

競争型推薦システムにおける合理的提案アルゴリズム

阪本 俊樹[†] 北村 泰彦[†] 辰巳 昭治[†]Toshiki SAKAMOTO[†], Yasuhiko KITAMURA[†], and Shoji TATSUMI[†]

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の要求を満たす情報を入手するための手段としては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。本稿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、各エージェントが自律的に推薦を行うため、利用者にとって必ずしも望ましい情報が推薦されるとは限らない。そこで、利用者の情報に対する効用を多属性効用関数で表現し、各属性に対する重みを学習していくことで、利用者にとってより望ましい情報を推薦する合理的提案アルゴリズムを提案する。エージェントの情報推薦戦略として、自己の利益を優先する利益優先型、推薦効率を優先する学習優先型などの戦略を示し、エージェントの利益と推薦効率がトレードオフの関係にあることを議論する。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) [4] や ExpertClerk [5] 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

[†] 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

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ー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、本稿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、利用者とインタラクションしながら、競争的に動作する。このようなエージェントの応用例としては、インターネット上での電子ショッピングを仲介するセールスエージェントが考えられる。従来の電子ショッピングを仲介するエージェントシステムとしては複数のショッピングサイトから同一の商品に関する情報を収集し、その中で最も安価なサイトを推薦する比較ショッピングエージェントが注目されていた。しかしながらこのような仲介エージェントは電子ショッピングを単なる価格競争だけにしまい、個別のショッピングサイトにとっては必ずしも歓迎できるサービスとはいえなかった。それに対して競争型情報推薦システムは複数のセールスエージェントがキャラクタを介して自由に競争できるような仮想空間を作ることにある。それぞれのセールスエージェントは単に価格の競争を行うだけでなく、商品に伴う特徴や付加価値を自由に利用者にアピールすることができる。

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

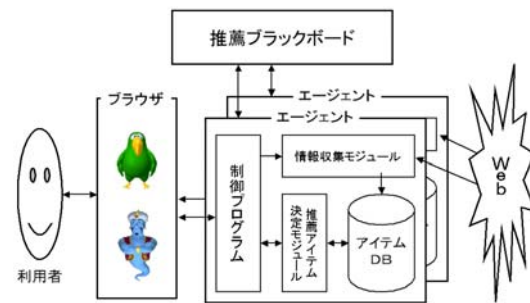


図 1 競争型情報推薦システムの構成

れる。そこで本稿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提案する合理的提案アルゴリズムを提案する。また情報推薦戦略として自己の利益を優先する利益優先型、推薦効率を優先する学習優先型などの戦略を示し、エージェントの利益と推薦効率がトレードオフの関係にあることを議論する。

以下、2章で競争型情報推薦システムについて、3章で合理的提案アルゴリズムについて述べ、4章ではエージェントの取る戦略による推薦の挙動の違いを考察し、5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は、図 1 に示すように、複数のエージェントとキャラクタインタフェース、推薦ブラックボード、ブラウザ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提案を行う。

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抽出ラッパを用いて行う。情報抽出ラッパは 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。

b) アイテム提案

エージェントはブラウザを制御し、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。

2.1.2 推薦ブラックボード

推薦ブラックボードでは、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、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

競争型情報推薦は複数のキャラクタエージェントが競争的にアイテムの推薦を行う。

ここで以下の議論で用いるいくつかの用語を定義しておく。*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* は利用者に提案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* が提案するアイテムである。*LeadingAgent* は利用者に最後に「Good」と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。

次に競争型情報推薦アルゴリズムを示す。

(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) 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。

(3) 各エージェントはそれぞれの情報源から情報収集を行う。

(4) *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。

(5) *ActiveAgent* はアイテムの推薦を行う。*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、(4) に戻る。

(6) 推薦アイテムに対して利用者が評価を行う。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、*ActiveAgent* が *LeadingAgent* となる。評価が「NoGood」の場合には何もしない。

(7) *AgentList* 内の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利用者の評価を獲得する。

(8) (4) へもどる。

アルゴリズム実行中、推薦するアイテムがなくなり、競争に負けたエージェントは *ActiveAgent* から除かれてゆき、アルゴリズムが終了した時点で *LeadingAgent* となっているものが競争に勝ち残ったエージェントとなる。

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「心齋橋でカレーを食べたい」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。

2.3.1 設定

レストラン推薦システムにおいて、エージェントは X と Y の2体とし、それぞれ個別のレストラン情報源を用いる。(注1)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。

2.3.2 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2に示す。

(1) 利用者は「心齋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 X 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 Y 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 Y 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 X は「予算 4000 円、距離 952 m」のレストラン D を推薦する。

(9) 利用者は NoGood を返答する。

(10) エージェント X 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 Y 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。

さて、以上の例において、エージェントは利用者の効用を特に考慮することなくレストランを推薦している。例えば、エージェント X はレストラン D を推



図2 レストラン推薦システムの動作画面の一例

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 の返答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推薦を避けることが望ましい。そのため利用者の効用を学習しながら情報推薦を行う合理的提案アルゴリズムを次章で述べる。

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、利用者に対して、非合理的な推薦が行われる可能性がある。そこで、利用者の効用を考慮した合理的提案アルゴリズムを提案する。

3.1 合理的な提案

合理的提案アルゴリズムでは、利用者の選好を多属性効用関数でモデル化する。アイテム C_i が X_1, X_2, \dots, X_n の n 個の属性を持ち、 X_j に関する属性値を $x_{i,j}$ とするとき、アイテム C_i を以下のように表す。

$$C_i = \langle x_{i,1}, x_{i,2}, \dots, x_{i,n} \rangle \quad (1)$$

また、アイテム C_i に対する利用者の効用を次式で表す。

$$U(C_i) = \sum_{j=1}^n w_j u_j(x_{i,j}) \quad (2)$$

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 $u_j(x_{i,j})$ は属性 X_j に関する効用関数であり、 $0 \leq u_j(x_{i,j}) \leq 1$

(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。

をとる関数である。したがって *AcceptedItem* C_1 と *ProposalItem* C_2 が存在し、 $U(C_1) > U(C_2)$ の場合、利用者は C_2 に対して NoGood の評価を与える。

合理的提案アルゴリズムでは *AcceptedItem* を基準として、推薦アイテムを以下の三つに分類できる。

- 領域 R : 合理的なアイテムが属する領域である。アイテム C_i が合理的とは、*AcceptedItem* より効用が大きい、つまり $U(C_i) > U(\text{AcceptedItem})$ を満たしているということである。

- 領域 I : 非合理的なアイテムが属する領域である。アイテム C_i が非合理的とは、*AcceptedItem* より効用が小さい、つまり $U(C_i) < U(\text{AcceptedItem})$ を満たすアイテムである。

- 領域 U : 合理的かどうか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 に属していることになる。

したがって合理的な提案とは、領域 R と領域 U に属するアイテムを利用者に提案することになる。

3.2 合理的提案アルゴリズム

エージェントにとって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。合理的提案アルゴリズム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、利用者の効用を推定してゆくものである。なお、本論文では、エージェントにとって利用者の持つ個別の属性に関する効用関数は既知であり、その重みのみが未知であると仮定している。したがって、合理的提案アルゴリズムでは、利用者モデルの重みを推測する。

ここではまず合理的提案アルゴリズム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。

a) 荷重限界点

荷重限界点とは、利用者モデルの重みの有効範囲を表すものである。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、 $0 \leq w_1 \leq 1, 0 \leq w_2 \leq 1, w_1 + w_2 = 1$ を満たす2次元空間における直線となり、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。

$$s_1 = [1, 0]$$

$$s_2 = [0, 1]$$

このときの重みの有効範囲と荷重限界点の状態を図3に示す。

つまり、2次元空間上で s_1, s_2 の2点を結んで

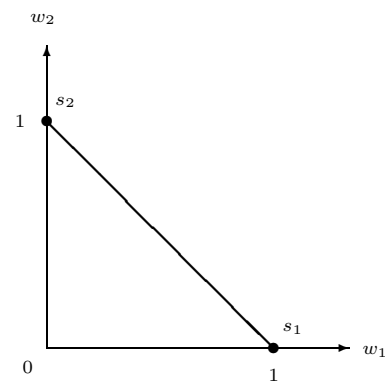


図3 2属性の場合の重みの有効範囲

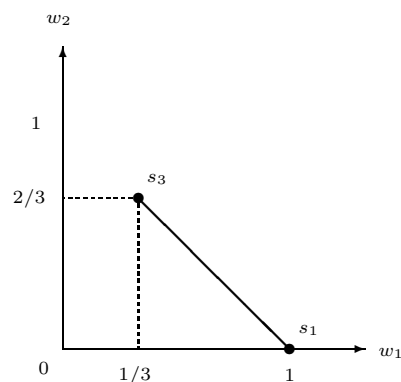


図4 重みの有効範囲(1)

きる直線が、重みの有効範囲を表していることになる。なお、属性数が m である場合、初期の荷重限界点は m 個あり、その状態は m 次元空間に拡張される。

例えば、荷重限界点が図3の状態、*AcceptedItem* が $q = [0.5, 0.5]$ であったとき、*ProposalItem* $p = [0.7, 0.4]$ のアイテムが提案され、利用者の応答が「Good」であった場合、重みの有効範囲は図4で示される。すなわち、荷重限界点 s_3 は $U(p) > U(q)$ より、 $0.7w_1 + 0.4w_2 > 0.5w_1 + 0.5w_2$ の不等式を解くことにより $s_3 = [1/3, 2/3]$ と得られる。また、利用者の応答が「NoGood」であった場合、 $U(p) < U(q)$ より、重みの有効範囲は図5で示される。このように、提案により荷重限界点が変更され、有効範囲が狭められていくことになる。

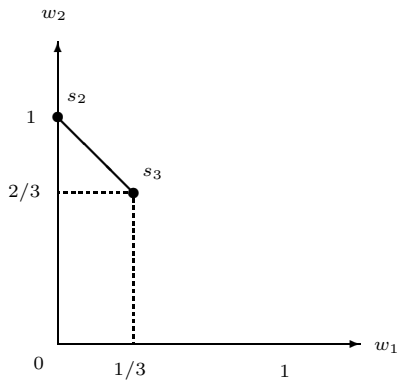


図 5 重みの有効範囲 (2)

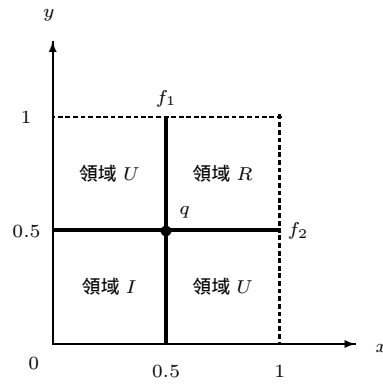


図 6 領域境界線と各領域の状態

b) 領域判定式

領域判定式とは、重みの有効範囲に対して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、これは荷重限界点ごとに設定される。AcceptedItem を q 、ProposalItem を p とすると、上記の荷重限界点 s_1, s_2 に対する領域判定式 f_1, f_2 は次のように表される。なお、各アイテムは $[x, y]$ の形で表されるものとする。

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

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

この領域判定式より得られる値が全て正の値であるなら、それは有効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、荷重限界点が上記の初期値であり、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 6 で示される。また、上記の図 4、図 5 に対応する領域の状態はそれぞれ図 7、図 8 で示される。

3.2.1 アルゴリズム

次に合理的提案アルゴリズムを示す。

(1) 初期設定：エージェントは収集したアイテムから $ItemList$ を作成する。また AcceptedItem を

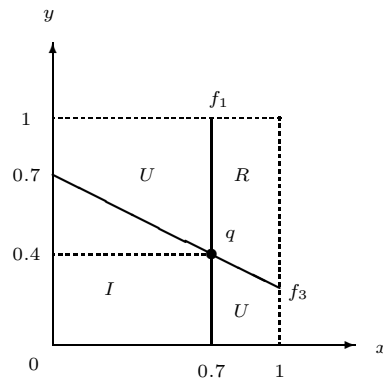


図 7 領域の状態 (1)

$[0, 0, \dots, 0]$ に初期化し、荷重限界点とそれに対応する領域判定式を設定する。

(2) アイテムの分類：ItemList の各アイテムを領域判定式に代入し、その結果から次のようにする。

- すべて正であるなら領域 R に分類する。
- すべて負であるなら領域 I に分類し、ItemList から削除する。
- それ以外は領域 U に分類する。

(3) ProposalItem の決定：各エージェントは領域 R と領域 U のどちらかの領域から推薦アイテムを選択する。

(4) 利用者の応答を獲得：領域 U からの提案のみ利用者へ問い合わせを行い、利用者の応答を獲得する。領域 R からの提案の場合は $U(ProposalItem) >$

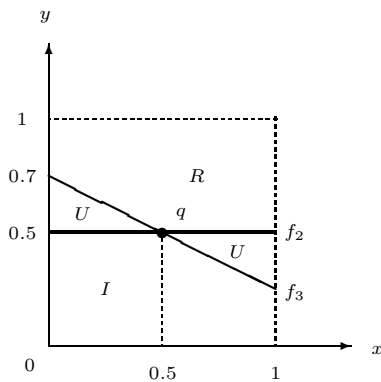


図 8 領域の状態 (2)

$U(\text{AcceptedItem})$ であるので問い合わせは行わなくてもよい。

(5) 荷重限界点と領域判定式を更新

- $ProposalItem$ が領域 R に属する場合: $ProposalItem$ を新たな $AcceptedItem$ とする。

- $ProposalItem$ が領域 U に属しており、利用者の応答が「Good」である場合: $U(ProposalItem) > U(AcceptedItem)$ より、値が負になる領域判定式とその対応する荷重限界点を削除し、新たな荷重限界点と領域判定式を設定する。また、 $ProposalItem$ を新たな $AcceptedItem$ とする。

- $ProposalItem$ が領域 U に属しており、利用者の応答が「NoGood」である場合: $U(ProposalItem) \leq U(AcceptedItem)$ より、値が正になる領域判定式とその対応する荷重限界点を削除し、新たな荷重限界点と領域判定式を設定する。

(6) (2) へもどる。

4. 考察

本章では、合理的提案アルゴリズムにおけるエージェントの推薦戦略について述べ、エージェントの利益と推薦効率の関係について考察する。

4.1 推薦戦略

合理的提案アルゴリズムにおいて、 $ProposalItem$ の決定は領域 R または I から選択されるだけの制約しかなかった。ここではその中からさらにどのアイテ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つのものが考えられる。

- 利益優先型: 合理性を満たすアイテムの中から、自己の利益が最大となるアイテムを選択し、推薦アイテムとする。これはあらかじめアイテムごとにエージェントの利益が定義されている場合である。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、その額がエージェントの利益となる。

- 学習優先型: 学習優先型では、合理性を満たすアイテムの中で提案を行っ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ということであり、これは領域 U からのアイテムの選択を意味している。

利益優先型はエージェントの利益に重点を置いたものであり、学習優先型は利用者にとっての推薦効率に重点を置いたものである。

エージェントが利益優先型の戦略を選択した場合、エージェントは提案アイテムを決定する際に、提案によって得られる自己の利益のみを考慮し、利用者にとっての効用や効率といったことは考慮しない。例えば、2属性アイテムの推薦において、 $AcceptedItem$ が $[0.5, 0.5]$ であるとき、利益優先型のエージェントが $[0.51, 0.51]$ のアイテムを提案を提案する場合 (このアイテムがエージェントにとっての利益が最大であったとする)、このアイテムは領域 R に属することになるため、提案による利用者モデルの学習は全く行われない。またアイテムの持つ属性ごとの効用が $AcceptedItem$ と比べてほとんど差がないため、利用者の得られる効用もわずかに増加するだけであると考えられる。このような提案が連続することは、推薦が長引くことになり、利用者にとっては逆に煩わしいものとなる。

これに対して、エージェントが学習優先型の戦略を選択した場合、優先的に領域 U のアイテムが提案されることになり、利用者モデルの学習が促進される。このことから考えて、推薦効率に関しては、学習優先型の方が優れていると考えられる。しかし、学習優先型ではエージェントの利益については考慮していないため、エージェントの利益とはトレードオフの関係になることが予想される。

5. 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、競争的に情報を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提案を行うための合理的提案アルゴリズムを提案した。また推薦戦略に応じたエージェントの利益と推薦効率の関係を考察した。

今後の課題として以下の3点に取り組む。

(1) 利益と効率の考慮：提案によってエージェントが得られる利益と利用者にとっての効率がトレードオフの関係にあることはすでに述べたが、これを戦略を組み合わせる(例えば、前半の3回は学習優先型で、残りは利益優先型とする)ことで、エージェントが自己の利益を考慮した上で、なおかつ効率的な推薦が行えることが考えられる。また学習優先型の戦略についても、アイテム間の類似度を求め、ある閾値をこえないもののみを提案する、といったことを付加することで効率が改善できると考えられる。

(2) 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。例えば、利用者が「NoGood」と答えるときに、それはこういう理由があるために好ましくない、といった内容にする。それにより、利用者モデルの推測がより効率的になると考えられる。

(3) より現実的な問題への応用：現段階では、利用者の応答に誤りや揺らぎがある、または利用者がモデルにない属性を重視している、といった場合、利用者モデルを推測する際の選好順序の矛盾が解消できず、適切な利用者モデルが推定できなくなると考えられる。よってこの点について考える必要がある。

文 献

- [1] Andre, E., Rist, T.: Adding Life-Like Synthetic Characters to the Web. In Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence 1860. Edited by Klusch, M., Kerschberg, L. Springer. pp 1-13 (2000).
- [2] Keeney, R.A., Raiffa, H.: Decisions with Multiple Objectives: Preference and Value Tradeoffs., Cambridge Univ. 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] Linden, G. N. L., Hanks, S.: Interactive Assessment of User Preference Models: The Automated Travel Assistant, In Anthony Jameson, Cecile Paris, and Carlo Tasso (Eds.) User Modeling: Proceedings of

- the Sixth International Conference, pp 67-78 (1997).
- [5] Shimazu, H.: ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing, Seventeenth International Joint Conference on Artificial Intelligence, pp 1443-1448 (2001).