

# 2P-4 状況理論に基づく知識獲得における 状況の再構成<sup>1</sup>

阿部 明典

NTTコミュニケーション科学研究所

### 1. はじめに

状況理論に基づく知識獲得<sup>(1)</sup> で得られた知識は状況毎に個別に得られた知識なので、その知識(状況)の再構成が必要となる。更に、それにより、階層性のなかった知識(状況)に階層性を与えることが可能となる。本論文では文献[1]で得られた知識において状況の再構成を行う手法を示す。事象や対象を単純な概念のクラスにグループ分けする手法としては分類のゴール、分類の評価基準、演繹的推論規則、帰納的推論規則を含む背景知識により、意味のある分類を行ない、ゴールに関する概念記述はゴール間依存ネットワークとして組織化することで低レベルの記述から高レベルの記述を導き出すことができる概念クラスタリング<sup>(2)</sup>がある。この手法では従来の属性に基づく概念クラスタリングと比べて、背景知識を与えることにより文脈依存のクラスタリングが可能になっている。しかし、本論文に示す手法では背景知識を明示的に与えずに状況の再構成(クラスタリング)を行なう。概念クラスタリングに於ける背景知識に相当するものはその知識の得られた状況と考えており、これにより状況が変わると意味を変える知識の再構成が可能となる。

### 2. 過去の推論、知識獲得の履歴による状況の再構成

従来のクラスタリング手法では前もって背景知識等を用意して初めてクラスタリングが可能になる。一方、本論文に示す手法では、背景知識を明示的に与えずに状況の再構成(クラスタリング)を行なう。特に、ここに示す状況の再構成は、文献[1]の知識が対象であり、状況が変わると意味を変える、状況により個別化された知識を支持する状況の再構成である。尚、状況とは知識の集合であると考えられる。例えば、ある対象を好ましい状況であると認識するのは「気分がいい」などの好感度の高い知識がそこに集合していると認識しているのである。具体的には、過去に於いて、同じように、状況が変わった場合に意味を変えた知識を集めてそれらを再構成すべき状況と看做し、再構成を行う。

文献[1]では推論の際、状況が異なった時に意味を変えた知識を特殊な知識として個別化して獲得し、その旨を知識に記録していた。従って、知識に記録されている過去の推論、知識獲得の履歴が背景知識になりうるので、特別に背景知識を与えることなく、状況を再構成できる。更に、その再構成されたものは、状況により意味を変える変え方が類似している知識の集合になる。例えば、過去に於いて同じように、誤りを指摘された場合、人間はそれらを同じ状況のものとしなす。そして、それらを集めることにより、より一般的な状況として状況の再構成を行なう。例えば、レストランにしか行ったことのない人がマクドナルドやロッテリアで前払いであるという知識を獲得したら、ファーストフードという状況ではお金の支払い方が前払いであると思うようなものである。従って、本論文の手法で再構成された状況は、状況が変わると意味が変わる知識を扱う推論により適したものとなり、知識の階層化も可能である。

### 3. 状況の再構成の戦略

#### 3.1 知識ベース及び、戦略

知識ベースとして、文献[1]の知識ベース(尚、文献[1]で基本知識ベースといていたのはここでは普通知識ベース、追加知識ベースといていたのは状況依存知識ベースという。)に加えて状況管理ベースと状況インデックスを用意する。状況管理ベースは、再構成された状況を管理するためのもので、第一項は新たに再構成された状況名、第二項は知識のリスト、第三項は元の状況名のリストである。状況インデックスは新たに構成された状況と元の状況の対応づけのためのインデックスである。第一項が元の状況の名前、第二項に新たに作成された状況の名前のリストがはいる。

#### 状況管理ベース

```
[situationA, {knowledge1, knowledge2, ...},
             {situation1, situation2, ...}]
.....
```

#### 状況インデックス

```
{[situation1], [situationA]}
{[situation2], [situationA]}
.....
```

図1 状況管理ベースと状況インデックスの構造

以下に戦略の概略を簡単に示す。詳細は例で示す。

**Step 1:** 状況データベースの状況のデータのうち、複数の知識に関して状況に2つ以上の共通集合があればそれを集める。ない場合は処理を終了する。尚、これは、最大の知識集合に対しての共通集合であり、それに該当する集合の中では最大集合であるものを選ぶ。そして、その共通集合に対して新たに名前をつけ、状況依存知識ベース、状況データベースの該当の知識の状況の項の名前をその名前書きかえる。そして、状況管理ベースに再構成した状況の状態を例や図1に示すように記録する。更に、状況インデックスに状況の集合の様子を登録する。

**Step 2:** 状況依存知識ベースの状況の名前を書き換えた知識に対応する普通知識ベースの知識(状況が変わると意味を変える知識の、状況が変わる前の知識)に対する状況も同様に再構成する。普通知識の知識のうち、前記した状況依存知識ベースの知識に対応する普通知識ベースの知識が成立する状況が支持する知識を全て集める。そして、その知識集合(=状況)に対して新たに名前をつけ、普通知識ベースの該当の知識の状況の項の名前を書きかえる。そして、状

1 Grouping Situations for The Knowledge Acquisition System Based on Situation Theory  
Akionori Abe  
NTT Communication Science Laboratories.  
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

況管理ベースに再構成した状況の状態を例や図1に示すように記録する。更に、状況インデックスに状況の集合の様子を登録する。

更に、再構成された状況の集合に関して同様のことを行う。処理過程は前述のものと同様であり、前述の過程を繰り返す。goto Step1。

### 3. 2 例

上記の状況の再構成の戦略を簡単な例で示す。図2に状況の再構成の前の知識ベースを示し、図3に状況の再構成の後の知識ベースを示す。以下の例では、図2に示す知識に関する状況を再構成する。まず、状況依存知識ベースの知識から始める(Step1)。状況データベースを見ると、例えば、<<pay, money, before, 0>>、<<serve, dish, yourself, 0>>、<<put, lap, in, trash, 0>>、<<pay, service, 1>>という知識に関しては{Mac, KFC, Lott}という状況の集合が最大の共通集合になり、知識も最大集合である。そこで、システムはその状況の共通集合にSA0と名前をつける。システムは、状況依存知識ベースの中の知識を{SA0, {<<pay, money, before, 1>>}}などと書き換え、状況データベースの中のデータを[<<pay, money, before, 0>>, {SA0, Mr\_D, Moss}]などと書き換える。次に、この再構成した状況ともとの状況が関係づけられるように状況管理ベースに[SA0, {<<pay, money, before, 1>>, <<serve, dish, yourself, 1>>, <<put, lap, in, trash, 1>>, ...}, {Mac, KFC, Lott}]と記録し、状況インデックスに[Mac, {SA0}], [KFC, {SA0}], [Lott, {SA0}]と記録する。次に、状況を再構成した状況依存知識に対応する普遍知識ベースの知識(例えば、<<pay, money, before, 1>>に対応する知識は<<pay, money, before, 0>>である)を支持する状況を再構成する(Step2)。Step1で再構成した状況が支持する知識に対応する普遍知識を支持する状況を集め、更に、その状況が支持する知識をも全て集め、一つの状況集合とみなす。この例の場合、状況の名前の集合は{Heichinro, Chez\_Ino, Sabattini}で、システムはその名前をSB0とする。そして、普遍知識ベースの中の知識を[SB0, {<<pay, money, before, 0>>}, 1]などと書き換える。次に、状況管理ベースに[SB0, {<<pay, money, before, 0>>, <<serve, dish, yourself, 0>>, <<put, lap, in, trash, 0>>, <<eat, food, on table, 1>>, ...}, {Heichinro, Chez\_Ino, Sabattini}]と記録し、状況インデックスに[{Heichinro, {SB0}}, {Chez\_Ino, {SB0}}, {Sabattini, {SB0}}]と記録する。これで、状況の再構成の1ターンが終わる。更に、状況の再構成が可能な場合は、処理はループする。上に示したのと同様に、Step1、Step2の処理を施すことにより図3に示すような知識ベースに再構成される。ここに示した例の場合、これ以上の再構成は出来ないで、これで処理を終了する。

### 4. まとめ

文献[2]では、学習を分節化された状況と情報の関係として一般化・特殊化するものと捉えている。そして、分節化とは情報が特定の組み合わせで現れたときにこれを過去の組み合わせと比較し、同じ組み合わせに対しては同じ対象を認識したとして、その部分を状況の残りの部分から分離するということとしている。本論文で示した手法では、もともとの処理システムに於いて、状況が変わると意味を変える知識を個別化するという性質があり、学習とはその個別化した状況を再構成することである。従って、再構成にあたっては、同じように個別化されたものをまとめる。又、知識ベースには個別化されたときの状況の情報がかかわれているので、それを背景知識として使うことが可能なので、背景知識を明示的に与えることなく状況が変わると意味を変える知識を使った状況依存推論がしやすい状況を再構成することが可能となる。尚、今回示したシステムでは複数の構成の可能性は考慮していない。例えば、例ではSA1として、{SA0, Mr\_D}と{SA0, J&B}の2つの可能性があるが、本システムでは先に処理した方のみ再構成を行っている。双方を処理するにはマルチインヘリタンスのような考え方が必要であろう。

### 謝辞

本研究の機会を与えていただいたNTTコミュニケーション科学研究所の西川清史所長、中野良平主幹研究員に感謝致します。

### \*\*\* 参考文献 \*\*\*

- [1] 阿部 明典：状況理論に基づく学習、信学会研究技報、A192-82、1992
- [2] 中島 秀之：状況依存性からみた知識像(その4) - 学習 -、Proc. of WOL'91、pp.155-159、1991
- [3] R.E. Stepp, R.S. Michalski：概念クラスタリング 構造をもつ対象のコール指向分類、発見的学習、pp.55-84、共立出版、1988

### 普遍知識ベース

```
{Heichinro, {<<pay, money, before, 0>>}, 1}
{Chez_Ino, {<<serve, dish, yourself, 0>>}, 1}
{Sabattini, {<<put, lap, in, trash, 0>>}, 1}
{Chez_Ino, {<<eat, food, on, table, 1>>}, 0}
{Chez_Ino, {<<pay, service, 1>>}, 1}
```

### 状況依存知識ベース

```
{Mac, {<<pay, money, before, 1>>}}
{KFC, {<<serve, dish, yourself, 1>>}}
{Lott, {<<put, lap, in, trash, 1>>}}
{Mr_D, {<<pay, service, 0>>}}
```

### 状況データベース

```
{<<pay, money, before, 0>>, {Mac, Lott, Mr_D, KFC, Moss}}
{<<serve, dish, yourself, 0>>, {Mac, KFC, Lott, J&B}}
{<<put, lap, in, trash, 0>>, {Mac, KFC, Lott, Hokka}}
{<<pay, service, 1>>, {Mac, KFC, Lott, Mr_D, J&B}}
```

図2 状況の再構成前の知識ベース

### 普遍知識ベース

```
{SB1, {<<pay, money, before, 0>>}, 1}
{SB1, {<<serve, dish, yourself, 0>>}, 1}
{SB1, {<<put, lap, in, trash, 0>>}, 1}
{SB1, {<<eat, food, on, table, 1>>}, 0}
{SB1, {<<pay, service, 1>>}, 1}
```

### 状況依存知識ベース

```
{SA1, {<<pay, money, before, 1>>}}
{SA1, {<<serve, dish, yourself, 1>>}}
{SA1, {<<put, lap, in, trash, 1>>}}
{SA1, {<<pay, service, 0>>}}
```

### 状況データベース

```
{<<pay, money, before, 0>>, {SA1, Moss}}
{<<serve, dish, yourself, 0>>, {SA0, J&B}}
{<<put, lap, in, trash, 0>>, {SA0, Hokka}}
{<<pay, service, 1>>, {SA1, J&B}}
```

### 状況管理ベース

```
{SA0, {<<pay, money, before, 1>>, <<serve, dish, yourself, 1>>, <<put, lap, in, trash, 1>>, <<pay, service, 0>>}, {Mac, Lott, KFC}}
{SB0, {<<pay, money, before, 0>>, <<serve, dish, yourself, 0>>, <<put, lap, in, trash, 0>>, <<eat, food, on, table, 1>>, <<pay, service, 1>>}, {Heichinro, Chez_Ino, Sabattini}}
{SA1, {<<pay, money, before, 1>>, <<pay, service, 0>>}, {SA0, Mr_D}}
{SB1, {<<pay, money, before, 0>>, <<serve, dish, yourself, 0>>, <<put, lap, in, trash, 0>>, <<eat, food, on, table, 1>>}, {SB0}}
```

### 状況インデックス

```
{Mac, {SA0}}
{Lott, {SA0}}
{KFC, {SA0}}
{Heichinro, {SB0}}
{Chez_Ino, {SB0}}
{Sabattini, {SB0}}
{SA0, {SA1}}
{Mr_D, {SA1}}
{SB0, {SB1}}
```

図3 状況の再構成後の知識ベース