

推薦理由を考慮した情報推薦システム

力, 規晃
徳山工業高等専門学校情報電子工学科

越村, 三幸
九州大学大学院システム情報科学府情報学部門

藤田, 博
九州大学大学院システム情報科学府情報学部門

長谷川, 隆三
九州大学大学院システム情報科学府情報学部門

<https://doi.org/10.15017/20057>

出版情報：九州大学大学院システム情報科学紀要. 16 (2), pp.75-81, 2011-09-26. 九州大学大学院システム情報科学研究所

バージョン：

権利関係：

推薦理由を考慮した情報推薦システム

力 規晃* · 越村 三幸** · 藤田 博** · 長谷川隆三**

Information Recommender System Considering Recommendation Reasons

Noriaki CHIKARA*, Miyuki KOSHIMURA**, Hiroshi FUJITA** and Ryuzo HASEGAWA**

(Received July 22, 2011)

Abstract: There are a lot of information recommender systems on the Web. These systems aim to find and provide useful information for the users. For example, many online shopping sites recommend merchandise which the user is likely to purchase. However, practical users do not always use these recommendations. In order to make an apt recommendation for the user, we need to give a plausible reason for it. However, the almost all existing systems give very simple or quantitative reasons. This paper aims to present clear and non-quantitative recommendation reasons which everybody is easy to understand. We make use of rules generated by Inductive Learning for the aim. We use Inductive Logic Programming (ILP) and Decision Tree. ILP generates first order predicate logic rules, and Decision Tree generates tree structures based on propositional logic. As an experience, we extract several rules from blogs with ILP and Decision Tree in order to recommend blogs or web pages which the bloggers are likely to be interested in. We succeeded in extracting several useful rules from blogs. We show that the rules could be used for not only recommendations but also giving their reason. The recommendation methods using in this paper are based on collaborative filtering and content-based filtering.

Keywords: Information recommender system, Inductive logic programming, Decision tree, Blog

1. はじめに

現在、インターネット上には無数の情報が溢れ、容易に有用な情報を得ることには困難が伴う。この問題を解決するシステムの一つに、情報推薦システムがある。これは利用者により有用な情報を見つけ出し提示するシステムで、ショッピングサイト等で多数のシステムが実用化されている。実際にそれらのサイトを閲覧している場合、ユーザーは提示された情報を活用する場合もあるが、これらを活用しない場合も多くある。

情報推薦システムで、より多くの情報を活用されるようになるためには、ユーザーの情報推薦システムへの信頼をより高める必要がある。情報推薦システムへの信頼を高める一つの方法は、推薦理由を提示し、推薦の透明性を高めることである³⁾。現在の情報推薦システムの推薦理由は極めて簡単な説明か、定量的な説明でしかない。

本研究では、明確かつ定量的ではない誰にでも理解しやすい推薦理由を示すことを目指している。このため、推薦と推薦理由の提示に、帰納学習で得られた規則を用いる。本論文で使用する帰納学習の手法は帰納論理プログラミング

(Inductive Logic Programming; ILP)²⁾と決定木¹¹⁾である。推薦システムに決定木を利用している例^{4,8)}はいくつかある。また、ILPを推薦システムに利用した例¹⁰⁾も少ないが存在する。

本論文では、まず帰納学習を用いた推薦手法を説明し、次にブログデータをILPや決定木のためにどのように加工して使うかを説明する。そして、ブログの著者に他のブログを推薦するシステムを用いた実証実験の結果を示す。その際に推薦と推薦理由提示に利用する規則をILPと決定木を用いて抽出する。抽出した規則を推薦システムに用いた場合を想定して考察を行う。

2. 推薦手法

推薦システムの推薦手法は大別すると内容ベースフィルタリングと協調フィルタリングに分類できる⁵⁻⁷⁾。前者は推薦アイテムの特徴に着目し、類似のアイテムを推薦する手法である。後者は推薦対象者に着目し、類似した人物の好むアイテムを推薦する手法である。帰納学習自体はこのどちらにも応用可能である。

帰納学習を用いた内容ベースフィルタリングの手法は次のようになる。

- (1-1) 帰納学習を用いて推薦対象のユーザーの好むアイテムについて学習を行い、規則を導き出す。

平成23年7月22日受付

*徳山工業高等専門学校情報電子工学科

**情報学部門

(1-2) この規則が成り立つ新たなアイテムを推薦する。
 岩濱⁴⁾や李⁸⁾は帰納学習の一手法である決定木をこれに類似した方法で利用している。

次に帰納学習を用いた協調フィルタリングの手法は次のようになる。

(2-1) 帰納学習を用いて推薦対象のユーザーの好むアイテムについて学習を行い、規則を導き出す。

(2-2) 多くのデータからこの規則が成り立つ別のユーザーを見つける。

(2-3) この別のユーザーの好むアイテムを推薦する。

(1-1) と (2-1) は共通の処理で実現可能である。本論文において、帰納学習の手法は ILP と決定木を用いる。(2-2) の部分は各ユーザーは複数のアイテムに関するデータを持っており、そのデータを規則に照らして、できるだけ多くのアイテムに規則がマッチしたユーザーを見つけることで実現する。

3. ブログからのデータ抽出

インターネット上のブログサイトからブログデータを収集した後、RSS の情報や、HTML の情報により、ルール抽出に必要なデータを取り出す。

3.1 ブログ

ブログとは Web 上の日記などの個人が記述した記事が時系列に配置されている Web コンテンツである。また、RSS 配信により XML 形式でブログの記事の情報を配信する。これによってより明確にブログの内容や更新の情報を第三者が取得できる。

3.2 データ抽出

ブログの RSS データからタイトル、カテゴリ、記事の時系列を取り出し、各ブログ記事の HTML データから、リンクと本文で使われている単語を取り出す。

記事の本文データは、形態素解析器 Sen¹²⁾ を利用して、形態素に分割し、名詞の単語のみを取り出す。その際に名詞が連続して現れる場合は結合して一つの単語として処理をする。例えば「携帯音楽プレイヤーの寿命が近い」を形態素解析し、名詞のみを取り出す場合、「携帯音楽プレイヤー」と「寿命」のみを取り出す。

4. Inductive Logic Programming

ILP とは一階述語論理に基づいた帰納学習の手法である。ILP の一般的な枠組みは、正例 E^+ と負例 E^- 、背景知識 B が節集合で与えられ、

$$\begin{cases} B \not\models E^+ \\ B \cup E^- \not\models \square \end{cases}$$

であるとき、

$$\begin{cases} H \cup B \models E^+ \\ H \cup B \cup E^- \not\models \square \end{cases}$$

を満たす仮説 H を見つけることである。本研究では ILP システムとして Prolog 上で動作する Aleph¹³⁾ を用いる。Aleph は、正例の規則性を示す仮説 H を Prolog のルール形式で出力する。

Aleph の場合、正例全てを満たす一つのルールを見つけられない場合、複数のルールを抽出して、そのルールのどれかがそれぞれの正例を満たすようにルール抽出を行う。通常はこれにより全ての正例を満たすことができるが、それでもどのルールも満たさない例外の正例が発生する場合もある。

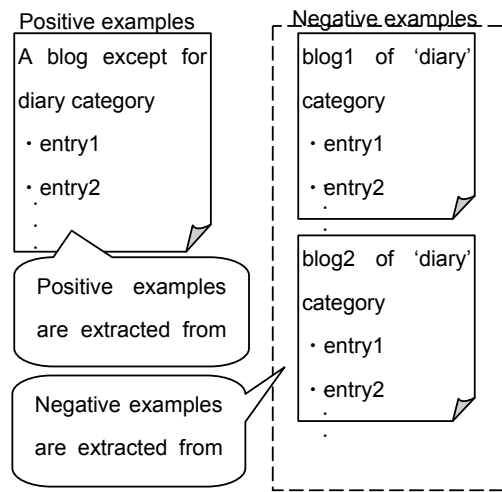


Fig. 1 Positive and negative examples.

4.1 正例と負例

本論文では、正例をある一つの着目したブログの各記事とする。この着目したブログの著者が推薦システムでの推薦対象のユーザーとなる。

Fig. 1 に正例と負例を示す。

負例は着目したブログと関係のないカテゴリのブログの記事を用いる。本研究では負例として「日記」のカテゴリのブログ記事のデータを用いる。「日記」を選ぶ理由は、ブログは一般的に Web 上の日記であると言われていること、及び、正例のカテゴリとは関係なく、特に特徴のないカテゴリという条件を満たしていると考えられることから選択した。

4.2 ブログデータからの背景知識

本研究では各ブログの抽出したデータから述語表現による背景知識を生成する。Table 1 に本研究で用いる知識表現を示す。また、ブログの一つの記事から生成された背景

Table 1 Knowledge representation in background knowledge.

Predicate	Meaning
prev(Entry1,Entry2)	Entry2 is the previous entry of Entry1.
next(Entry1,Entry2)	Entry2 is the next entry of Entry1.
link(Entry,Url)	Url is linked from Entry.
category(Entry,Cat)	Entry 's category is Cat.
title(Entry,Title)	The title of Entry is containing Title.

contain('http:// /', 'http:// / × × .html').
title('http:// / × × .html', 'アゲハチョウの観察 3').
title('http:// / × × .html', 'アゲハチョウ').
title('http:// / × × .html', '観察 3').
ord('http:// / × × .html', -2).
category('http:// / × × .html', '育児').
link('http:// / × × .html', 'http:// /').

Fig. 2 Background knowledge from an entry of blog.

next(C1,C2):- contain(B,C1),contain(B,C2),ord(C1,T1), ord(C2,T2),T1+1:=T2.
prev(C1,C2):- contain(P,C1),contain(P,C2),ord(C1,T1), ord(C2,T2),T1-1:=T2.

Fig. 3 Additional rules to background knowledge.

知識の例を Fig. 2 に示す．ここで、カテゴリやタイトルはブログの著者が自由に決められるため、カテゴリとタイトルに関しては形態素解析を行い、それに含まれる名詞もそれぞれカテゴリやタイトルとして扱う．

また、リンクに関しては、'http://ホスト名/パス/ファイル名.html' という URL ヘリンクが張られていた場合、このアドレスへのリンクの知識を生成するとともに、このページの属するサイトにリンクが張られているという意味で、ホスト名だけ残して、'http://ホスト名' ヘリンクが張られているという知識も生成する．

そして、述語 next, prev を表現するために、Fig. 3 のような追加のルールを背景知識に導入する．変数 C1, C2 はブログの記事を示し、変数 B はブログを示す．また、変数 T1, T2 はタイムスタンプを示す．従って、最初のルールは C1 と C2 は B に含まれ、C1 のタイムスタンプ T1 で、C2 のタイムスタンプは T2 であり、T1 + 1 が T2 と等しいときに、C1 の次の記事は C2 と言えることを示している．2 番目のルールは C1 と C2 は B に含まれ、C1 のタイムスタンプ T1 で、C2 のタイムスタンプは T2 であり、T1 - 1 が T2 と等しいときに、C1 の前の記事は C2 と言えることを示している．

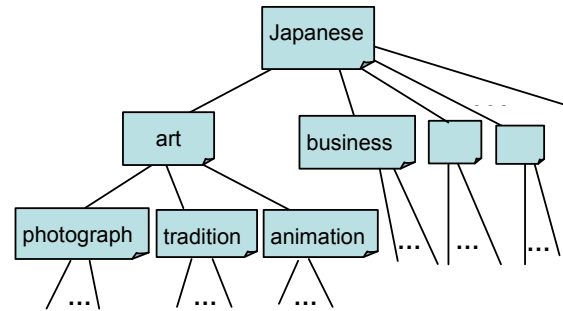


Fig. 4 Hierarchies of ODP.

4.3 ODP から得られる包含関係の背景知識

ODP(Open Directory Project)⁹⁾とは、Web ページをその話題ごとに階層的なディレクトリ (カテゴリ) に分類した世界最大の Web ディレクトリである．本研究ではこの ODP の階層構造をカテゴリ及びタイトルの包含関係の背景知識に用いる．

例えば、Fig. 4 の左端の「アート」と「写真」の関係から

$category(X, 'アート') :- category(X, '写真')$.

$title(X, 'アート') :- title(X, '写真')$.

という 2 つの知識が得られる．最初のルールは「写真」カテゴリに属するものは「アート」カテゴリに属することを意味し、2 番目のルールはタイトルが「写真」の場合、タイトルは「アート」に関するものであることを意味する．

このような包含関係の知識を導入する理由は、ブログのカテゴリやタイトルは自由に決めることができ、複数の類似のカテゴリやタイトルを同様に扱うためである．

5. 決定木

決定木は最も普及している帰納学習の手法である．多くの訓練例のデータから規則としてクラスに分類するための木構造を学習によって得る．ID3, C4.5¹¹⁾等多くの決定木システムが存在し、様々な分野で応用されている．本論文では決定木の学習にデータマイニングツール Weka¹⁴⁾に含まれる C4.5 の Java による実装 J48 を用いる

5.1 解析用データの生成

Weka に与えるデータの形式は ARFF と呼ばれる．本研究では推薦対象者のブログの記事とそれ以外の記事を分類する決定木を構築するための ARFF データを生成する．ここで、対象のブログの記事とそれ以外の記事は ILP の正例と負例に対応する．

クラス属性は、推薦対象者のブログの記事であることの本質を扱い、'blog' という名前に設定する．この値が 'yes' の場合は対象のブログ記事であることを示し、'no' の場合はそれ以外の記事であることを示す．ILP の場合と同様の

理由で「日記」カテゴリのブログ記事を対象の記事ではないとし'no'に設定する。

決定木のノードと成りうる属性については、各カテゴリ、タイトル及びリンクの有無を属性値として生成した。カテゴリ及びタイトルについては、形態素解析を行い、それに含まれる名詞もカテゴリやタイトルとして扱う。これらの属性値に加えて、ODPの階層構造から得られる包含関係により、属性値を設定する。例えば、Fig. 4のような階層構造がある場合、「写真」カテゴリの属性値が'True'ならば、「アート」カテゴリの属性値も'True'に設定し、さらに階層構造でこれを繰り返すため Japanese カテゴリの値も'True'にするまた、リンクについても ILP の場合と同様に、あるページにリンクがある場合、そのページへのリンクの有無の属性を'True'にするとともに、そのページの属するサイトにリンクが張られているという属性の値も'True'にする。

ILP と明らかに異なるのは、記事の前後関係は属性として扱っていないことである。扱わない理由は、前後関係を一つ導入する度にデータの属性が倍増していくためである。

6. 実験

ILP と決定木について実際のブログデータにより帰納学習を行い、規則の抽出を行った。そして、その規則を他のブログデータに適用し、推薦可能であるかを調べた。

6.1 実験手法

本研究の実験ではブログサイトのトップページで次のようなカテゴリ分類されているブログデータを用いる。

カテゴリ (39 種類)

- 日記, 日常, 趣味, 旅行・地域, 写真, バイク,
- 自転車, 車, ガーデニング, 鉄道,
- 経済・政治・国際, ニュース, 文化・芸術,
- 学問・資格, 英語, スポーツ, サッカー, 釣り,
- 野球, ウェブログ関連, 携帯・デジカメ,
- パソコン・インターネット, 映画・テレビ, 音楽, 書籍・雑誌, ゲーム, アニメ・コミック,
- 芸能・アイドル, ペット, 猫, 犬, 野鳥,
- グルメ・クッキング, 育児, 住まい・インテリア, 心と体, 恋愛, ダイエット, 美容・コスメ

これらのカテゴリから「日記」以外の 38 カテゴリに属するブログ (1531 件) を対象とし、1531 回の実験を行った。

ILP の場合、その一つのブログに属する記事をそれぞれ正例とし、負例は「日記」カテゴリのブログ記事 (418 件) とした。ブログを推薦可能であるかを調べるため、対象のブログを除く 1530 件のブログに対してその規則が成り立つ記事が存在するかを調べた。

決定木の場合も同様に、対象のブログに属する記事と、そうでないブログ記事(「日記」カテゴリのブログ記事 418 件)

を分類する決定木を構築する実験を 1531 件のブログを対象に、1531 回の実験を行った。その後、対象のブログを除く 1530 件のブログに対してその規則が成り立つ記事が存在するかを調べた。

ここで、ILP システム Aleph は SWI-Prolog5.6.64 上で実行し、Weka はバージョン 3.6.2 を用いた。また、実験で用いた PC は CPU が Core2Duo の 1.83GHz で、メモリ 2GB、OS は WindowsVista である。

6.2 実験結果

ILP と決定木で規則の抽出の実験を行った。

ILP の抽出された規則の例として、'旅行・地域' カテゴリのあるブログ記事を正例とした場合の抽出ルールを Fig. 5 に、'ニュース' カテゴリのあるブログの記事を正例とした場合の抽出ルールを Fig. 6 に示す。

```
[Rule 1] [Pos cover = 2 Neg cover = 0]
entry(A) :- prev(A, B),
category(B, 'グルメ・クッキング'),
category(A, '旅行・地域').
[Rule 2] [Pos cover = 2 Neg cover = 0]
entry(A) :- next(A, B),
category(B, '旅行・地域'),
category(A, 'グルメ・クッキング').
...
```

Fig. 5 Rules extracted from a blog in the 'travel/area'category using Aleph.

```
[Rule 1] [Pos cover = 2 Neg cover = 0]
entry(A) :- prev(A, B),
category(B, 'グルメ・クッキング'),
link(B, 'http:// .com').
[Rule 2] [Pos cover = 2 Neg cover = 0]
entry(A) :-
category(A, 'グルメ・クッキング'),
link(A, 'http:// .com').
[Rule 3] [Pos cover = 2 Neg cover = 0]
entry(A) :- next(A, B),
category(B, 'グルメ・クッキング'),
link(B, 'http:// .com').
[Rule 4] [Pos cover = 2 Neg cover = 0]
entry(A) :- prev(A, B),
title(B, 'イタリア').
...
```

Fig. 6 Rules extracted from a blog in the 'news' category using Aleph.

Fig. 5 のルールは、10 の記事を持つ (正例の数:10) ブログからの抽出例である。この例では抽出ルールは 4 つの

記事に当てはまるが、ルールを当てはめることができない例外の記事が6つあった。

Fig. 5 の Rule1 は、記事 A は '旅行・地域' カテゴリであり、その直前の記事 B は 'グルメ・クッキング' カテゴリであることを示している。Rule2 もほぼ同様の関係を示しているのだが、Rule2 は記事 A は 'グルメ・クッキング' カテゴリであり、その直後の記事 B は '旅行・地域' カテゴリであることを示している。また、ルールごとに 'Pos cover' と 'Neg cover' の数値が付いているが、'Pos cover' はそのルールが当てはまる正例の数を示し、'Neg cover' はそのルールが当てはまる負例の数を示している。

また、Fig. 6 のルールは、10 の記事を持つ (正例の数:10) ブログからの ILP の抽出例である。この例では8つの記事はこの抽出ルールが成り立つが、例外となるルールを当てはめることができない記事が2つあった。

具体的に Fig. 6 の Rule1 は、記事 A の前の記事 B は 'グルメ・クッキング' カテゴリであり、記事 B から 'http://.com' へのリンクがあることを示している。この URL はある有名動画サイトのアドレスである。同様に、Rule2 は、記事 A は 'グルメ・クッキング' カテゴリであり、'http://.com' へのリンクがあることを示している。Rule3 も、記事 A の次の記事 B は 'グルメ・クッキング' カテゴリであり、記事 B から 'http://.com' へのリンクがあることを示している。また、Rule4 は記事 A の前の記事 B はタイトルに 'イタリア' を含むことを示している。

```
no(428.0/10.0)
```

Fig. 7 Rules extracted from a blog in the 'travel/area' category using Weka(J48).

```
tit; イタリア = T: yes(2.0)
tit; イタリア = F
| linkto:http://.com = T
| | cat; 日記・コラム・つぶやき = T: no(3.0)
| | cat; 日記・コラム・つぶやき = F: yes(3.0)
| linkto:http://.com = F: no(420.0/5.0)
```

Fig. 8 Rules extracted from a blog in the 'news' category using Weka(J48).

次に、'旅行・地域' カテゴリの Fig. 5 と同じブログから決定木で規則を抽出した結果を Fig. 7 に示す。Fig. 7 を見ると、木構造は表示されずに、全てこの対象のブログの記事ではないことを示す 'no' が表示されている。即ち、この例では決定木は規則の抽出に失敗している。

また、'ニュース' カテゴリの Fig. 6 と同じブログから決定木で規則を抽出した結果を Fig. 8 に示し、それから得られた決定木を Fig. 9 に示す。

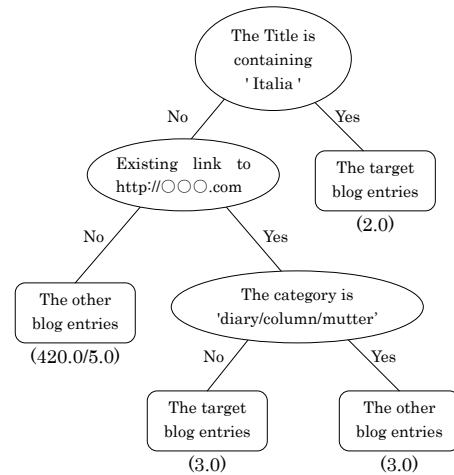


Fig. 9 A decision tree extracted from a blog in the 'news' category.

Fig. 9 で決定木の葉の下のカッコ内の数値はそこに分類された事例 (記事) の数を示し、'/' で区切られその後数値がある場合は誤分類が存在し、その数を示している。'http://.com' は ILP の場合に規則に現れた URL と同じ有名動画サイトである。5つの記事が対象の記事として正しく分類されているが、それと同数の5つの記事が誤分類されている。

次に Table 2 に ILP と決定木で、規則の抽出できたブログの数と推薦可能なブログの数を示す。ここで推薦可能なブログとは、抽出された規則を用いてその規則が成り立つ別の記事を見つけることができたブログのことである。また、この実験で1531件の規則抽出に要した時間を Table 3 に示す。

Table 2 The number of blogs from which we succeed to extract rules and the number of blogs for which we can be recommend other blogs.

Method	The number of blogs from which we succeed to extract rules	The number of blogs for which we can be recommend other blogs
ILP	1374 (89.8%)	1088 (71.1%)
Decision tree	1398 (91.3%)	1126 (73.5%)

Table 3 The times of rule extraction.

Method	The times of rule extraction(hour:minute)
ILP	2:03
Decision tree	2:11

6.3 実験についての考察

Fig. 5 の ILP の抽出ルールから想像できるが、このブログは旅行に行って食事をしたことを、食事に関しては 'グル

メ・クッキング' カテゴリの記事として、食事以外の旅先での出来事については' 旅行・地域' カテゴリの記事として書いている。ILP の場合、記事の前後関係を prev 述語や next 述語によって簡単に表現できるため、Fig. 5 のようにルール抽出に成功している。しかし、決定木の場合、前後関係はないため、2 つの傾向の異なる記事があるような場合、決定木は規則の抽出が困難となり、Fig. 7 のように決定木構築の失敗という結果となったと考えられる。

次に、Fig. 6、Fig. 8 及び Fig. 9 について考察をする。この対象のブログは、イタリア人が著者で、イタリアに関する記事や、料理の作り方の動画を動画サイトにアップロードし、それを紹介している記事を含み、そして、この著者は告知の内容を含む記事を' ニュース' カテゴリにしている。このブログについて、ILP と決定木の両手法ともに' イタリア' がタイトルに含まれるという規則は抽出できている。そして、ILP は' グルメ・クッキング' カテゴリで動画にリンクが張られていることを規則としてうまく抽出している。一方、決定木はタイトルに' イタリア' が含まれておらず、' http:// .com' にリンクが張られていて、' 日記・コラム・つぶやき' カテゴリでないことを規則として抽出している。これでは著者の嗜好が絞り込まれていない。

これは、決定木は対象でない事例の誤分類を許すが、ILP は負例を満たすルールは許さないためであると考えられる。抽出された規則は推薦に用いることを考えているため、限定された規則でも、その規則が成り立つ別の事例が見つければ問題ない。かえってより嗜好の近いブログや事例を見つければ都合である。

Table 2 から両手法とも、90%程度のブログで規則が抽出でき、さらにその規則を情報推薦に用いた場合、70%のブログで推薦を行うことができることが分かる。また、ILP のほうが決定木よりも若干その割合が低い、これは先にも書いたが、ILP は負例を満たすルールは許さないためであるため、決定木よりも厳しい規則を抽出しようとしているためだと考えられる。

ILP と決定木を比較して、決定木は規則抽出率、推薦可能率ともに若干高く、決定木の方が若干推薦可能な記事を見つけることは容易だと言える。

しかし、ILP の方が、記事の前後関係に代表されるように、規則の表現力が高い。本論文では示していないが、リンク先の情報も用いた関係を規則として抽出することも可能である¹⁾。そのため、より複雑な関係が成り立つ類似性の高い記事を見つかることが可能となる。また、Fig. 6 と Fig. 8 の例のように、決定木はブログの嗜好をうまく絞り込めていない場合もあるため、一概にこの数値が高いから決定木が良いとはいえない。

一般的に ILP の規則の抽出は探索を行うため決定木よりも遅いイメージがある。しかし Table 3 から判るが、むしろ若干 ILP のほうが速いという結果だった。本実験の ILP

に関しては探索順序を探索木の枝数が極端に増えるであろう next 述語や prev 述語を先に探索するように設定することにより決定木と同等以上のスピードで規則抽出を行うことが可能であった。

7. 推薦理由の提示についての考察

この推薦対象のユーザーのブログから抽出された規則が同じように成り立つ別のブログの記事を見つければ、その別のブログやその記事をユーザーに推薦することが可能となる。また、そのときの推薦理由も規則をもとに示すことが可能となる。

ILP で推薦を行う場合、例えば、内容ベースフィルタリングで Fig. 6 の Rule2 がマッチした記事を推薦を行えば、推薦理由として

「あなたのブログ記事と同じように' グルメ・クッキング' カテゴリの記事で <http:// .com> ヘルリンクを張っているブログ記事です。」

という文章を推薦ページと同時に示すことができる。また、協調フィルタリングでマッチした記事を含むブログを推薦する場合、推薦理由として、

「あなたと同じように' グルメ・クッキング' カテゴリの記事で <http:// .com> ヘルリンクを張っている記事を書いた人のブログです。」

という文章を推薦ページと同時に示すことができる。このようにわかりやすく平易な言葉で、推薦理由を示すことが可能となる。

Fig. 5 の Rule1 の場合も同様で、内容ベースフィルタリングの推薦理由は

「あなたのブログ記事と同じように' 旅行・地域' カテゴリの記事で、その前の記事が' グルメ・クッキング' カテゴリであるブログ記事です。」

となり、協調フィルタリングの推薦理由は

「あなたと同じように' 旅行・地域' カテゴリの記事での前の記事が' グルメ・クッキング' カテゴリである記事を書いた人のブログです。」

と説明できる。

一方、決定木の場合は、当てはまった決定木のルートから葉へのパスの属性を用いて、推薦理由を示すことができる。内容ベースフィルタリングで Fig. 8 の決定木でタイトルに' イタリア' を含むある別の記事推薦する場合、推薦理由として、

「あなたのブログ記事と同じように、タイトルに' イタリア' を含むブログ記事です。」

と言う文章を示すことができる。また、協調フィルタリングで、決定木で同じクラスに分類される別の記事を含むブログを推薦する場合、

「あなたと同じようにタイトルに' イタリア' を含む記事を書いた人のブログです。」

と推薦ブログと同時に示すことができる。

ただし、Fig. 8 の決定木でルートから葉へのパスが'tit; イタリア=F' で'linkto;http:// .com=T' で'cat; 日記・コラム・つぶやき=F' の場合、内容ベースフィルタリングの推薦理由は次のようになる。

「あなたのブログ記事と同じように、タイトルに' イタリア' を含まないで、http:// .com ヘルリンクを張っていて、' 日記・コラム・つぶやき' カテゴリではないブログ記事です。」

このように、否定の' ~でない' という理由が多く並ぶこととなり、推薦理由は冗長となり、解りにくくなる。属性値が'F' となる否定の属性を推薦理由に使用しないという方法も考えられるが、場合によっては、着目したルートから葉へのパスが否定の属性しか含まないという場合もありうる。

以上から、ILP と決定木のどちらも推薦に用いることができる。ただし、推薦理由の提示に関しては、ILP のほうがルールがそのまま理由として用いることができ、使い勝手が良い。

8. おわりに

本論文は、ILP と決定木を用いて、内容ベースフィルタリングと協調フィルタリングで情報推薦が可能であることを、実際のブログデータを用いた実験で確認した。その際に、帰納学習で得られた規則を利用することで、推薦理由を示すことが可能であり、特に ILP の抽出ルールは推薦理由を提示するのに使い勝手が良いことを示した。

今後は、推薦結果や推薦理由をユーザーに示し、どのように受け入れられるかを検証していきたい。

謝辞

本研究は科研費 (21500102) の助成を受けたものである。

参考文献

- 1) N.Chikara, M.Koshimura, H.Fujita, R.Hasegawa, Rule Extraction from Blog Using Inductive Logic Programming, wi-iat, vol. 3, pp.269-272, 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010
- 2) 古川康一, 尾崎知伸, 植野研, 帰納論理プログラミング, 共立出版, 2001
- 3) J.L.Herlocker, J.A.Konstan, and J.Riedl, Explaining Collaborative Filtering Recommendations, Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, pp. 241-250, 2000
- 4) 岩濱数宏, 土方嘉徳, 西田正吾, 決定木を用いた音楽情報フィルタリングシステムとその有効性の検証, 電子情報通信学会論文誌 D1, Vol.J88-D-I, No.3, pp.642-656, 2005
- 5) 神鷹 敏弘, 推薦システムのアルゴリズム (1), 人工知能学会誌, 22(6), pp.826-837, 2007
- 6) 神鷹 敏弘, 推薦システムのアルゴリズム (2), 人工知能学会誌, 23(1), pp.89-103, 2008
- 7) 神鷹 敏弘, 推薦システムのアルゴリズム (3), 人工知能学会誌, 23(2), pp.248-263, 2008
- 8) 李鵬, 山田誠二, 帰納学習を用いた映画推薦システム, 人工知能学会全国大会論文集, 3H2-03, 2004
- 9) "Open Directory Project", <http://www.dmoz.org/>
- 10) M.Oka, H.Mori, M.Saito, An Information Filtering Method Based on User's Moods, Situations, and Preferences, Human Interface, Part I, HCI 2007, LNCS4557, pp.715-719, 2007
- 11) J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993, 翻訳: 古川康一監訳: AIによるデータ解析, トッパン, 1995
- 12) "Sen Project", <http://ultimania.org/sen/>
- 13) A.Srinivasan, The Aleph Manual, <http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>, 1999
- 14) Weka 3 - Data Mining with Open Source Machine Learning Software in Java, <http://www.cs.waikato.ac.nz/ml/weka/>