

## ユーザ履歴を用いた TV 番組選択支援エージェントの実装と評価

中島 太郎\* 渡辺 尚\* 樽口 秀昭\*\*

\* 静岡大学, \*\* ヤマハ株式会社

放送のデジタル化に伴う多チャンネル化により、ユーザが番組選択の際に要求される労力も大きくなる。本稿ではユーザが過去に視聴した番組や入力したキーワードなどを保存した履歴データベースを用い、MBRによりユーザの過去の特性から現在入力されたキーワードの意味を推測し、番組表からユーザが見たいと予想される番組を探し出すユーザエージェントを構築する。

まずユーザが入力するキーワードのパターンを分類し、視聴履歴とユーザが現在入力したキーワードの関係を調べた。次にキーワード入力のない場合の視聴履歴も含めたユーザの視聴履歴全般を利用し、検索対象期間以外の入力がない場合にも推薦できるように拡張した。

## An Implementation of a TV Program Selection Support Agent and its Evaluation

Taro Nakajima\* Takashi Watanabe\* Hideaki Taruguchi\*\*

\*Shizuoka University, \*\*Yamaha Corporation

This paper discusses a user support system for selecting TV programs with user agent which use history database of users. This system is needed when a TV user has wide selection of programs available, because of digital broadcasting. When a user wants to see a program, the user inputs keywords for search, then this agent judge suitable programs depending on the situation from history by MBR(Memory-Based Reasoning).

In this paper, we experiment history-keyword relation. The agent use the history not only when user input keywords, but also when user only input date and span for search.

### 1 はじめに

BS デジタル放送が始まり、地上波のデジタル化も予定されている。このように放送電波がデジタル化され、それに伴って多チャンネル化が生じる [1] と、より多くの番組が提供され、ユーザの選択肢が広がる。しかし、その一方ユーザは膨大な情報から自分の見たい番組を選択する労力を求められる。そのためにユーザは各ジャンルに点数を付けてエージェントに好みを教えたり [2]、あるいはユーザの手を煩わせず、視聴時間によって番組やその出演者などの評価を決めたり [3] することでユーザを補助する方式が検討されている。しかしこれらの方式では、ユーザが自分の好みが変わるたびにそれをエージェントに教える必要がある

たり、好みの変化を反映するのが難しかったりする。そこで本稿では選択されなかったという情報も加味してユーザに番組を推薦する方法を考える。番組選択支援エージェント (Testa) はユーザの代わりにこれらの膨大な情報を処理し、ユーザが見たい番組を探すのを助けるものである。そのために Testa はユーザが取った行動を履歴として記録し、それをもとにユーザが番組検索のために入力したキーワードの意味を推測して、それを置換や補完して番組表の検索を行なう。

### 2 番組選択支援エージェント (Testa)

図 1 に本システムの概観図を示す。

ユーザは見たい番組をキーワードを入力して探す。Testa では、過去にユーザが入力したキーワード群とその際に視聴した番組に関するデータと選択されなかつ

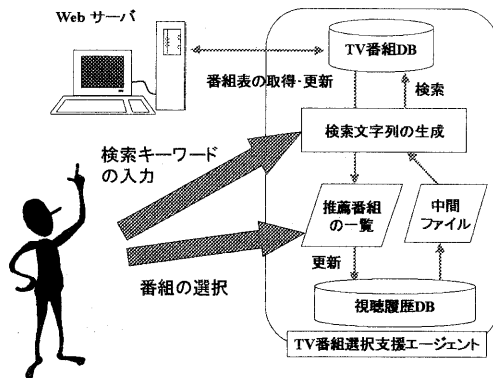


図 1: 番組選択支援エージェントの概観

た番組に関するデータを視聴履歴データベースに保存している。そこから MBR(Memory-Based Reasoning)<sup>[4]</sup>を利用して、ユーザが現在入力したキーワード以外に、ユーザが現在入力したキーワードを置換や補完して検索文字列を生成する。Web から得られた番組表からユーザが見たいと予想される番組を探すために、検索文字列で番組表を検索し、番組評価値が高い順にソートして Web から得られたその番組の情報とともにユーザに提示する。ユーザがその中からある番組を選択し、視聴すると、それらの情報を視聴履歴データベースに追加する。それら視聴履歴データベースは、またユーザがキーワードを入力した際に検索文字列を生成するために用いられる。これを携帯で利用できるように実装を行なっている。

## 2.1 視聴履歴データベース

まず、図 1 のうち、視聴履歴データベースについて詳しく述べる。視聴履歴データベースはユーザが番組を視聴したと判断できた場合に更新され、その際に記録するデータは

- ユーザが番組検索用に入力したキーワード
- ユーザが選択した番組の情報とその推薦順位、その番組評価値
- ユーザが選択しなかった番組のうち、選択した番組より上位で推薦した番組の情報とその推薦順位、番組評価値

である。また、番組の情報というのは日付、曜日、放送開始時間、放送局、ジャンル名、番組タイトル、番組内容(出演者等)といった、Web サーバから得られる情報である。これらは以下のような形式で得られる。

01: 静岡朝日テレビ

02: 1月 23 日

03: 火

04: 21:54

05: 23:09

06: ニュースステーション

07: [S]

09: ▽10年余で550店、米店との関係は？

あの中古本書店急成長の秘密…その舞台裏は

▽祝14場所ぶり優勝生出演横綱貴乃花

10: ニュースステーション

◇米店が古本を回収し始めている。なぜ米店なのか。

不況のさなか、急激に業績を伸ばし、

十年余りで五百五十店舗を出店した中古本書店が...

## 2.2 中間ファイル

次に視聴履歴データベースから作成される中間ファイルについて述べる。中間ファイルは視聴履歴データベースから検索候補文字列を生成するために利用する。前提として、ユーザがある時、ある番組を探すために入力したキーワードはその出現位置によらず、対等な関係にあり、各キーワードは結果としてユーザが視聴した番組の各データ(放送日、放送開始時刻、放送局、タイトル、放送内容など)と強い相関があるとする。

中間ファイルは前節で示した時系列に並べた視聴履歴データベースからユーザが見たい番組を探すために入力したキーワードに注目し、キーワードごとに選択した番組の情報を検索候補文字列とし、その評価値(履歴評価値)を求めたものである。

### 2.2.1 履歴評価値の算出方法

履歴評価値は以下の式により算出される。

$$E_p = G \times ON \times \sum_{i=0}^N \{(1 + NS) \times R\} / N$$

$$E_m = ON' \times \sum_{j=0}^{N'} \{(1 + NS') \times R'\} / N'$$

$$E = \left( \sum_{a=0}^C E_p - \sum_{a=0}^C E_m \right) / C$$

$E_p$  はユーザが選択した番組履歴による評価値であり、 $E_m$  はユーザが選択した番組より上位に推薦した番組が選択されなかった番組履歴による評価値である。このように本システムではユーザが選択しなかったという情報も利用する。評価値を求める際には選択履歴の順位 ( $G$ )、選択した番組履歴と選択されなかった番組履歴との不一致度 ( $NS$ )、履歴の新しさによる重み値 ( $ON$ )、番組を推薦した際の番組評価値 ( $R$ )、選択さ

れなかった番組履歴数 (N), ユーザが入力したキーワードの出現回数 (C) を評価項目とする。

### 2.3 検索文字列の生成

以上のように作成された中間ファイルの検索候補文字列から実際に番組表を検索する際に使用する検索文字列を選択する。具体的には、まず現在ユーザが入力したキーワードと中間ファイル中の過去に入力されたキーワードとの一致度を MBR を使用して計算する。

それをそのキーワードが入力された際の検索候補文字列の履歴評価値に乘じ、検索文字列とすかどうか判断する。こうして検索候補文字列から検索文字列を選択し、履歴評価値をもとにその評価値を求める。

### 2.4 番組表の検索

2.3のようにして、現在ユーザが入力したキーワードごとにその検索文字列を生成し、番組表の各番組と同じタイプ(曜日, 時間, 放送局など)ごとにその一致度を計算し、番組評価値を求める。こうすると現在のユーザが入力したキーワード数分の番組評価値が得られるので、その平均値を最終的な番組評価値として用い、ユーザに推薦する判断基準とする。

### 2.5 比較方法

ユーザが入力した曜日・時刻を、それぞれ  $n.d, n.t$  とし、履歴中の番組を視聴した曜日・時刻を、それぞれ  $p.d, p.t$  とする。

$$\Delta d(n.d, p.d) = [n.d - p.d] / 3$$

$$\Delta t(n.t, p.t) = [n.t - p.t] / 12.0$$

ここで、 $[n.d - p.d]$  は、7日間で表示した曜日のうえでの差であり、 $[n.t - p.t]$  は、24時間表示の時刻のうえでの差である。従って、月曜日と金曜日の差は3であり、1:00と23:00の差は2.00である。よって、 $3 \geq [n.d - p.d] \geq 0$ ,  $12.0 \geq [n.t - p.t] \geq 0$ となる。

また放送内容などでの文字列の比較ではまず、文字列の区切り文字として

"◇" "▽" "「" "」" "..." " " " ? " " " を考える。

その上で、

1. 2つの文字列が連続して一致する最大数を数え、短い方の文字数で割る。
2. 2つの文字列にともにある文字を数え、文字数の大きな方で割る。
3. 上で計算した2つのうちの大きな方を2つの文字列間の一致度とする。

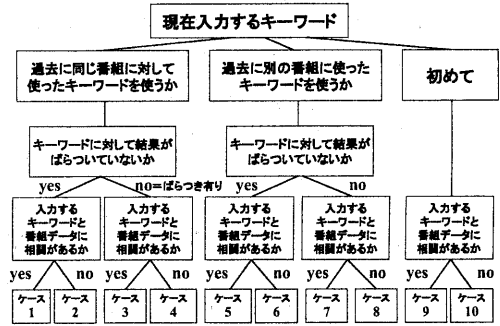


図 2: 現在のユーザ入力の種類

の方法を使って比較を行なう。

## 3 サンプル視聴履歴を用いた評価実験

### 3.1 現在のユーザ入力の分類

ユーザが検索用に入力したキーワードに注目すると、ユーザが現在入力したキーワードは図2のように分類できる。

現在ユーザが入力したキーワードは、まずそのキーワードを過去に使用したことがあるかないかで分類される。さらに過去に使用したことがあるキーワードは、今回探したい番組を選択、視聴するのに使用したことがあるかないかで分類される。これらは視聴履歴が目的の番組を検索するのに有効に働くか、かえって害を及ぼすか、あるいは役に立たないかの違いによる。

次に過去に使用したキーワードの特徴により分類できる。そのキーワードを入力した際にユーザが視聴した番組の種類が1つなら、そのキーワードと番組には強い相関があるといえる。逆にそのキーワードを入力した際に視聴した番組の種類が多ければ多い程そのキーワードと番組を結びつけて考えるのが難しくなる。

また現在のキーワードが目的の番組を探すのに有効に働くかどうかで分類できる。現在のキーワードが文字列であった場合は、有効度は番組情報がキーワードと字面的に近いものを含むかどうかによるため、Webサーバから得られる番組情報とユーザがその番組に対して認識しているキーワードが一致しているか、しないによる。

以上の組合せにより現在ユーザが入力するキーワードは10パターンに分類できる。

### 3.2 1回目の推薦順位についての実験

3人の被験者に対して3週間分の視聴履歴を収集する実験を行なった。次に収集した視聴履歴を用いて以下のように本方式の評価実験を行なった。

- 最初の2週間分の視聴履歴を視聴履歴データベースの初期データとし、それらは全て1位で推薦され、選択したものと仮定する。
- 最初の2週間分の視聴履歴の各番組に対する番組評価値(=PV:Past Value)を0.3,1.0,3.0,6.0とする。
- 現在ユーザが入力したキーワードの評価値(=NV:Now Value)を中間ファイルの評価値(履歴評価値)に基づいて、

方式1 各キーワードごとの最有力置換候補が持つ履歴評価値の平均値

方式2 方式1の値の1/2倍

方式3 方式1の値の2倍

方式4 中間ファイル中の最大の履歴評価値

方式5 方式4の値の1/2倍

方式6 方式4の値の2倍

方式7 定数 1.0

の各場合を比較する。

- 現在ユーザが入力したキーワードの評価値と過去の番組評価値の組み合わせにより、最後の1週に視聴した番組が、実際に視聴した日の何番組に推薦されるかを調べる。

#### 3.2.1 実験結果

- ケース1, ケース2  
視聴履歴において既に、現在ユーザが入力したキーワードと目的の番組との間に強い相関がある場合に於ける。現在ユーザが入力したキーワードが有効なケース1の場合はもちろん、ケース2の場合でも希望の順位で推薦できた。
- ケース3  
ケース1と比較すると視聴履歴において、現在ユーザが入力したキーワードと目的の番組との間の相関が弱い場合であるが、出現頻度の高い番組については希望の順位で推薦できた。逆に出現頻度の低い番組についてはPVを重視するほど出現頻度の高い番組の影響が強くなり、上位で推薦するのは難しくなった。
- ケース4  
ケース3より目的の番組を検索するのがさらに困難な場合である。現在ユーザが入力したキー

ワードが当てにならず、視聴履歴に頼る場合である。この場合はNVを重視しない方式2と方式5が他に比べて上位で推薦できた。

- ケース5, ケース7  
視聴履歴が目的の番組を検索するのに邪魔になる場合である。この場合はNVを重視した方式3と方式6が他に比べて上位で推薦できた。
- ケース6, ケース8, ケース10  
これらのケースでは現在ユーザが入力したキーワードからも視聴履歴からも目的の番組を推測しにくいいため、推薦することが難しい。
- ケース9  
このケースは現在ユーザが入力したキーワードのみによる検索になりやすい。そのため、推薦順位は現在ユーザが入力したキーワードがどれだけ目的の番組を特徴付けられるかによった。

#### 3.2.2 実験の考察

以上のように全て1位で視聴したと仮定して初期データを与え、その番組評価値(PV)と現在ユーザが入力したキーワードの評価値(NV)を変えて推薦順位を調べた。PVを強調すると良い結果が得られるケースがあれば、NVを強調すると良い結果が得られるケースもあり、すべてのケースについて共通の最適な組み合わせはなかった。PVもNVも強調しすぎない、PVが1.0で、NVが方式1または方式4の組合せが適切であった。

### 3.3 2回目以降の推薦順位についての実験

次に同じ番組を繰り返し視聴する場合の推薦順位について実験を行なった。ある番組を視聴する際に入力するキーワードは図2のように分類できた。同じキーワードを使って同じ番組をまた視聴する場合、図2のケース9はケース1に、ケース10はケース2にそれぞれ該当する。同様にケース5と7はケース3に該当し、ケース6と8はケース4に該当する。ケース1から4はそのままケース1から4に該当する。また前節より、ケース1と2は希望の推薦順位で推薦できた。したがって、ここではケース3と4の場合について実験する。

実験方法としては、1回目の時と同じ視聴履歴を用い、3週目のユーザの視聴履歴をトレースする。このとき1回目の推薦順位の実験結果を踏まえ、PVとして1.0と3.0を使い、NVとして方式1と方式4を使う。

### 3.3.1 実験結果

#### ケース 3

図3にこのケースの例を示す。このような場合、出現頻度の高い「ニュースステーション」に対しては1回目からうまく推薦できた。ここではそれより出現頻度の低い「ニュース7」がどれくらいの順位で推薦されるかを実験した。

検索用に入力されたキーワード	ユーザに提示した番組	ユーザが選択した番組
ニュース	1位 1月15日 21:54 1.0点 テレビ朝日 ニュースステーション	
...		
初期去2週間の	ニュース	1位 1月10日 21:54 1.0点 テレビ朝日 ニュースステーション
...		
過去1週の	ニュース	1位 1月20日 19:00 1.0点 NHK総合 ニュース7
...		
ニュース	1位 1月21日 19:00 1.0点 NHK総合 ニュース7	
×2週間分(月曜から金曜までと土日で違うニュース番組を視聴)		
番組目的の	ニュース	7位 2月4日 19:00 2.0点 NHK総合 ニュース7

図 3: ケース 3 の具体例

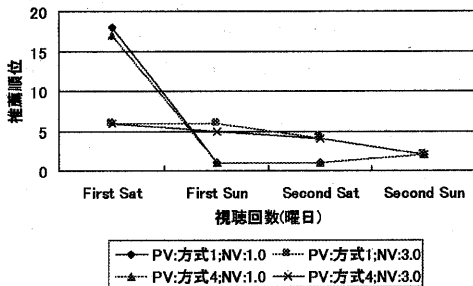


図 4: ケース 3 : "ニュース" の入力 で「ニュース7」を視聴する場合

図3のように月曜から金曜までは「ニュースステーション」を視聴し、土曜と日曜には「ニュース7」を視聴するのを2週間続けた場合の「ニュース7」の結果を図4に示す。NVとして方式1, 方式4を用いた場合のいずれも、PVとして1.0を用いた場合の1回目の推薦順位は3.0を用いた場合より低い、2回目以降はそれより上位で推薦できた。

#### ケース 4

図5にこのケースの例を示す。ここでは出現頻度はどの番組も同じであるが、入力キーワード"ニュース

検索用に入力されたキーワード	ユーザに提示した番組	ユーザが選択した番組
ニュース	1位 1月15日 21:54 1.0点 テレビ朝日 ニュースステーション	
...		
過去2週間の	ニュース	1位 1月15日 22:54 1.0点 SBS NEWS23
...		
過去1週の	ニュース	1位 1月15日 23:20 1.0点 テレビ朝日 ニュースJAPAN
...		
ニュース	1位 1月19日 21:54 1.0点 テレビ朝日 ニュースステーション	
...		
ニュース	1位 1月19日 22:54 1.0点 SBS NEWS23	
...		
ニュース	1位 1月19日 23:20 1.0点 テレビ朝日 ニュースJAPAN	
×2週間分(月曜から金曜まで3つのニュース番組を視聴)		
番組目的の	ニュース	7位 2月29日 22:54 2.0点 SBS NEWS23

図 5: ケース 4 の具体例

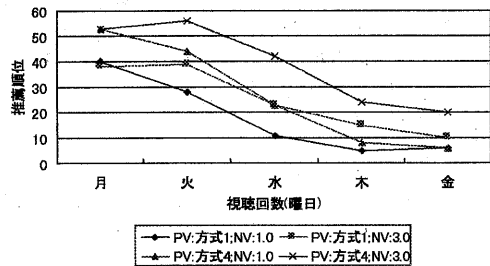


図 6: ケース 4 : "ニュース" の入力 で「筑紫哲也NEWS 23」を視聴する場合

"が有効でない「筑紫哲也NEWS 23」がどれくらいの順位で推薦されるかを実験した。

図6に図5と同じように月曜から金曜まで3つのニュース番組の視聴を繰り返した場合の「筑紫哲也NEWS 23」の推薦順位の変化を示す。1回目の推薦順位はどれも高くはないが、NVとして方式1, PVとして1.0を用いた場合は他に比べて1回目の推薦順位も高く、2回目以降も徐々に上位で推薦できるように変化した。またこの間「ニュースステーション」, 「ニュースJAPAN」は約3位で推薦されていた。

### 3.3.2 実験の考察

ケース4の結果では、他の組合せと比較して1回目も2回目以降も推薦順位の高いNVとして方式1, PVとして1.0を用いた場合が適切であった。この組合せでは、ケース3の1回目の推薦順位は他の組合せより低い。しかし、ユーザ側の視点に立つと、ユーザには入力したキーワードの有効度は分からないので、ケース3とケース4のどちらも出現頻度に応じた順位で推

薦されることが望ましい。ケース3の1回目の推薦順位は、それより出現頻度の高いケース4の1回目の推薦順位よりは高い。したがって、NVとして方式1、PVとして1.0を用いた場合が適切であるといえる。

#### 4 ユーザの視聴履歴全般への拡張

本節では、前節までのようにユーザが番組内容の特徴付けるキーワードを入力した場合だけでなく、検索対象期間の入力だけで番組内容などの特徴を示すキーワードの入力がなくても、視聴履歴から番組を推薦できるように Testa を拡張する。また、キーワード入力があった場合 (=NF:No Feature) を1まとめにし、あった場合 (=FI:Feature Input) と合わせて利用する。NF と FI の一致度を仮定することにより、現在入力されたキーワードが FI の場合には NF の視聴履歴も利用して検索文字列を生成し、逆に現在入力されたキーワードが NF の場合には FI の視聴履歴も利用して検索文字列を生成する。

##### 4.1 評価実験

3人の被験者に対して以下のように本方式の評価実験を行なった。

- 2週間分の番組表から実際に視聴した番組とキーワード(あれば)を記入してもらい、初期DBを作る。
- NF と FI の一致度として、0,0.5,1の固定値を用いる。
- 初期DBをもとに1週間実験した。

以下にユーザごとに FI に属するキーワードを入力した場合、しなかった場合、それらの合計での、選択順位の平均値を示す。

NFとの一致度		0	0.5	1
入力あり	User A	29	83	33
	User B	61	78	42
	User C	102	84	166
入力なし	User A	7.3	6.0	13.5
	User B	14.0	7.6	5.8
	User C	8.1	8.1	14.3
合計	User A	3.6	7.6	6.7
	User B	7.2	7.7	4.8
	User C	9.0	8.2	15.3

図 7: NF と FI の一致度と平均選択順位

##### 4.2 実験結果の考察

ユーザによって結果が良い NF と FI の一致度が異なった。前節での実験はユーザが視聴した番組が何位

で推薦されるかという、ユーザの視聴履歴の後追いであったのが、本節での実験では実際に Testa が推薦番組の一覧をユーザに提示し、ユーザが視聴したい番組を選択するのでユーザの様々な視聴特性が影響した。

#### 5 おわりに

番組選択支援エージェント (Testa) は多くの番組の中からユーザが希望する番組を推測し、提示する。そのためにユーザが入力したキーワードや視聴した番組に関するデータだけでなく、選択した番組より上位に推薦した番組に関するデータも視聴履歴データベースに保存しておき、この情報も使ってユーザが入力したキーワードを推測する。

今後はユーザの TV 番組の視聴全般に本システムを適用できるように検討する。

#### 参考文献

- [1] 金 淵培, “エージェント技術の放送への応用” 映像情報メディア学会誌 Vol.52, No.4, pp.447-451, 1998
- [2] 村崎 康博, 柴田 正啓, 金 淵培, 山田 一郎, “エージェントテレビのための自動番組選択手法の検討” 1998年映像情報メディア学会冬季大会, p83
- [3] 矢川 雄一, 鶴飼 ひろみ, 田中 哲雄, 柴田 正啓, 金 淵培, “個人の嗜好に合ったテレビ番組を自動編成するエージェントの検討” 信学技報 AI98-55, pp9-16, 1998
- [4] C.Stanfill, D.Waltz, “Toward Memory-Based Reasoning” Communication of ACM, vol.29, no.12, pp.1213-1223, 1986