

事例に基づく推論のための事例ベース構築方法*

4C-4

前田 茂

新世代コンピュータ技術開発機構

1 はじめに

現在、知識処理、特に、エキスパートシステムなどの分野では、比較的強力なルールベースと比較的単純な推論機構を備えたシステムが一般的である。けれども、推論の柔軟性や知識獲得の面で、多くの問題を抱えている。

これに対して近年、事例に基づく推論 (Case-Based Reasoning) [Kolodner 87] が注目を集めるようになってきた。事例に基づく推論とは、従来の一般的なルールを用いる推論に対して、過去の経験や問題解決過程を直接利用して推論を行うものである。事例に基づく推論の特長は以下の通りである。

1. 一般的なルールを獲得するよりも直接的な事例を獲得する方が容易であるため、知識獲得の問題が緩和できる
2. ルールの連鎖ではなく、過去の事例の修正適用により問題解決をはかるので、推論の効率化がはかれる
3. 不完全な知識のもとでも、妥当な解を得ることができる

以上の点から、事例に基づく推論は、過去の事例が豊富に存在するが、一般的な問題解決方法が存在しなかったり、問題解決に対する知識が十分ではなかったりする領域—特に設計問題などの合成型問題—に有効である。

以下では、事例に基づく推論を簡単に概観し、本稿の要点となる事例ベースの構築方法について述べる。

2 事例に基づく推論

事例に基づく推論とは、以下のような処理を行うものである。

1. 問題が与えられた時に、それと類似の過去の事例を事例ベースから検索する
2. 検索された事例を現在の問題に適応するように修正する
3. 得られた解が問題解決に成功した時は、これを新しい事例として事例ベースに蓄える。失敗した時は、失敗事例は次回からの失敗の回避のために蓄えられ、次の解を生成する。

ここで、重要なのが、事例ベースからの類似事例の効率の良い検索、事例の適切な修正、事例の適用に失敗した場

合の対策である。本稿では、これらの点について有効な事例ベースの構築方法について提案し、また、本方法を用いた推論方式についても触れる。

3 事例ベースの記憶

事例ベースに記憶される事例は属性、属性値、その間の関係の三つ組の集合で表されるものとする。属性の集合を $P = \{p_1, \dots, p_n\}$ 、属性値の集合を $V = \{v_1, \dots, v_n\}$ 、それぞれの間の関係の集合を $R = \{r_1(p_1, v_1), \dots, r_n(p_n, v_n)\}$ とすると、事例は $Case(P, V, R)$ で表される。

次に、任意個の関係 $\{r_i, \dots, r_m\}$ の集合を概念 $C(r_i, \dots, r_m)$ と呼ぶことにする。さらに、事例ベース中の事例に共通に表れる概念に対してカウントを付加する。カウントは、その概念を含む事例の個数である。これは、事例ベース中の事例間に表れる関係 (の集合) の抽象度を表し、この値が大きいほど一般的な概念 (上位概念) であると考えることができる。このカウントを自動的に付けることにより概念の階層化を自動的に行うことができる。

事例ベース中では、事例はカウントによってあらかじめソートされたカウント付き概念へのインデックスの並びとして表される。個々の事例は概念に対するインデックスを持つ。これは、 $Case(C_1, \dots, C_n)$ と表すことができる。したがって、重複する概念を個々の事例が持つ必要はなく、記憶効率が改善される。また、その事例に特徴的な関係、概念をカウントを用いて簡単に検出することができる。

事例ベースでは、事例は逐次的に蓄積される。なぜなら、ある問題に対して類似事例を修正適用された結果は、次からの問題解決のために事例ベースに新しい事例として蓄えられるからである。したがって、事例ベースの逐次的な再構築が重要である。

新しい事例が事例ベースに加わった場合の処理は次のように行う。

1. 事例に含まれる関係にマッチする事例ベース中の関係を検索し、各関係に属する概念とそのカウントを得る。そのカウントにしたがって事例中の関係をソートする。
ここで、事例ベース中の関係 $r_i(p_i, v_i)$ と事例中の関係 $r_j(p_j, v_j)$ の間に半順序関係 $r_i \succeq r_j$ (or $r_i \preceq r_j$) が与えられる時、関係 r_i と関係 r_j はマッチしたと見なし、概念中の $r_j(r_i)$ を $r_i(r_j)$ で置き換える。このことで、陽に表されている関係間の上下関係を積極的に利用した、柔軟なマッチングを行うことができる。
2. 事例中の対応する概念が同じ関係の集合を同一の概念 C_i の構成要素と見なし、事例ベース中の概念の再構築を行う。

*Construction method of Case-Base for Case-Based Reasoning
Shigeru MAEDA
Institute for New Generation Computer Technology

- (a) C_i が事例ベース中の概念 C_b と同一の時は、単に、 C_b のカウントを1増加させる。
- (b) C_i が C_b の概念の部分集合である時は、 C_i を新しい上位概念として C_b から分離し、 C_i のカウントを1増加させ、 C_b を指していた事例の全てに対し、 C_i を指すようにする。
- (c) C_i が事例ベース中に存在しない時は、カウント1を与える。

以上のステップを事例中の対応する概念 C_i がなくなるまで繰り返す。

3. 事例を各概念のインデックスの並びとして記憶する。

これにより、事例ベースに事例が蓄積されていくにしたがって、事例ベースは概念のカウントによる階層化が自動的になされることになる。

4 類似事例検出モデル

ここでは、ある問題が与えられた時に、人間が自分の過去の体験の中からその問題解決に最も適切な類似事例を検索する方法について考察する。

人間は、記憶からある特定の事例を検索する場合に、その事例に関する細かい属性までをすべて並べ挙げたりしない [Kolodner 83]。自然に、検索すべき事例の特徴的な要素を無意識に抽出し、それによって検索を行う。このとき、ある大きな概念によって探索空間のしぼりこみを行う。つまり、特徴的な属性による連想的検索と、比較的大きな概念による探索空間の絞り込みによる両方向の検索を行っているものと考えられる。これにより、全解探索を回避し、効率的な検索を行うことができる。

5 類似事例の検索

4章で述べた様に、人間は記憶を検索する時に必ず、検索する事例に特徴的な要素を用いる。ここでは、この考えに基づき、前述の事例ベースから類似事例を効率的に検索する方法について述べる。

今、検索したい入力事例が関係の集合 $\{r_1, \dots, r_n\}$ で表されるものとする。この中には、上で述べたことより、必ず入力事例に特徴的な関係を含むものと仮定できる。しかし、一般に入力事例単独では、あらかじめ特徴的な関係を決定することが困難な場合が多い。したがって、まず最初に入力事例に特徴的な関係を検出することが重要である。ところで、事例ベースは、概念の階層構造として表されているので、ある事例に特徴的な関係と、一般的な概念を容易に判別することができる。以上のことから、4章のモデルにしたがって、以下のように類似事例の検索を行う。

1. 入力事例に含まれる関係のカウントを全て0にする。
2. 3章の1.の処理を行い、入力事例の各関係に対応する概念とそのカウントを得る。
3. 得られたカウントが1の関係の集合 $\{r_1, \dots, r_m\}$ のそれぞれの要素に対して、その関係を含む事例を得る。カウントが1の関係は、入力事例に特徴的な（他のただ一つの事例と共有する）関係であり、これにより検索の効率化がはかれる。

4. 得られた類似事例が複数の場合は、カウントが最大な概念を用いて検索のしぼり込みを行う。
5. 上の2つのステップを単一の類似事例が求まるか、入力事例に含まれる関係がなくなるまでそれぞれカウントを増加、減少させながら繰り返す。

以上の処理を行うことにより、入力事例に含まれる特徴的な概念を抽出し、これを用いて全解探索することなしに効率的に類似事例の検索を行うことができる。

6 類似事例の修正と失敗からの回復

本方法を用いて、事例に基づく推論を行う場合、類似事例の修正は次のように行う。

1. 入力事例でカウントが0（入力事例のみに存在する）の関係の集合 $R = \{r_1, \dots, r_n\}$ を得る。
2. R の各要素に対して、類似事例に領域知識を用いて修正を加える。
3. 修正を加えた結果、修正が事例に大幅な影響を与える場合は、制約伝播の考えを用いて修正が必要な個所を特定し、修正を加える。

また、事例の適用に失敗した場合、次の解を生成する方法には、次の方法が考えられる。

1. 入力事例でカウントが1の関係を持つ他の事例を選択する。
2. 入力事例でカウントが0を持つ関係のあるものを、類似事例のカウントが1を持つものと置換した事例を選択する。
3. 入力事例の上位にある概念を含む事例を選択する。

いずれの場合にも、概念とそのカウントを用いて効率的に処理を行うことができる。

7 まとめ

以上、事例に基づく推論に適した事例ベースの構築法を提案した。本方法の特長は、概念の自動的な抽出と階層化である。これにより、記憶効率の向上、類似事例の検索の効率化、柔軟な推論、失敗からの回復をはかることができる。また、関係間の順序関係に関する知識を与えることにより、従来の単純なパターンマッチに比べてより柔軟な類似事例の検索を行うことができる。

参考文献

- [Kolodner 83] Kolodner, Janet L., *Maintaining Organization in a Dynamic Long-Term Memory*, Cognitive Science, 7(4), pp.243-280, 1983.
- [Kolodner 87] Kolodner, Janet L., *Extending Problem Solving Capabilities Through Case-Based Inference*, Proceedings of the 4th Annual International Machine Learning Workshop, 1987.