスタの連結方法を変えながら，カリキュラム１を用いてユニットをクラスタリングした結果の中で，得られたＦ値の最大値は約 0.82 であった．このときのデンドログラムを図３に示す．この結果は以下の条件の下で得られた（ちなみに，ユニットにかける時間数がわからない状況を想定し，特徴ベクトルの成分をブール値にして実験した場合のＦ値の最大値は約 0.66 であった）．

- カリキュラム１を列方向で標準化する．
- 相違度はブレイ・カーティス距離 $\Sigma(u-v)/\Sigma(u+v)$ で計算する．
- 連結法は Ward の最小分散法を用いる．

3.3.1　考察（カリキュラム１）

上述の結果から，カリキュラム１を用いれば，CSBOK の 80％程度は再現できることがわかる．コアユニットのみでの話ではあるが，授業の構成要素から，BOK のエリアをある程度生成できることが期待される．アルゴリズムのサポート無しに，人手でこれだけ再現するのは難しいだろう．

3.4　結果（カリキュラム２）

相違度の指標やクラスタの連結方法を変えながら，カリキュラム２を用いてユニットをクラスタリングした結果の中で，得られたＦ値の最大値は約 0.54 であった．この結果は以下の条件の下で得られた．

- 列方向の標準化は行わない（カリキュラ

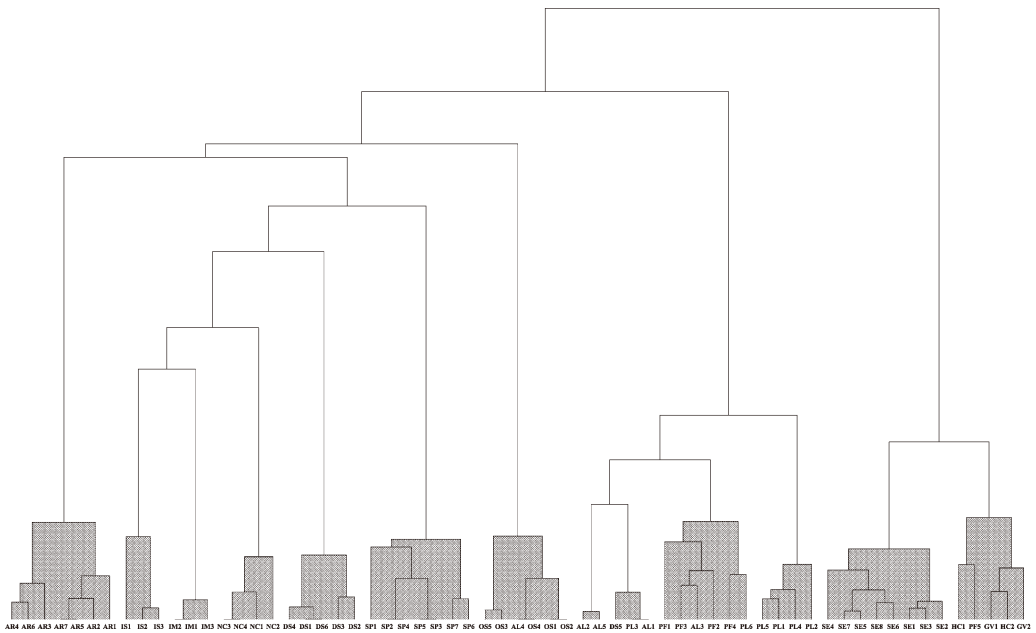

図３　カリキュラム１を用いてユニットをクラスタリングした結果（$F=0.82$）．塗りつぶした部分がクラスタである．

ム１の場合とは異なる）．
- 相違度はブレイ・カーティス距離 $\sum(u-v)/\sum(u+v)$ で計算する．
- 連結法は群平均法を用いる（カリキュラム１の場合とは異なる）．

3.4.1 考察（カリキュラム２）

カリキュラム２を用いるよりも，カリキュラム１を用いた方が，CSBOK をよく再現できる理由は２つ考えられる．

第１に，カリキュラム１は伝統的なカリキュラムであるのに対して，カリキュラム２は新しいカリキュラムである．もし，BOK 自体が伝統的な思想で作られていたとすれば，カリキュラム１のほうがそれによく対応している可能性がある．

第２に，カリキュラム１はカリキュラム２よりも授業数が多い．データが多いほうがクラスタリングがうまくいく可能性がある．

3.5 結果（カリキュラム１と２）

カリキュラム１とカリキュラム２をあわせて１つのカリキュラムと見なし，そのカリキュラムを用いて，相違度の指標やクラスタの連結方法を変えながらユニットをクラスタリングした結果の中で，得られたＦ値の最大値は約 0.76 であった．

3.5.1 考察（カリキュラム１と２）

カリキュラム１とカリキュラム２を合わせた場合の結果（$F=0.76$）は，データが多いにもかかわらず，カリキュラム１だけを用いた場合（$F=0.82$）よりも悪くなっている．このことは，BOK を作成する際に，複数のカリキュラムを混ぜることの危険性を示唆しているが，カリキュラム２の結果のような小さいＦ値（$F=0.54$）につながるカリキュラムを混ぜたのが原因だという可能性もある．

4 課 題

本節では，本提案手法の課題について述べる．

CC2001 の授業例の記述は，以下のような項目からなっている．

- 概要
- 履修に必要な条件
- シラバス（ふだん大学教員が書いているもの）
- カバーされるユニット（CC2001 ではシラバスと類似しているが，別のものである．クラスタリングのためにはこの記述が必要だが，ふだん大学教員がこれを意識することはない．）

カリキュラム１やカリキュラム２を見ると，ユニットは授業間で共有されていることがわかる．しかし，カリキュラムから単純に授業の構成要素を取り出すと，授業ごとに独立なユニット群ができるだけになる危険がある．この危険を回避するためには，授業間で共通のユニットが存在するように，ユニットを整理しなければならない．

ユニットの整理する方法にはさまざまなものが考えられるが，その一例を以下に挙げる．

- 授業の構成要素をユニットとして登録する．
- １つの授業で独自のユニットは３個までとする．
- 他に必要なユニットは，既存のユニットを再利用して補わなければならない（既存のユニットの名前を変えてもよい）．

この条件は，他の参加者の作業がある程度進まないと自分の作業も進められないことにつながるが，多くの教員の参加を促し，集合知の力を増大させる効果もあるだろう．

このように，BOK 構築に参加するために

は，自分の専門領域（授業）についての情報を提供することのほかに，他の授業と共有できるユニットを見いだすという作業が必要だと思われる．

5　まとめ

木構造で表現できる BOK（BOK 木）を複数人で構築する試みである WikiBOK をサポートするために，授業の構成要素をユニットとして抽出し，そのクラスタリング結果をエリアとする方法を提案した．CC2001 のカリキュラムから CSBOK を再現する実験を行い，提案手法が BOK 構築に有効であることが確認できた．この手法を用いて実際に BOK を生成するためには，授業間で共有可能なユニットを見いだす必要があるが，その方法も提案した．新生学問分野のような，BOK なしに教育が行われている分野の BOK 構築に，本手法が実際に有効であることを確認するのが今後の課題である．

参考文献

[1] The Joint Task Force on Computing Curricula IEEE Computer Society and Association for Computing Machinery (2001) *Computing Curricula 2001 Computer Science*

[2] Project Management Institute (2008) *A Guide to the Project Management Body of Knowledge (PMBOK Guides)*, Project Management Institute

[3] 増永良文，石田博之，伊藤一成，伊藤　守，清水康司，荘司慶行，高橋　徹，千葉正喜，長田博泰，福田亘孝，正村俊之，森田武史，矢吹太朗（2012）「知識体系（BOK）創成支援システム WikiBOK の研究・開発」『第 3 回ソーシャルコンピューティングシンポジウム講演論文集』67–72