

CBR を用いた Product Configuration - Online Shopping への適用 -

稲越 宏弥 岡本 青史 太田 唯子 湯上 伸弘
(株) 富士通研究所

1. はじめに

本論文では、我々が提案する CBR と制約充足を統合した問題解決手法を、パソコンのオンラインショッピングの問題に適用し、本手法の有効性を示す。本システムは類似事例を使用することで、不十分なユーザーのニーズを補完することができ、以下の従来手法の問題点を解消している。

- 価格順ソートなどは、ユーザーのニーズとは無関係で、販売実績と無関係に人気のない商品を勧める可能性がある。
- ユーザーの要求の与えかたによって、商品の候補数が適量で多く、意思決定のための情報を十分に提供できない。
- そのため、希望に合致した商品を決めるまでに、試行錯誤しながら何度もインタラク션을繰り返す必要があり、入力への負荷が大きい。

はじめに、本システムが取り扱うオンラインショッピングの問題を記述する。次に本システムが備える特徴的な機能について説明を行う。最後に今後の課題について検討する。

2. 問題の記述

パソコンのオンラインショッピングの場合、購入目的、筐体、CPU、OS および機種の種類などが変数となる。このような変数 z_i , $i = 1, \dots, n$ からなるベクトル空間を Z 用いて次のように記述する。

各変数にはそれぞれ定義域が定義されている。例えば変数 OS の定義域は {W98, WNT, W2K} である。また、パソコンの仕様を定義するための制約条件は、属性値の可能な組合せを定義する。例えば、機種 C4/665 は、CPU が P667、筐体は CT (コンパクトタワー)、OS は W98 と W2K が選択できるといった、属性値の組合せである。

次に、事例ベースについて説明する。事例ベース K は過去の販売実績を、 N 個の事例 c_j として保持する。

各事例 c_j は、各属性 z_i の値 v_{ji} のベクトルとして表現できる。

$$K = \{c_j | j = 1, \dots, N\},$$

$$c_j = (v_{j1}, \dots, v_{jn})$$

事例ベースの例を表 1 に示す。表 1 中の属性「目的」は、購入目的を意味しており、値 I はインターネット向け、および値 R は研究開発向けを表わしている。このような属性は、パソコンの仕様とは直接関係がないため、制約充足器では使用されない。このような属性を定性的属性と呼ぶ。

K	目的	筐体	CPU	OS	機種	...
c_1	I	CT	P667	W98	C4/665	
c_2	I	CT	C533	W98	C4/535c	
c_3	I	NA4	C500	W98	NE4/50C	
c_4	R	MT	A700	WNT	M4/707	
:						

表 1: 事例ベース

3. 商品ランキング

本システムでは、定性的属性を含むユーザーの要求に対する類似事例を検索する。定性的属性は類似事例を介して属性値の分布に影響を与えるので、属性値の分布を目的関数として定量化することにより、ユーザーのニーズを補完したり推測するのに応用できる。そのため価格などの固定的な基準による手法より、ユーザーの希望に近いランキングが可能である。

以下では具体的な処理の手順を説明する。

3.1. 要求入力

要求を空間 Z 上のベクトル q で表わす。ユーザーは全ての属性に属性値を指定する必要がなく、不十分な要求をシステムは処理することができる。

$$q = (u_1, \dots, u_n) \in Z$$

例えば、インターネット用にコンパクトタワーのパソコンの購入をしたいというように、目的と筐体だけを要求として指定することができる。

3.2. 類似事例検索

本システムでは、要求 q と事例 c_j の間の距離関数 $d(q, c_j)$ を属性 z_i に重み w_i をつけて、次のように定義する。

$$d(q, c_j) = \sum_i w_i \delta_{ji},$$
$$\delta_{ji} = \begin{cases} 0 & \text{if } u_i = v_{ji} \\ 1 & \text{if } u_i \neq v_{ji} \end{cases}$$

この距離関数 $d(q, c_j)$ の元で、閾値 ε 以内にある事例を類似事例とし、類似事例集合を \hat{K} と表わす。

3.3. 目的関数生成

類似事例集合 \hat{K} から、制約充足最適化において解のランキングの評価基準として使う目的関数 f_q を生成する。本システムでは、類似事例集合 \hat{K} に対して各属性 z_i の出現確率 $Prob(z_i = y_{ji})$ を用いて目的関数を生成している。制約充足解 $y_j = (y_{j1}, \dots, y_{jn})$ の目的関数の値は次のようになる。

$$f_q(y_j) = \sum_{i=1}^n Prob(z_i = y_{ji}) \cdot I(z_i = y_{ji}),$$
$$I(z_i = y_{ji}) = \begin{cases} 1 & \text{if } z_i = y_{ji} \\ 0 & \text{otherwise} \end{cases}$$

3.4. 制約充足最適化

上述のように生成された目的関数 f_q を用いて、制約充足解 y_j のランキングを行う。ランキングは f_q の値が大きいものから数件を出力する。

制約充足解を求めるアルゴリズムは高速化の為に Arc Consistency [2] を使用している。

4. 代替商品の推薦

3節の手順によって得られる商品がない場合に、以下に述べるような要求の条件を緩和することにより、代替商品を推薦する。

要求 q で属性値が指定された属性のうち1つだけを選んで値を未確定とするように条件を緩和した要求を生成する。それぞれの要求に対する制約充足解を、目的関数 f_q によってランキングした結果を、代替商品として提示する。

5. 要求入力ナビゲーション

3節の手順で得た商品ランキングの結果、ユーザが満足する解が得られない場合、要求に新たに条件を追加する作業を繰り返し、商品の仕様を詳細化する必要がある。このとき、現在の要求に対する類似事例集合 \hat{K} に基づいて、解の絞り込みに有効な設問をユーザに示す。

本システムでは、新たに追加する要求を選択する基準としてインフォメーションゲイン [1] を用いる。

要求 q の属性値 z_s が未確定だったとする。現在の類似事例集合 \hat{K} の中で、属性 z_t が値 v_{tj} を取る確率を p_j 、属性 z_s が値 u_{sj} を取る確率を q_i とする。また、 $z_s = u_{si}$ としたとき、 $z_{tj} = v_{tj}$ となる確率を、 r_{ij} とする。

$$Info_s(t) = \sum_j p_j \log p_j - \sum_i q_i \sum_j r_{ij} \log r_{ij}$$
$$p_j = Prob(z_t = v_{tj}),$$
$$q_i = Prob(z_s = u_{sj}),$$
$$r_{ij} = Prob(z_t = v_{tj} | z_s = u_{si}).$$

出力する各属性 z_t について上記の和を求める。

$$Info_s = \sum_t Info_s(t)$$

この $Info_s$ が大きな値を取る属性 z_s を解の絞り込みに有効な属性としてユーザに提示する。ユーザはこの情報を用いることで、効率的な解の絞り込みを行うことができる。

6. 今後の課題

本手法を他分野へ適用することなどを通じて、本手法の有効性を検討したり、従来手法と比較した場合の有効性を測る客観的評価尺度についても今後の研究が必要である。

また、事例を時系列データとして扱うことによって、時間を勘案した商品の推薦を行うことを検討している。

参考文献

- [1] Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.
- [2] Tsang, E. *Foundations of Constraint Satisfaction*. Harcourt Brace & Company., 1993.