

マルチキャラクタを用いた競争型情報推薦システム

Competitive Information Recommendation System Using Multi-Characters

阪本 俊樹*1
Toshiki Sakamoto

回り道 康弘*2
Yasuhiro Mawarimichi

北村 泰彦*1
Yasuhiko Kitamura

辰巳 昭治*1
Shoji Tatsumi

*1 大阪市立大学大学院 工学研究科
Graduate School of Engineering, Osaka City University

*2 NTTアドバンステクノロジー株式会社
NTT Advanced Technology Corporation

Recommendation systems draw the attention of practitioners in B-to-C electronic commerce. In an independent recommendation system such as used in amazon.com, a user cannot compare the information with that in other information sources. In a broker-mediated recommendation system, the broker takes the initiative of recommendation, so the information provider cannot recommend its information directly to the user.

In this paper, we propose a competitive information recommendation system in which multi character agents recommend their information competitively, and apply it to the field of restaurant recommendation. Each agent recommends restaurants from its own point of view, the user tells good or bad about them. In our competitive information recommendation system, a user can compare recommended information, and the information providers can recommend their information directly to the user. Consequently, this system has an advantage for both of the user and the information provider.

1. はじめに

近年、B-to-Cの電子商取引を支援する手段として推薦システムが注目されている。従来の情報推薦システムには、amazon.com*1のように情報提供者が単一の情報源からユーザへ推薦を行う独立型情報推薦システムや、DealTime*2のようにブローカが複数の情報源から情報収集して推薦を行うブローカ型情報推薦システムがある[2]。

独立型情報推薦システムでは、情報源が単一であるため、ユーザは他の情報源との比較はできない。これに対しブローカ型情報推薦システムでは、ユーザは複数の情報源の情報推薦をブローカから受けることができ複数の情報を比較できる。しかしブローカ型情報推薦システムでは、情報提供者にあった情報推薦の主導権がブローカに握られているため、情報提供者が持つ推薦基準が情報推薦に反映されず、情報提供者にとって十分なメリットがあるとは言えない。

本研究では、複数の情報源がキャラクタエージェントを介して直接ユーザと相互作用を行い、競争的にユーザへ推薦を行う、競争型情報推薦システムを提案し、レストラン推薦に応用する。また競争型情報推薦システムにおいて、エージェントが非合理的な提案を行わないことを保証する合理的提案アルゴリズムを提案する。競争型情報推薦システムでは、情報提供者は独自にエージェントを開発することで独自の基準で推薦を行うことができ、またユーザは複数の情報源からの情報を比較することができる。このことから競争型情報推薦システムは、情報提供者とユーザの双方にメリットのあるシステムといえる。

2. 競争型情報推薦システム

2.1 システム構成

競争型情報推薦システムは、推薦ブラックボードとエージェント、ブラウザとキャラクタインタフェースによって構成される。システム構成を図1に示す。

連絡先: 阪本俊樹, 大阪市立大学大学院工学研究科, 大阪市住吉区杉本 3-3-138, Tel:06-6605-2778, Fax:06-6605-2778, toshi@kdel.info.eng.osaka-cu.ac.jp

*1 <http://www.amazon.com/>

*2 <http://www.dealtime.co.jp/>

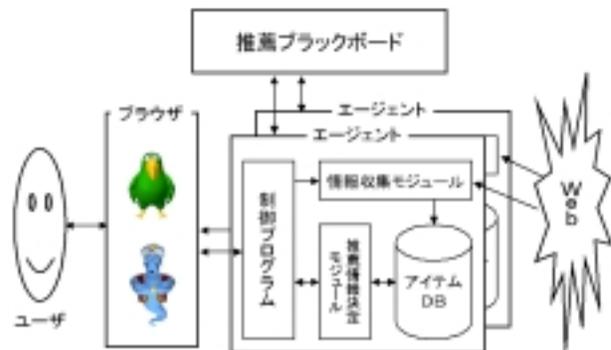


図 1: 競争型情報推薦システムの構成

このシステムでは各エージェントは競争的に情報推薦を行う。競争的とは、他のエージェントが推薦した情報よりも優れた情報を推薦するということである。このためには、各エージェントは他のエージェントが推薦した情報を知る必要があり、その情報は推薦ブラックボードを参照することで得られる。推薦ブラックボードはエージェントが各情報を共有する場である。また、エージェントは関連付けられた情報源から情報集を行いユーザへ情報推薦を行う。

2.2 推薦ブラックボード

(1) エージェント間で共有される情報の保持

推薦ブラックボードでは、エージェント間で共有される情報を保持する。保持する情報とは次の3つである。

ユーザの要求 ユーザが自分の得たい内容をシステムに対して入力したものである。例えばレストラン推薦においては、「難波でスパゲッティを食べたい」というのがユーザの要求である。

推薦アイテム エージェントの推薦した情報の内容である。例えばレストラン推薦においては、レストラン名と、レストランの価値を決定付ける構成要素である、平均予算、最

寄駅からの距離、などの情報をユーザへ推薦する単位ごとにまとめたものである。レストランの価値を決定付ける構成要素を属性と呼び、それぞれのとる値を属性値と呼ぶ。

ユーザの評価 推薦アイテムに対してユーザが自分の要求に合うものかどうかを判断したものである。このシステムでは、ユーザは推薦アイテムに対して Good か NoGood の評価を下す。

2.3 ブラウザとキャラクタインタフェース

ブラウザとキャラクタインタフェースの制御については、各エージェントによって行われる。ブラウザにはエージェントにより推薦された情報を表示し、その情報についてのユーザとの相互作用をキャラクタインタフェースにより行う。ブラウザやキャラクタインタフェースを介する情報は、同時に推薦ブラックボードに書き込まれ、エージェント間で共有される。

また、キャラクタインタフェースは情報推薦の視覚化に役立っている。つまり文字だけの情報推薦に比べて、キャラクタが登場して発言することにより、情報推薦の過程をユーザは簡単に見て取ることができる。またキャラクタの動きや発言内容により、ユーザの興味を引くことにも役立つ。

2.4 エージェント

(1) 情報収集

各エージェントは、関連付けられた情報源からユーザの要求を満たす情報を収集してアイテムデータベースの作成を行う。Webからの情報収集には、MetaCommander[1]とTemplateFilterを用いる。

MetaCommanderとは、スクリプトによりWebからの情報を取得するためのツールである。これにより各エージェントはユーザの要求から独自のスクリプトを生成し、Webページのダウンロードを行う。

TemplateFilterとはフォーマットの決まったページから情報抽出する際に、あらかじめ用意しておいたテンプレートファイルと比較することで、指定した部分の情報のみを抜き出すことのできるツールである。これにより各エージェントはダウンロードしたWebページから必要な情報を抽出する。

(2) 推薦アイテムの決定

情報収集を終えたエージェントは、個別の戦略に基づいて推薦アイテムの決定を行う。

(3) ブラウザとキャラクタインタフェースの制御

エージェントは推薦情報をユーザに表示するため、ブラウザを制御し画面の切り替えを行う必要がある。1つのエージェントが一度に推薦できる情報は1つだけであり、新たな情報が推薦されるたびに画面は切り替えられていくことになる。

エージェントはブラウザの制御をするとともに、キャラクタインタフェースの制御も行う。これはキャラクタに推薦情報についてのコメントを発言させるものである。また、推薦情報についてのユーザの評価を得る必要がある場合、キャラクタはユーザへ評価を問う。

2.5 処理の流れ

次にこのシステムにおける情報推薦の流れを示す。

1. ユーザからの要求を受け取り、それを推薦ブラックボードに書き込む
2. 各エージェントは推薦ブラックボードを参照し、ユーザの要求を獲得する

3. 各エージェントは関連付けられた情報源から情報収集を行う
4. いずれかのエージェントが戦略に基づいてアイテムの推薦を行う
5. 必要があれば推薦アイテムに対してユーザが評価を行う
6. 各エージェントは推薦ブラックボードを参照し、推薦アイテムの内容とそれに対するユーザの評価を獲得する
7. 4へもどる

なお、このシステムが終了するのはいずれかのエージェントがアイテムを推薦することができなくなったときである。

3. 合理的提案アルゴリズム

競争型情報推薦システムでは、情報推薦を行うエージェントが複数存在し、それらが自律的に提案を行うことから非合理的な提案が行われる可能性がある。そこで、それぞれのエージェントが提案内容の合理性を判断し、合理的な提案を行うための合理的提案アルゴリズムについて説明する。合理的な提案とは、エージェントが提案を行う際に、ユーザにとって明らかに不要なアイテムの提案を行わないということである。

合理的提案アルゴリズムでは、アイテムを合理性に基づく複数の領域へ分類し、アイテムが属する領域を参照した上で、提案を行う。ただし、提案を行う度に領域は変化するので、適切に更新して行く必要がある。

ここでは、2属性アイテム問題を扱っている。また問題を単純化するために、以下の2点を仮定する。

- 属性値はすべて正の値をとり、属性値が低いほどそのアイテムの効用(価値)が高い
- ユーザは提案に対して、GoodとNoGoodの評価のみを行う。

3.1 2属性アイテム問題における合理的提案手法

(1) 3領域の定義

アイテムを分類する3領域を以下のように定義する。

領域 R : 受理可能なアイテムが属する領域

領域 I : 受理不可能なアイテムが属する領域

領域 U : 受理可能性が不明なアイテムが属する領域

ここでの受理可能なアイテムとは、推薦アイテムがすでに推薦されたアイテムと比較したとき、すべての属性において優れたアイテムのことである。また受理不可能とは、すべての属性において劣ったアイテムのことである。例えば、レストラン情報においてすでに予算 = 2000円、距離 = 150mのアイテムが推薦されているとすると、予算 = 1500円、距離 = 100mのアイテムは受理可能なアイテムであり、領域 R に属する。予算 = 3000円、距離 = 200mのアイテムは受理不可能なアイテムであり領域 I に属する。また予算 = 1500円、距離 = 200mアイテムは受理可能性が不明なアイテムであり領域 U に属する。

領域 I のアイテムはユーザにとって明らかに不要であるため、提案は R と U の2領域から行われる。また R からの提案はユーザにとって明らかに有用であるため、ユーザへ評価を問う必要がない。

(2) 合理的提案アルゴリズム

以下に、合理的提案アルゴリズムを示す。この合理的提案アルゴリズムは、2.5節の4.から7.に相当する。

アイテム P が2属性を有し、それぞれの属性値が x, y であるとき、アイテム P を $\langle x, y \rangle$ で表す。またユーザが Good の評価を下したものを受理アイテムと呼び、そのうち領域の境界となるアイテムを境界アイテムと呼ぶ。

なお、すでに $n-1$ 個のアイテムが提案されている時の領域をそれぞれ $R_{n-1}, I_{n-1}, U_{n-1}$ とする。 $P_{A_1}, P_{A_2}, \dots, P_{A_m}$ は境界アイテム、 m は境界アイテムの個数、また新たに提案するアイテムを $P_n = \langle x_n, y_n \rangle$ とする。

1. 領域 R 又は U から戦略に基づいてアイテムを選択し、提案を行う。
2. 領域 U からの提案についてはユーザに評価を問う。
3. 領域状態を更新する。

(a) 領域 I の更新

$$I_n = I_{n-1} \cup \{ \langle x, y \rangle \mid (x \geq x_n) \wedge (y \geq y_n) \} - \{ \langle x_n, y_n \rangle \}$$

(b) 領域 R の更新

- i. $P_n \in R_{n-1}$

$$R_n = \{ \langle x, y \rangle \mid (x \leq x_{A_1}) \wedge (y \leq y_{A_1}) \} \cup \dots \cup \{ \langle x, y \rangle \mid (x \leq x_{A_m}) \wedge (y \leq y_{A_m}) \}$$

- ii. $P_n \in U_{n-1}$

A. 利用者の評価が「Good」

$$R_n = R_{n-1} \cup \{ \langle x, y \rangle \mid (x \leq x_n) \wedge (y \leq y_n) \}$$

B. 利用者の評価が「NoGood」

$$R_n = R_{n-1}$$

(c) 領域 U の更新

$$U_n = \overline{R_n \cup I_n}$$

4. 1. に戻る。

(3) 領域の推移

アイテムの提案が行われた時の領域の推移についてレストラン推薦を例にして考える。レストラン情報は x を予算(円), y を距離(m)とし $\langle x, y \rangle$ で表すものとする。

領域 R からの提案

すでに、 $P_0 = \langle 2000, 150 \rangle$ のレストランが提案されており P_0 が境界アイテムであるとき、 $P_1 = \langle 1200, 60 \rangle$ のレストランを提案するときの領域の推移を考える。 P_1 は P_0 と比べて、2属性の両方ともが優れているので、領域 R に属している。よってユーザへ評価を問う必要はない。 P_1 を提案すると領域は図2のようになる。 P_1 の提案により、 $(x \geq 1200)$ かつ $(y \geq 60)$ の領域が領域 I となり、 $(x \leq 1200)$ かつ $(y \leq 60)$ の領域が領域 R となり、それ以外は領域 U となる。

領域 U からの提案

すでに、 $P_2 = \langle 1200, 150 \rangle$ のレストランが提案されており P_2 が境界アイテムであるとき、 $P_3 = \langle 2400, 90 \rangle$ のレストランを提案するときの領域の推移を考える。 P_3 は P_2 と比べて予算は劣っているが距離は優れている。よって P_3 は領域 U に属している。ユーザの評価が Good の場合の領域は図3のようになる。 P_3 の提案により、 $(x \geq 2400)$ かつ $(y \geq 90)$ の領域が領域 I に、 $(x \leq 2400)$ かつ $(y \leq 90)$ の領域が領域 R に加えられる。

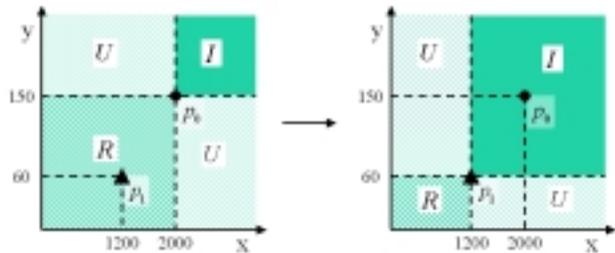


図2: 領域 R からの提案による領域の推移

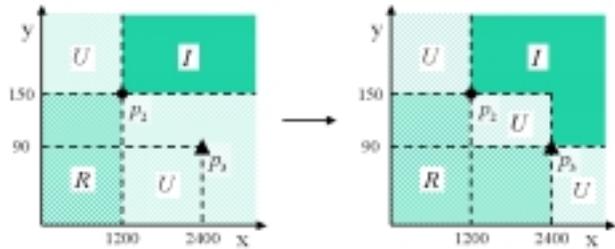


図3: 領域 U からの提案 (ユーザの評価が Good の場合)

ユーザの評価が NoGood であった場合の領域は図4のようになる。 P_3 がユーザに受理されなかったためこの場合の領域 R の変化はない。また、領域 I の推移は、ユーザの評価が Good であった場合と同じである。

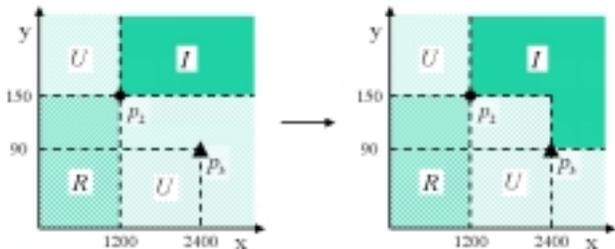


図4: 領域 U からの提案 (ユーザの評価が NoGood の場合)

4. レストラン推薦システム

競争型情報推薦システムの例題として、レストラン推薦システムを示す。このシステムでは、ユーザが「何処で何を食いたい」という要求を送ることにより、2体のエージェントが競争しながら次々と情報推薦を行うものである。

4.1 設定

レストラン推薦システムにおいて、エージェントは2体とし、それぞれの情報源を Yahoo!グルメ^{*3}と関西ウォーカー^{*4}の2つのサイトとする。エージェントが推薦するレストラン情報

*3 <http://gourmet.yahoo.co.jp/>

*4 <http://www.walkerplus.com/kansai/gourmet/>

の属性を、「平均予算」と「最寄駅からの距離」の2属性とし、それぞれ数値の低いものほど価値が高いものとする。以下、2つの属性の属性名を「予算」と「距離」と呼ぶ。また2体のエージェントはいずれも、受理不可能でないアイテムの中で、予算の最も高いレストランを推薦する、という戦略をとる。

4.2 動作例

次にレストラン推薦システムの動作例を示す。また、動作画面の一例を図5に示す。

1. ユーザは「心斎橋でカレーを食べたい」という要求を送る。
2. 関西ウォーカー・エージェントは「予算 3000 円, 距離 91 m」のレストランを推薦する。
3. ユーザは Good を返答する。
4. Yahoo! グルメ・エージェントは「予算 1000 円, 距離 336 m」のレストランを推薦する。このレストランは1のレストランと比較して、予算は優れているが距離は劣っている。よってユーザへの問い合わせを行う。
5. ユーザは NoGood を返答する。
6. ユーザの返答が NoGood であったので, Yahoo! グルメ・エージェントは先ほどのレストランよりも距離の短い「予算 1000 円, 距離 209 m」のレストランを推薦する。ユーザへの問い合わせを行う。
7. ユーザは Good を返答する。
8. 関西ウォーカー・エージェントは「予算 500 円, 距離 952 m」のレストランを推薦する。このレストランは1のレストラン, 5のレストランと比較して、予算は優れているが距離は劣っている。よってユーザへの問い合わせを行う。
9. ユーザは Good を返答する。
10. Yahoo! グルメ・エージェントは「予算 800 円, 距離 170 m」のレストランを推薦する。このレストランは5のレストランと比較して、予算・距離の両方が優れている。よってユーザは Good を返答すると予想できるので、ユーザへの問い合わせは行わない。
11. 関西ウォーカー・エージェントの推薦するレストランがなくなった。これによりシステムは終了する。



図 5: レストラン推薦システムの動作画面の一例

5. まとめと今後の課題

複数の情報提供者がそれぞれのエージェントを介してユーザと相互作用を行うことにより、競争的に情報を推薦する競争

型情報推薦システムを提案し、レストラン推薦に応用した。競争型情報推薦システムは、ユーザは複数の情報を比較でき、また情報提供者は独自のエージェントを用いることにより自分の推薦基準を反映することができるため、ユーザと情報提供者の双方にメリットのあるシステムである。

また競争型情報推薦システムでは、複数のエージェントが自律的に情報推薦を行うため、非合理的な提案が行われる可能性が生じた。そこでエージェントがアイテムの合理性を判断して合理的な提案を行うための合理的提案アルゴリズムを提案した。

今後の課題として、以下の3点について考える必要がある。

情報推薦に用いる属性を多属性にする レストラン推薦システムにおいては、属性は2つであったがこれを3属性以上にすることを考える必要がある。属性数を増やすことにより、必然的に領域 U からの提案が多くなり、それにもなってユーザへの問い合わせを行う回数が増加し、またユーザとのより複雑なインタラクションが発生することになり、それへの対応が問題となる。

エージェントの分散化 現在は1台のサーバで2体のエージェントを動かしているが、これを複数のサーバで個別にエージェントを動作させ、それらを結びつけての情報推薦の実現を目指す。これによりインターネット上で様々なエージェントによる競争型情報推薦が可能になる。

協力型情報推薦システムの開発 各エージェントが互いに協力して情報推薦を行う協力型情報推薦について考える必要がある。この場合、各エージェントの持つ情報の属性が同じものについては、ユーザの要求する制約に従って協力して情報推薦を行うことになる。例えば金額に制約がある場合、片方のエージェントが金額の高い情報を推薦するなら、他方のエージェントは推薦する情報の金額を低くするといったことが考えられる。

参考文献

- [1] 北村泰彦, 野崎哲也, 辰巳昭治, "スクリプトに基づくWWW情報統合支援システムとゲノムデータベースへの応用", 電子情報通信学会論文誌, J81-D-1(5):451-459, 1998.
- [2] 北村泰彦, "インターネットエージェント", bit, 31(2):7-14, 1999.
- [3] Sonia Bergamasch, Domenico Beneventano, "Integration of Information from Multiple Sources of Textual Data", Matthias Llusch(Ed.), Intelligent Information Agents, pp.53-77, Springer Verlag, 1999.