

## 競争型情報推薦システムとその合理的推薦手法

阪本 俊樹<sup>†</sup>      北村 泰彦<sup>††</sup>      辰巳 昭治<sup>†</sup>

A Competitive Information Recommendation System and Its Rational Recommendation Method

Toshiki SAKAMOTO<sup>†</sup>, Yasuhiko KITAMURA<sup>††</sup>, and Shoji TATSUMI<sup>†</sup>

あらまし

現在、インターネット上の多様で膨大な Web 情報の中から利用者が自らの要求を満たす情報を入手するための手段として、検索エンジンが広く利用されている。しかしながら検索エンジンは利用者の要求がキーワードによって明確に表せないような場合には有用であるとはいえない。そこで、システム側から利用者側に興味のあるような情報を提供する手段として情報推薦システムが注目されている。本論文では個別の情報源を持つ複数のエージェントがキャラクタを介して利用者とのインタラクションしながら、競争的に利用者への情報の推薦を行う競争型情報推薦システムを提案する。競争型情報推薦システムでは、各エージェントが単に自律的に推薦を行うだけでは、利用者にとって望ましい情報が推薦されなかったり、利用者がそれを入手するまで多くのやり取りが必要になる場合も考えられる。そこで本論文では、推薦情報から得られる利用者の効用を多属性効用関数で表現し、エージェントが各属性に対する重みを学習しながら、利用者にとって望ましい情報のみを推薦する合理的推薦手法を提案する。さらに、エージェントの情報推薦戦略として、自己の利得を優先する利得優先型、利用者の効用学習を優先する学習優先型の戦略を示し、利得優先型が優れていることを評価実験を通して示す。

キーワード 推薦システム, 競争, マルチエージェント, 多属性効用関数, 合理性

### 1. ま え が き

現在、インターネット上には多様で膨大な量の情報が蓄積されており、それは日々膨張を続けている。この膨大なデータベースから利用者の要求を満たす情報を入手する手段として、検索エンジンをはじめとする情報検索システムがある。情報検索システムでは利用者はあらかじめその要求を何らかの具体的な形式で表現できることが仮定されており、利用者が情報源に対して能動的にアクセスすることが求められている。しかし、情報検索システムでは、利用者が要求表現が明確でなかったり、曖昧であったりした場合、その検索結果は多量のノイズを含んでいたり、必要な情報を含んでいなかったりする、といった問題が生じる。そこでこれとは逆に、システム側が利用者側に情報を推薦

するアプローチとして情報推薦システムがある。

情報推薦システムでは情報検索システムとは対照的に、システム側で利用者の要求を推測する機能が必要となり、必然的に利用者とのインタラクションが重要になる。このような情報推薦システムの具体例として ATA(Automated Travel Assistant) [6] や ExpertClerk [8] があげられる。

ATA は利用者の旅行プランニングを支援するシステムである。ATA では、CCA(candidate/critique agent) が利用者とのインタラクションから利用者の興味モデルを学習していく。CCA は利用者の興味モデルを多属性効用関数の荷重和として表現しており、利用者とのインタラクションから荷重や関数を調整することで真のモデルに近づけている。

ExpertClerk は、人間の店員をまねたキャラクタが商品データベースを用いてインターネット上の買物客を案内するシステムである。ExpertClerk では、キャラクタが買物客と自然言語で対話し、NBA(navigation by asking) と NBP(navigation by proposing) という、2つの案内モードを用いてインタラクションの効

<sup>†</sup> 大阪市立大学大学院工学研究科, 大阪市  
Graduate School of Engineering, Osaka City University, 3-3-138, Sugimoto, Sumiyoshi-ku, Osaka, 558-8585 Japan

<sup>††</sup> 関西学院大学理工学部, 三田市  
School of Science and Technology, Kwansei Gakuin University, 2-1, Gakuen, Sanda-sho, Hyogo, 669-1337 Japan

率化を図っている。NBA は質問に概念的な階層構造をもたせて、それぞれの質問の持つ情報量から情報利得を最大化する質問を行う。また、NBP では対照的な3つのサンプル商品を提示し、それに対する買物客の評価を次の提案に反映しようとしている。

ATA, ExpertClerk は、ともにキャラクタを用い、利用者とのインタラクションを繰り返すことで、利用者が満足するアイテムを提示することを目的とした推薦システムである。ただエージェントは単体であり、複数のエージェントによる情報推薦を目的としたものではない。

複数のエージェントが協調的に推薦を行う協調型情報推薦システムも存在する。これは個別の視点をもつ複数のエージェントが協調的に動作することにより情報推薦をより効果的に行おうとするものである。例えば、同一の商品に対して異なる視点からその商品を推薦することは利用者の商品選択の基準をより早く明確化することにつながる。このようなシステムの具体例に Inhabited Market Place [1] がある。このシステムでは、車の売買を行う複数のキャラクタエージェントが登場し、それぞれの立場から意見を述べる。例えば、あるエージェントは車の最高速度についての意見を述べ、別のエージェントは燃費についての意見を述べる。このような意見を聞き比べながら利用者は漠然と持っていた車への要求を明確にしていく。ただし、このシステムではシナリオに基づいてエージェントが意見を述べるだけであり、利用者からの判断に応じてその動作を変更することのないノンインタラクティブなシステムである。

これに対して、本論文で扱う競争型情報推薦システムは複数のエージェントが自律的に情報の推薦を行い、利用者とのインタラクションしながら、競争的に動作する。競争型情報推薦システムでは、各エージェントは自律的に情報推薦を行うことから、利用者にとって必ずしも望ましくないような情報が推薦されたり、無駄なインタラクションが生じたりするような状況が考えられる。そこで本論文では多属性効用関数に基づき、エージェントが利用者にとって合理的な情報を推薦する合理的推薦手法を提案する。また情報推薦戦略として自己の利得を優先する利得優先型、利用者の効用学習を優先する学習優先型の戦略を示し、利得優先型の方が性能的に優れていることを評価実験を通して示す。

以下、2章で競争型情報推薦システムについて、3章で合理的推薦手法について述べる。また4章ではエー

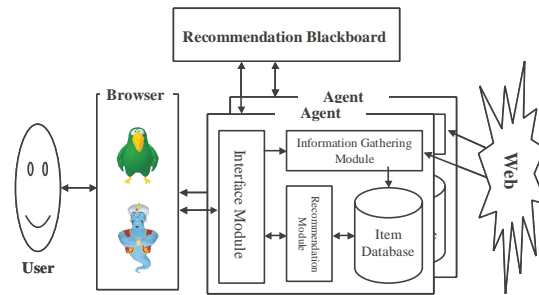


図 1 競争型情報推薦システムの構成

Fig. 1 Architecture of competitive information recommendation system.

ジェントの推薦戦略とその性能を示す。5章では関連事例について述べた後、6章でまとめとする。

## 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 [4] は、図 1 に示すように、複数のエージェント (Agent) とキャラクタインタフェース、推薦ブラックボード (Recommendation Blackboard)、ブラウザ (Browser) から構成されている。

#### 2.1.1 推薦エージェント

エージェントは情報推薦を行う主要な構成要素であり、推薦アイテムに関する情報源からの情報収集、キャラクタインタフェースを介したアイテム推薦を行う。

##### a) 情報収集

エージェントは情報源からの推薦アイテムに関する情報収集を情報収集モジュール (Information Gathering Module) を用いて行う。情報収集モジュールは MetaCommander [3] と Template Filter から構成されている。

MetaCommander はスクリプトを記述することにより、Web ページをダウンロードするためのツールである。MetaCommander では URL で指定される Web ページの単純なダウンロードだけでなく、フォーム形式の入力インタフェースを介する Web ページのダウンロードも可能である。エージェントは利用者の要求からスクリプトを自動的に生成し、Web ページのダウンロードを行う。

Template Filter はフォーマットの決まった Web 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、指定した部分のデータのみを抜き出すツールである。これにより

各エージェントはダウンロードした Web ページから推薦アイテムに必要な情報を抽出する。

例えばレストラン推薦システムの場合、それぞれの推薦エージェントはインターネット上のレストラン情報サイトにアクセスし、その Web ページをダウンロードし、そのページからレストラン推薦に必要な店名、平均予算、住所などの情報を抽出する。

抽出された情報はアイテムデータベース (Item Database) に登録される。

#### b) アイテム推薦

エージェントはインターフェースモジュール (Interface Module) を介してブラウザ (Browser) を制御し、推薦アイテムを表す Web ページを利用者に提示する。また同時にキャラクタインタフェースの制御も行い、キャラクタに推薦アイテムについてのコメントを発言させる。また、推薦アイテムについての利用者の評価を得る必要がある場合、キャラクタはダイアログボックスを介して利用者へ評価を問う。このようにキャラクタエージェントを利用することで、インタラクティブに情報推薦を行うことが可能になる。

推薦アイテムの選択は推薦モジュールによって実行されるが、この詳細は 3 章で議論されている。

##### 2.1.2 推薦ブラックボード

推薦ブラックボード (Recommendation Blackboard) では、エージェント間で共有される以下の 3 つの情報を保持する。

- 利用者の要求：利用者の情報推薦に対する要求である。例えば、レストラン推薦において、「天王寺でお好み焼きを食べたい。」という要求をすれば（場所：天王寺、ジャンル：お好み焼き）というデータが書き込まれる。
- 推薦アイテム：エージェントが推薦したアイテムの情報である。推薦したエージェント名、アイテム名、アイテムの属性名と属性値のリストが書き込まれる。
- 利用者の評価：推薦されたアイテムに対する利用者の評価情報である。推薦したエージェント名、アイテム名、評価値が書き込まれる。評価値は Good, NoGood のいずれかである。

##### 2.1.3 キャラクタインタフェース

キャラクタは MS-Agent により実装されている。キャラクタ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、情報推薦の主張点や過程を利

用者は容易に理解することができる。またキャラクタの動きや発言内容により、利用者の興味を引くことにも役立っている。

## 2.2 競争型情報推薦アルゴリズム

ここでは複数のキャラクタエージェントによる競争的なアイテム推薦の方法について議論する。

まずアルゴリズムで用いるいくつかの変数を定義しておく。*AgentList* はアイテム推薦に参加するエージェントのリストである。初期値は全エージェントである。*ActiveAgent* は利用者に推薦を行うことのできるエージェントであり、*AgentList* から選択される。*ProposalItem* は *ActiveAgent* が推薦するアイテムである。*LeadingAgent* はその推薦アイテムを利用者によって Good と直近に評価されたエージェントであり、*AcceptedItem* はそのときに評価されたアイテムである。*ActiveAgent*, *LeadingAgent*, *ProposalItem*, *AcceptedItem* の初期値は全て空値である。

競争型情報推薦アルゴリズムを以下に示す。

(1) 利用者からの要求が推薦ブラックボードに書き込まれる。

(2) *AgentList* 内の各エージェントは推薦ブラックボードを参照し、利用者の要求を獲得する。またそれぞれの情報源から情報収集を行う。

(3) *AgentList* の中から *ActiveAgent* をランダムに選択する。*AgentList* が空であれば終了する。

(4) *ActiveAgent* は *ProposalItem* を決定し、推薦する。*ActiveAgent* が推薦可能なアイテムを持たない場合は *AgentList* から除外され、ステップ (3) に戻る。

(5) *ProposalItem* に対して利用者は (*AcceptedItem* との間で) 評価を行い、その結果は推薦ブラックボードに書き込まれる。このとき、評価が Good の場合は、*ActiveAgent* は *AgentList* から除かれ、*LeadingAgent* が *AgentList* に加えられる。また *ActiveAgent* が *LeadingAgent* となる。さらに *ProposalItem* が *AcceptedItem* に設定される。評価が NoGood の場合には何もしない。

(6) *AgentList* 内の各エージェントは推薦ブラックボードを参照し、*ProposalItem* とそれに対する利用者の評価を獲得する。

(7) ステップ (3) へもどる。

*LeadingAgent* の初期値は空値であり、以後ステップ (5) において、利用者から直近にそのアイテムが Good と評価されたエージェントに入れ替わってゆく。

したがって *LeadingAgent* になりうるエージェントの数はつねに 1 である。アルゴリズム実行中、ステップ (4) において、推薦するアイテムがなくなったエージェントは *AgentList* から除かれてゆく。*AgentList* が空になると、アルゴリズムは終了し、その時点で *LeadingAgent* が競争に勝ち残ったエージェントであり、*AcceptedItem* が利用者により受け入れられたアイテムである。

### 2.3 レストラン推薦への応用

次に競争型情報推薦システムの応用例としてレストラン推薦システムを示す。レストラン推薦システムでは、利用者が「心斎橋でカレーを食べたい」というような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。

#### 2.3.1 設定

エージェントは  $\alpha$  と  $\beta$  の 2 体とし、それぞれ個別のレストラン情報源を用いる<sup>(注1)</sup>。エージェントが推薦するレストラン情報の属性を、「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効用が大きいとする。以下、2 つの属性名を「予算」と「距離」とする。

#### 2.3.2 動作例

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 2 に示す。

- (1) 利用者は「心斎橋でカレーを食べたい」という要求を送る。
- (2) エージェント  $\alpha$  は「予算 3000 円、距離 91 m」のレストラン A を推薦する。
- (3) 利用者は Good を返答する。
- (4) エージェント  $\beta$  は「予算 1000 円、距離 336 m」のレストラン B を推薦する。このレストランはのレストラン A と比較して、予算は優れているが距離は劣っている。
- (5) 利用者は NoGood を返答する。
- (6) 利用者の返答が NoGood であったので、エージェント  $\beta$  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 C を推薦する。
- (7) 利用者は Good を返答する。
- (8) エージェント  $\alpha$  は「予算 4000 円、距離 952 m」のレストラン D を推薦する。

(注1): プロトタイプでは Yahoo!グルメ (<http://gourmet.yahoo.co.jp/>) 関西ウォーカー (<http://www.walkerplus.com/kansai/gourmet/>) を用いた。



図 2 レストラン推薦システムの動作画面

Fig. 2 A snapshot of restaurant recommendation system.

(9) 利用者は NoGood を返答する。

(10) エージェント  $\alpha$  は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。

以上のようにどちらかのエージェントが推薦するものがなくなるまで競争的にレストランの推薦を続ける。この例では利用者はエージェント  $\beta$  からのレストラン C を最終的に受け入れていることになる。

## 3. 合理的推薦手法

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、それらが自律的に推薦を行うことから、利用者が望まない非合理的な推薦が行われる可能性がある。例えば、前章の例では、エージェント  $\alpha$  はレストラン D を推薦しているが、これは先に推薦したレストラン A よりも予算と距離の両方の属性で劣っており、利用者が NoGood の評価を行うのは明らかである。したがって、競争型情報推薦システムではエージェントはこのような非合理的な提案を避けることが望ましい。そのために利用者の効用を学習しながら推薦を行う合理的な推薦手法を提案する。

### 3.1 合理的な推薦

まず合理的な推薦とは何かを定義しよう。本論文では、利用者がアイテムに対して抱く効用を多属性効用関数でモデル化する。すなわち、アイテム  $c$  は  $X_1, X_2, \dots, X_n$  の  $n$  個の属性を持つと仮定し、 $X_j$  に関する属性値を  $x_j$  とするとき、アイテム  $c$  を以下のように表す。

$$c = \langle x_1, x_2, \dots, x_n \rangle$$

また、アイテム  $c$  に対する利用者の効用を次式で表す。

$$U(c) = \sum_{j=1}^n w_j u_j(x_j) \quad (1)$$

ここで  $w_j$  は属性  $X_j$  に対する重みであり、 $\sum_{j=1}^n w_j = 1, 0 \leq w_j \leq 1$  を満たす。また  $u_j(x_j)$  は属性  $X_j$  に関する効用関数であり、 $0 \leq u_j(x_j) \leq 1$  をとる関数である。したがって *AcceptedItem*  $q$  と *ProposalItem*  $p$  が存在し、 $U(q) < U(p)$  の場合、利用者は  $p$  に対して Good の評価を与え、 $U(q) \geq U(p)$  の場合、利用者は  $p$  に対して NoGood の評価を与える。

合理的推薦アルゴリズムでは *AcceptedItem*  $q$  を基準として、アイテムを以下の三つに分類する。

- 領域  $R$ : 合理的なアイテムが属する領域である。領域内のアイテム  $p$  は、*AcceptedItem*  $q$  より効用が大きい。つまり  $U(p) > U(q)$  である。
- 領域  $I$ : 非合理的なアイテムが属する領域である。領域内のアイテム  $p$  は、*AcceptedItem* より効用が同じか小さい。つまり  $U(p) \leq U(q)$  である。
- 領域  $U$ : 合理的か非合理的かまだ判断できないアイテムが属する領域である。領域  $R$ , 領域  $I$  に属していないアイテムは領域  $U$  に属する。

したがって合理的な推薦とは、領域  $R$  と領域  $U$  に属するアイテムのみを利用者に提案し、利用者にとって Good の評価を得ることのない領域  $I$  に属するアイテムを提案しないことである。

### 3.2 効用の推測

エージェントにとって式 (1) で示される利用者の効用が明らかであれば、領域  $U$  は存在せず、推薦アイテムは領域  $R$  と  $I$  に分類できるが、本論文ではそれがエージェントにとって未知の場合を扱う。すなわち、提案する合理的推薦手法では合理的な推薦を行いながら、それに対する利用者の評価を元に、利用者の効用を推測してゆく。この推測にあたって、本論文では、以下の仮定をおいている。

(仮定 1) 利用者の選好は式 (1) で表されるように、各属性の効用関数の線形加重和として表される。

(仮定 2) 式 (1) 中において各属性の効用関数は既知であるとし、重みのみ未知であるとする。

したがって、ここでの効用の推測とは式 (1) における各属性の重みを推測することである。さて、この二つの仮定の妥当性についてはさらに議論を加えよう。仮定 1 に関しては文献 [2], [5], [7] があげられるように

人間の選好を定式化する手段として研究され、応用されてきたものであるといえる。仮定 2 に関しては一般的には妥当な仮定とはいえないが、近似的に得られることもある。例えば、同一の商品であれば安い方が望ましく、レストランであれば駅や駐車場から近い方が望ましいという一般的な傾向がある。一方、あまり安すぎるレストランを好まない人間もあり、その傾向に個人差がある場合には単純な近似的手法では対処できない。それぞれの利用者の効用関数をどのように推定するかは重要であるが、これは今後の研究課題である。

それでは合理的推薦手法に必要な二つの概念、荷重限界点と領域判定式について述べる。以下の議論では簡化のために属性数が 2 の場合をまず議論する。

#### a) 荷重限界点

合理的推薦手法では利用者の効用における各属性の重みを推測する。荷重限界点とは、利用者モデルの重みの推測範囲の限界を表すものであり、推測が進むにつれてその範囲は狭まってゆく。例えば、属性数が 2 である場合、重み  $w_1, w_2$  の有効範囲の初期値は、

$$\begin{aligned} 0 \leq w_1 \leq 1 \\ 0 \leq w_2 \leq 1 \\ w_1 + w_2 = 1 \end{aligned} \quad (2)$$

を満たす。これを荷重限界点  $s_1, s_2$  を用いて表すと次のようになる。

$$s_1 = [1, 0]$$

$$s_2 = [0, 1]$$

ここで荷重限界点  $s_1$  は  $w_1 = 1$  の場合、 $s_2$  は  $w_2 = 1$  の場合を表している。このときの重みの推測範囲は図 3 に示すように荷重限界点を結ぶ直線として表され、利用者の真の重みはこの直線上の 1 点となる。

#### b) 領域判定式

領域判定式とは、荷重限界点を与えられたときに、アイテムがどの領域に属するのかを判定するためのものであり、これは荷重限界点の数だけ存在する。*AcceptedItem* を  $q$ , *ProposalItem* を  $p$  としたとき、2 属性空間における荷重限界点  $s_1, s_2$  に対する領域判定式  $f_1, f_2$  を以下に示す。なお、アイテムは各属性の効用関数値対の行列  $[u_1(x_1), u_2(x_2)]$  で表されるものとし、 $T$  は転置行列を表す。

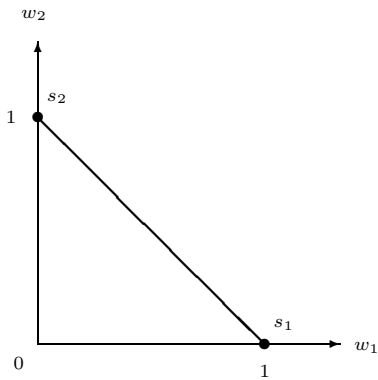


図 3 荷重限界点の初期値

Fig. 3 The initial state of weight boundary points.

$$f_1(p, q) = s_1(p - q)^T$$

$$f_2(p, q) = s_2(p - q)^T$$

与えられた *ProposalItem*  $p$  に対し、全ての領域判定式の値が正であるなら、それは推測範囲内のすべての重みにおいて、アイテム  $p$  の効用がアイテム  $q$  の効用より大きくなり、アイテム  $p$  は領域  $R$  に分類される。逆に全て負の値であるなら、それはアイテム  $p$  の効用がアイテム  $q$  の効用より小さくなり、アイテム  $q$  は領域  $I$  に分類される。それ以外であるなら効用が大きいかわ小さいかわが不確定ということになり、領域  $U$  に分類される。

例えば、荷重限界点が式 (2) で示す初期値であり、*AcceptedItem* が  $q = [0.5, 0.5]$  であるなら、そのときの領域境界線とアイテムを分類する領域の関係は図 4 で示される。ここでの領域判定式は以下の通りである。

$$\begin{aligned} f_1 &= [1, 0] \cdot [u_1(x_1) - 0.5, u_2(x_2) - 0.5]^T \\ &= u_1(x_1) - 0.5 \end{aligned} \quad (3)$$

$$\begin{aligned} f_2 &= [0, 1] \cdot [u_1(x_1) - 0.5, u_2(x_2) - 0.5]^T \\ &= u_2(x_2) - 0.5 \end{aligned} \quad (4)$$

ここで式 (3) は荷重限界点  $s_1$  が表す  $w_1 = 1$  を想定したときの境界線を表している。このとき  $x_2$  の効用は無視されているので、 $u_1(x_1)$  の値が 0.5 より大きければ少なくとも非合理的とはいえない。また式 (4) は

荷重限界点  $s_2$  が表す  $w_2 = 1$  を想定したときの境界線を表している。このとき  $x_1$  の効用は無視されているので、 $u_2(x_2)$  の値が 0.5 より大きければ少なくとも非合理的とはいえない。したがって両式の値が 0 より大きい領域  $R$  は合理的であり、0 以下である領域  $I$  は非合理的であるといえる。またそれ以外の領域  $U$  は不確定である。

### c) 領域の更新

領域はアイテムの推薦が行われるごとに更新されてゆく。とくに、領域  $U$  内のアイテム  $p$  を利用者に提示した場合には、その評価に応じて更新が異なる。すなわち、*AcceptedItem* を  $q$  としたとき、利用者の評価が Good である場合には  $U(p) > U(q)$  を満たすように、また NoGood である場合には  $U(p) \leq U(q)$  を満たすように、領域判定式を更新する必要がある。

例えば、図 4 に示すように、荷重限界点が図 3 の初期状態で、*AcceptedItem* が  $q = [0.5, 0.5]$  であったとき、*ProposalItem*  $p = [0.7, 0.4]$  が推薦され、利用者の評価が Good であったとしよう。この場合、 $q$  の値を式 (3) と式 (4) に代入すると、式 (4) は負の値となり、矛盾するので削除する。次に、 $U(p) > U(q)$  より得られる  $0.7w_1 + 0.4w_2 > 0.5w_1 + 0.5w_2$  と式 (2) から  $1 \geq w_1 > 1/3$ ,  $2/3 > w_2 \geq 0$  が得られ、新たな荷重限界点  $s_3 = [1/3, 2/3]$  が得られる。その結果、領域の境界は図 5 のように更新される。ここで  $f_3(p, q) = s_3(p - q)^T$  である。また、利用者の評価が NoGood であった場合は、 $U(p) \leq U(q)$  より、領域の境界は図 6 のようになる。このように、領域  $U$  からの推薦が行われた場合、利用者からの評価に応じて、荷重限界点が変更され、推測範囲が狭められてゆく。

また領域  $R$  からアイテム推薦を行った場合は、利用者が Good の評価を行うことが自明であるので、荷重限界点に変化はなく、重みの推測範囲は変化しない。ただし推薦アイテムの効用に応じて、領域  $R$  の範囲は減少し、逆に領域  $I$  の範囲は増加する。例えば、図 4 の状態において、領域  $R$  ないのアイテム  $r = [0.7, 0.7]$  を推薦したとすると、領域は図 7 のように変化する。ここで  $f_1(p, r) = s_1(p - r)^T$ ,  $f_2(p, r) = s_2(p - r)^T$  である。このときに図 4 の状態では領域  $R$  に含まれていたアイテム  $q$  が図 7 では領域  $I$  に含まれるようになってきていることに注意してほしい。これは図 7 の状態では  $U(q) < U(r)$  であるといえるからである。これは過去に推薦したものをもう一度推薦することは合理的でないことも意味している。

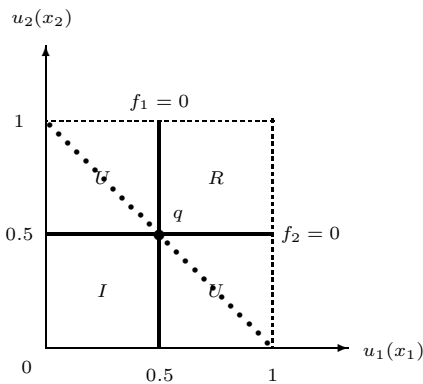


図 4 領域境界線と各領域  
Fig. 4 Regions and boundaries.

競争型情報推薦が開始される時点では領域は  $U$  のみ存在する．その後、あるアイテム  $q$  が Good と評価された時点で、図 4 に示すように領域は  $R, I, U$  に区分される．その後、領域  $R$  から推薦が行われると図 7 のように変化し、領域  $R$  が狭められる．また領域  $U$  から推薦されるとそれに対する利用者からの評価に応じて図 5 と図 6 に示すように、領域  $U$  の範囲が狭められる．この際、真の利用者の効用は重みが確定した属性効用関数値の線形加重和であるので、図 4 の太破線で示されるように、領域  $U$  はそれを示す直線へと収束してゆくことになる．この特徴は、状態空間を線形分割し、領域  $U$  の領域を狭めてゆく本手法の妥当性を示している．

#### d) 3 属性以上の場合

属性数が  $m$  である場合、荷重限界点は初期状態で  $m$  個存在し、推測が進むにつれてその数は増加する．ただし、推測範囲は荷重限界点により囲まれる  $m$  次元空間上の凸多面体として表現可能である．領域判別式の数は荷重限界点の数と等しくなる．荷重限界点の更新は凸多面体上での交点を考慮する必要がある．

## 4. 推薦戦略とその性能評価

本章では、合理的推薦手法におけるエージェントの推薦戦略とその性能評価についての述べる．

### 4.1 推薦戦略

合理的推薦手法において *ProposalItem* の決定法として、領域  $R$  または  $U$  から選択するという制約しか与えていない．ここではその中からさらにどのアイテム

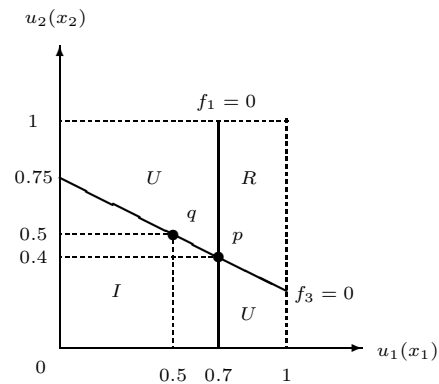


図 5 領域  $U$  からの推薦：Good の場合  
Fig. 5 Recommendation from region  $U$ : “Good” case.

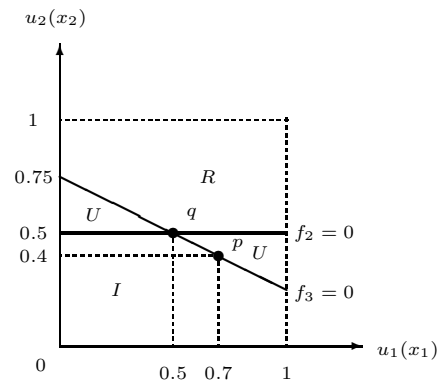


図 6 領域  $U$  からの推薦：NoGood の場合  
Fig. 6 Recommendation from region  $U$ : “NoGood” case.

ムを選択すべきかに関する推薦戦略について述べる．推薦アイテムを決定するための戦略には以下の 2 つが考えられる．

- 利得優先型 (Best Profit Strategy) : 合理性を満たすアイテムの中から、自己の利得が最大となるアイテムを選択し、推薦アイテムとする．例えば、レストラン推薦の場合は推薦に成功した場合、何らかの報酬がレストランから支払われるならば、その額が最大になるレストランから順に推薦する．

- 学習優先型 (Best Learning Strategy) : 学習優先型では、合理性を満たすアイテムの中で推薦を行っ

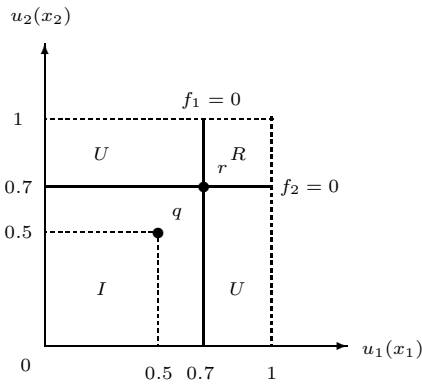


図 7 領域  $R$  からの推薦  
Fig. 7 Recommendation from region  $R$ .

たときの学習効果が最大となるアイテムを選択し、推薦アイテムとする。学習効果が最大とは重みの有効範囲の変化の期待値が最大になるアイテムを選択することであり、これは領域  $U$  を最も小さくするアイテムといえる。当然これは領域  $U$  からのアイテムの選択を意味している。領域  $U$  からのアイテムが存在しない場合には、領域  $R$  からのアイテムをランダムに選択する。

#### 4.2 評価シミュレーション

利得優先型と学習優先型の性能をシミュレーションを行うことにより確認した。

##### 4.2.1 設定

エージェント数は 3 で、アイテムの属性数は 2 であるとする。アイテムの属性値  $[x_1, x_2]$  は  $0 \leq u_1(x_1)^2 + u_2(x_2)^2 \leq 1$  を満たすようにランダムに設定している。これにより、 $[1, 1]$  のように、すべての属性で優れているアイテムが存在しないようにしている。もしそのようなアイテムが存在するならば、エージェントの効用学習は意味がないからである。またそれぞれのアイテムに 0 から 1 の範囲でエージェントの利得をランダムに設定している。

実験は次のように行った。

(1) ランダムな属性値と利得をもつアイテムを指定された個数生成し、各エージェントに均等に分配する。

(2) 利用者モデルにおける重みをランダムに決定する。

(3) 3 体のエージェントすべてを利得優先型あるいは学習優先型のいずれかの戦略に設定する。競争

型推薦を実行し、最終的な *AcceptedItem* より、エージェントの効用（利得）、利用者の効用を得る。また推薦が収束するまでに必要とした推薦回数を調べる。

アイテム数を 100 から 1000 まで変化させ、それぞれの場合に以上のプロセスを 10000 回繰り返し、その平均値を求めた。

##### 4.2.2 結果と考察

シミュレーションにおける、推薦戦略によるエージェントの効用の比較を図 8 に、利用者の効用の比較を図 9 に、推薦回数の比較を図 10 示す。

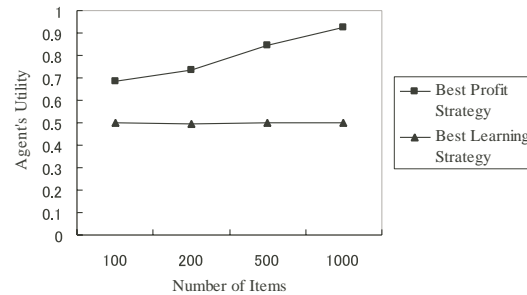


図 8 エージェントの効用の比較  
Fig. 8 Comparison of agent's utility.

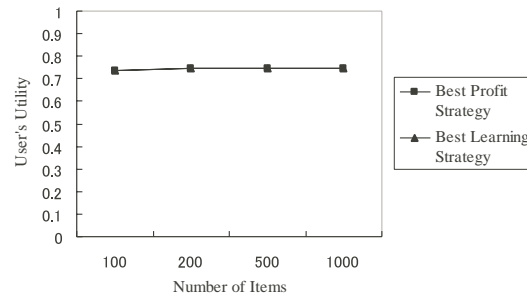


図 9 利用者の効用の比較  
Fig. 9 Comparison of user's utility.

図 8 では、利得優先型の方が学習優先型よりもエージェントの効用が高くなっていることを示している。利得優先型ではエージェントは自己の利得を最大化するように推薦を行うのに対し、学習優先型ではエージェントの利得は考慮していないので当然の結果といえる。アイテム数が増えるにつれエージェントの効用が上がるのはエージェントの効用と利用者の効用を同時に上げるアイテムの存在確率が増えるからである。



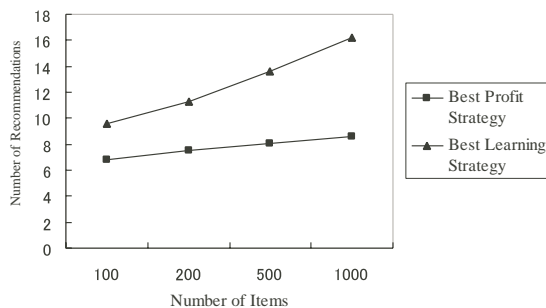


図 10 推薦回数の比較

Fig. 10 Comparison of the number of recommendations.

図 9 では、どちらの戦略でも利用者の効用に差は見られないことを示している。これは、利得優先型、学習優先型のどちらの戦略であっても、最終的な推薦アイテムの決定は利用者が行うので、同様のアイテムに収束するためである。

図 10 では、利得優先型の方が学習優先型よりも少ない推薦回数で推薦が収束していることを示している。これは、学習優先型では利用者の効用を学習するために領域  $U$  からのアイテムを優先的に推薦するが、領域  $R$  からのアイテムを推薦しないので、利用者が望むアイテムを入手するのに時間がかかることを意味している。むしろ、利益優先型では領域  $R$  と  $U$  の両方から推薦を行っているので、領域  $R$  がより早く絞り込まれ、利用者の望むアイテムにより早く収束しているといえる。

以上の実験結果から利用者の効用を優先的に学習しようとする学習優先型よりも、利己的にアイテムを推薦し、その結果を通じて利用者の効用を学習する利得優先型の方が性能的に優れていることが明らかになった。これはエージェントが利用者の効用を考慮して協調的にふるまうよりも、利己的にふるまう方が推薦プロセスが早く収束するという競争型情報推薦システムの一つの特徴を示している。

## 5. 関連事例

近年、B-to-C 電子商取引を支援する手段としてさまざまな情報推薦システムが開発され、実用化の段階にまで至っている。従来の情報推薦システムは、amazon.com<sup>(注2)</sup>のように情報提供者が単一の情報源から利用者へ推薦を行う独立型情報推薦システムと Deal-

Time<sup>(注3)</sup>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に大きく分類できる。本節では情報提供者と情報利用者の視点からそれぞれのシステムの利点欠点を述べ、本論文でわれわれが提案する競争型推薦システムとの違いについて議論する。

独立型情報推薦システムは、単一の情報源から推薦が行われるので、利用者は他の情報源との比較はできない。したがって情報提供者は利用者に対して情報を直接推薦することができるので、情報提供者にとってよりメリットのあるシステムといえる。

これに対しブローカ型情報推薦システムでは、ブローカが複数の情報源から情報を収集し、その比較が可能ないようにして利用者に推薦する。したがって、利用者は複数の情報を比較できるという利点がある。しかし情報推薦の主導権はブローカに握られているため、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。例えば商品販売の場合、ブローカ型情報推薦システムでは売り手が提供できる様々なオプションがブローカによりフィルタリングされてしまい、単なる価格競争の場になってしまう場合が起こりうる。したがってブローカ情報推薦システムは情報利用者によりメリットのあるシステムといえる。

本論文で提案する競争型情報推薦システムでは情報提供者と利用者の間にブローカを介在させるのではなく、相互作用可能な仮想的な場を提供し、そのやり取りを通して情報推薦を行おうとするものである。したがって情報提供者はエージェントを介して利用者に情報を直接推薦することができる一方、情報利用者は複数のエージェントと相互作用しながら、希望の情報を入手することができる。したがって競争型情報推薦システムは情報提供者と利用者の双方にメリットのあるシステムであるといえる。その反面、情報利用者が希望の情報を入手するまでに多くの相互作用が必要になる場合もある。

## 6. まとめ

複数の情報提供者がそれぞれのエージェントを介して利用者と相互作用を行うことにより、競争的に情報推薦する競争型情報推薦システムを提案した。競争的な情報推薦は電子商取引社会の発展に伴い、今後ますます重要になってくると考えられる。また競争型情報

(注2): <http://www.amazon.com/>(注3): <http://www.dealtime.com/>

推薦システムにおいてエージェントがアイテムの合理性を判断して推薦を行うための合理的推薦手法を提案した。また二つの推薦戦略を示し、その性能を明らかにした。

今後の課題としては以下の点に取り組む必要がある。

(1) よりよい推薦戦略：本論文では利得優先型と学習優先型の二つの推薦戦略のみを議論したが、さらに効率の良い推薦戦略がないかを研究する必要がある。例えば、学習優先と利得優先を組み合わせることにより、エージェントの利得と学習効果を両立させ、推薦回数を減少させることが可能になるかもしれない。

(2) 効用関数の推定：現段階では、利用者のアイテム評価に誤りや揺らぎがなく、また個別属性の効用関数が自明という設定で議論を行ってきた。このような前提が成立しない場合への対処が必要になる。

(3) 異種エージェント間の競争：本論文ではエージェントが推薦するアイテムの属性や推薦戦略は一樣として議論を進めてきた。これらがエージェントごとに異なる異種エージェント間での競争も興味ある研究課題である。

(4) より高度なインタラクションの実現：現段階では、利用者とのインタラクションは、利用者に推薦アイテムに対して Good か NoGood かを問うだけとしているが、これをより対話的にすることが考えられる。例えば、利用者が NoGood と答えるときに、それに付随する理由を同時に加えることができれば、利用者の効用の推測が促進される。

(5) 現実問題への応用：本提案手法を現実の電子ショッピングシステムなどに実装し、実用的な見地からその評価を行う必要がある。

謝辞 本研究の一部は文部科学省科学研究費（特定領域研究「情報洪水時代におけるアクティブマイニングの実現」）、日立製作所システム開発研究所、イメージ情報科学研究所からの支援を受けて行ったものである。また本研究の先行システムの開発に協力していただいた回り道康博氏（現 NTT アドバンステクノロジー）に感謝の意を表します。さらに貴重なコメントをいただいた査読者の方々にも感謝します。

## 文 献

- [1] E. Andre and T. Rist: "Adding Life-Like Synthetic Characters to the Web", in Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence 1860, eds. M. Klusch and L. Kerschberg, pp 1-13, Springer (2000).
- [2] R. A. Keeney and H. Raiffa: "Decisions with Mul-

iple Objectives: Preference and Value Tradeoffs", Cambridge University Press (1993).

- [3] 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, pp 451-459 (1998).
- [4] Y. Kitamura, T. Sakamoto, and S. Tatsumi: "A Competitive Information Recommendation System and Its Behavior", in Cooperative Information Agents VI, Lecture Notes in Artificial Intelligence 2446, eds. M. Klusch, S. Ossowski, and O. Shehory, pp 138-151, Springer (2002).
- [5] 小橋康章: 決定を支援する, 東京大学出版会刊 (1998).
- [6] G. N. Linden and S. Hanks: "Interactive Assessment of User Preference Models: The Automated Travel Assistant", in Proceedings of the Sixth International Conference on User Modeling, pp 67-78 (1997).
- [7] 岡本眞一: "商品選択問題についての多属性効用関数法の応用", 経営情報科学, 2, 3, pp.227-234 (1989-12).
- [8] H. Shimazu: "ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing", Proceedings of Seventeenth International Joint Conference on Artificial Intelligence, pp 1443-1448 (2001).

(平成 x 年 xx 月 xx 日受付)

### 阪本 俊樹

2001 大阪市立大学工学部情報工学科卒。  
2003 同大学院修士課程了。2003 NEC システムテクノロジー (株) 入社。在学中、情報推薦システムの研究に従事。

### 北村 泰彦 (正員)

1983 大阪大学基礎工学部情報工学科卒。  
1988 同大学院博士課程了。工学博士。1988 大阪市立大学工学部電気工学科助手。同情報工学科助教授を経て、現在関西学院大学理工学部情報科学科教授。マルチエージェントシステム、ヒューリスティック探索、WWW 情報統合の研究に従事。IEEE, AAAI, ACM, 人工知能学会, 情報処理学会, ソフトウェア科学会等の会員。2001 人工知能学会全国大会優秀論文賞, CIA-02 System Innovation Award 受賞。

### 辰巳 昭治 (正員)

1970 大阪大学工学部通信工学科卒。1972 同大学院修士課程了。1972 川崎重工業入社。1978 大阪大学大学院博士課程了。工学

博士．豊橋技科大学を経て，現在大阪市立  
大学工学部情報工学科教授．統計的パター  
ン認識，意思決定問題，画像処理用並列プ  
ロセッサの開発，VLSI 向き相互結合網の構成法などの研究に  
従事．電子情報通信学会，情報処理学会，ソフトウェア科学会，  
IEEE 等の会員．