

協調フィルタリングに基づくソフトウェア機能推薦システム

大杉直樹[†] 門田暁人[†]
森崎修司^{††} 松本健一[†]

高機能アプリケーションソフトウェアには、ユーザはその存在に気付いていないが有用な機能(未知有用機能)が多数存在する。本論文では、協調フィルタリングを用いることで、個々のユーザに対して未知有用機能の候補を厳選して提示(推薦)するシステムの提案、および、その評価を行う。提案システムでは、複数のユーザから機能実行履歴を自動収集し、それらに協調フィルタリングのアルゴリズムを適用することで、各ユーザに対して、有用度が高いと推定される順に未使用機能を推薦する。本論文で新たに提案する2つのアルゴリズム(Rank correlation アルゴリズム, Sequence-based アルゴリズム)を含む6つのアルゴリズムを用いた場合の推定の精度をNDPM(Normalized Distance-based Performance Measure)により評価した結果、すべてのアルゴリズムが、ランダムに推薦順序を決定した場合より精度が高く、未知有用機能の候補を厳選してユーザに提示できることを確認した。以上のことから、提案システムは、高機能アプリケーションソフトウェアにおける未知有用機能の発見に役立つと期待される。

Software Function Recommender System Based on Collaborative Filtering

NAOKI OHSUGI,[†] AKITO MONDEN,[†] SHUJI MORISAKI^{††}
and KEN-ICHI MATSUMOTO[†]

High-Functionality Applications (HFA) includes a large amount of useful yet unbeknown functions. This paper proposes a Software Function Recommender System based on *Collaborative Filtering (CF)* and evaluates the accuracies of the system's recommendations. The proposed system recommends useful yet unbeknown functions to each user, as follows: the proposed system automatically collects histories of software function execution from many users, then estimates the usefulness of unused functions using CF algorithms, and recommends users with the functions ranked by their estimated usefulness. We conducted experimental evaluation of six CF algorithms including 2 new algorithms (Rank correlation algorithm and Sequence-based algorithm) by using NDPM (*Normalized Distance-based Performance Measure*). The result of the experiments showed that the average NDPM of all the algorithms were better than that of randomly produced recommendation. This suggested the proposed system can be used for discovering useful yet unbeknown functions of HFA.

1. はじめに

今日、ワープロソフトや表計算ソフトなどのアプリケーションソフトウェアは多様なユーザのニーズを満たすために非常に多くの機能を提供するようになってきている⁵⁾。しかし、個々のユーザが実際に使用する機能は、提供される機能のごく一部にすぎない。Microsoft Office XP の各アプリケーションを対象に

筆者らが行った調査でも、22名のユーザそれぞれが1年間に使用した機能の割合は、それらアプリケーションが提供する機能の2~15%であった。

ユーザが使用しなかった機能(未使用機能)には、そのユーザにとって既知であるが有用でない機能とともに、ユーザはその存在に気づいていないが使用すれば有用な機能(未知有用機能)も多数含まれている可能性がある⁵⁾。Microsoft Word 2000 と PowerPoint 2000 を対象に著者らが行った実験でも、それらソフトウェア上での機能実行履歴を被験者9名の間で相互参照可能としたところ、各被験者は平均16.1個の未知機能を発見し、そのうち約4割が有用であったと答えている¹¹⁾。

[†] 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute
of Science and Technology

^{††} 株式会社インターネットイニシアティブ
Internet Initiative Japan, Inc.

著者らが開発し、実験で用いたシステムは、機能実行履歴を自動収集するとともに、各被験者に対して、未使用機能のうち「他被験者が1回以上実行している機能」を未知有用機能の候補として提示できる¹¹⁾。ただし、実験において被験者に提示された候補数は、1人あたり63.7個にも達した。実験では、未使用機能のうち「他被験者が1回以上実行している機能」を未知有用機能の候補として各被験者に提示した¹¹⁾。この方式は、候補選出が容易である反面、被験者数(機能実行履歴を提供し、未知機能発見支援を受けようとするユーザ数)が増加すれば、提示される候補数が著しく増加するという問題がある。高性能アプリケーションソフトウェア(High-Functionality Application: HFA)における、より効果的な未知機能発見支援を実現するためには、未知有用機能の候補をより厳選してユーザに提示する方式やシステムの開発が必要である。ただし、HFAを使用するユーザは多種多様であるため、ユーザの多様性(類似性)を考慮して各機能の候補を厳選することが望ましい。

本論文では、著者らが開発した「機能実行履歴を用いたソフトウェア機能発見支援システム^{10),11)}」に協調フィルタリングの手法を導入することで、個々のユーザにとってより厳選された未知有用機能候補を提示(推薦)するシステムの提案、および、その実験的評価を行う。協調フィルタリングは、大量の情報の中から有用な情報を選び出す情報フィルタリングの一手法である。一般には、他のユーザの意見(情報の有用度に対する評価値)に基づいて「ユーザ間の類似度評価」と「情報有用度推定」を行い、有用度の高い情報をユーザに提示(推薦)する¹²⁾。提案システムでは、ユーザの意見の代わりに、ユーザによる機能実行履歴、特に、機能ごとの実行頻度を用いて「ユーザ間の類似度評価」と「未使用機能の有用度推定」を行い、有用度が高いと推定される順に未使用機能をユーザに提示(推薦)する。ソフトウェアの機能実行履歴は自動収集が容易であり、提案システムでも、履歴収集から機能推薦に至るすべての処理を自動化している。一方、

「ユーザ間の類似度評価」と「情報有用度推定」のアルゴリズムはこれまでに多数提案されている^{1),3),18)~20)}が、4.5節で述べるように、機能の実行頻度には異常値(他と大きく離れた値)が含まれるため、どのようなアルゴリズムの精度が高いのか明らかでない。そこで、提案システムでは、異常値を回避するための新たに提案する2つのアルゴリズム(Rank correlation アルゴリズム, Sequence-based アルゴリズム)を含む6つのアルゴリズムを実装し、それらの精度評価実験を行う。

以降、2章では、協調フィルタリングの概要を述べ、3章では、提案システムの構成と機能推薦手順を述べる。4章では、提案システムで利用可能な協調フィルタリングアルゴリズムを紹介し、5章では、それらアルゴリズムによる機能推薦精度の評価実験について述べる。6章では、評価実験の結果について考察を行い、7章では、関連研究を紹介する。8章では、まとめと今後の課題を述べる。

2. 協調フィルタリングのアルゴリズム

協調フィルタリングは、大量の情報の中から有用な情報を選び出す情報フィルタリングの一手法である¹²⁾。ここで、情報とは、記事、ウェブページ、書籍、楽曲、映画作品など、その有用度(関心度)がユーザによって異なるもの(アイテム)である。また、「有用な情報を選び出す」とは、ユーザにとって未知のアイテム(未知アイテム)の有用度を推定し、有用度の高いアイテムをそのユーザに提示(推薦)することである。

協調フィルタリングの特徴は、未知アイテムの有用度推定において、他ユーザの意見(既知アイテムに対する有用度評価値: vote 値)を利用する点にある¹²⁾。すなわち、協調フィルタリングへの入力、各ユーザの各アイテムに対する vote 値を要素とする行列(vote 値行列)であり、その出力は、あるユーザにとって未知のアイテムの vote 値(vote 値行列の特定の要素)の推定値である(図1参照)。本論文では、vote 値行列から未知アイテムの vote 値を推定し、有用度による順位付けを未知アイテムに対して行うアルゴリズムのことを、協調フィルタリングのアルゴリズム、あるいは単に、推薦アルゴリズムと呼ぶこととする。

協調フィルタリングの手法が初めて導入された Netnews 記事推薦システム Tapestry では、記事の推薦を受けようとするユーザ(システムユーザ)は、記事に対する関心が自分と似ている他ユーザ(類似ユーザ)を指定する必要があった⁶⁾。推薦アルゴリズムは単純で、システムユーザの未知記事のうち、類似ユーザが

文献 11) のシステムでは、「機能実行履歴のユーザ間での相互参照」が可能であり、「ある機能を1回以上実行した他ユーザの割合により未使用機能をソートし表示すること」も可能である。「他ユーザの情報を利用するかどうか」という観点からすれば、協調フィルタリング技術を用いたシステムといえる。ただし、「ユーザ間の類似度評価」機能や「情報有用度推定」機能は有しておらず、個々のユーザにとってより厳選された未知有用機能候補を推薦することはできない。本論文における「協調フィルタリングの手法の導入」とは、これら2つの機能をシステムに付加することを意味する。

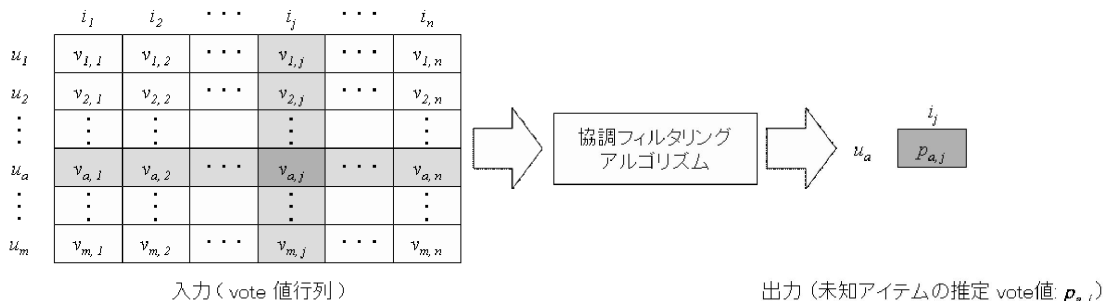


図 1 協調フィルタリングアルゴリズムに対する入出力
Fig. 1 Input and output of collaborative filtering algorithm.

表 1 GroupLens と提案システムの比較
Table 1 Comparison between GroupLens and the proposed system.

	GroupLens	提案システム
推薦アイテム	未知機能	未使用機能
入力	vote (記事の有用度に対する評価値, 5 段階評価)	機能の実行頻度 (0.0~1.0 の実数, 自動収集が容易)
出力	未知記事に対する推定 vote 値	未使用機能の有用度の推定値

高い vote 値を与えている記事が推薦される。ウェブページ推薦システム PHOAKS では、類似ユーザを指定する必要はないが、推薦アルゴリズムは比較的単純で、システムユーザの未知ページのうち、vote 値を与えているユーザが多いページが推薦される²¹⁾。

Usenet 記事推薦システム GroupLens では、まず、vote 値行列に基づいて、システムユーザと他ユーザとの類似度 (記事に対する関心の類似度) の推定が行われる¹⁸⁾。そして、推定した類似度で重み付けされた vote 値行列を用いて推薦記事が決定される。類似度推定の精度が高くなれば、システムユーザが類似ユーザを指定した場合と同様の推薦が可能となる。

提案システムにおける推薦アルゴリズムでは、ソフトウェア機能に対するユーザの vote 値は用いず、代わりに、ユーザによるソフトウェア機能実行履歴、特に、機能ごとの実行頻度を用いている (表 1 参照)。多数のソフトウェア機能それぞれの有用度をユーザに評価してもらうことは現実的ではない。機能実行履歴は自動収集が容易であり、推薦作業全体の自動化にも適している。なお、推薦作業の自動化を目指し、提案システムでは、システムユーザが類似ユーザを指定する方式はとっていない。利用可能な 6 つの推薦アルゴリズムのうち 4 つでは、システムユーザと他ユーザとの類似度推定を行っている。

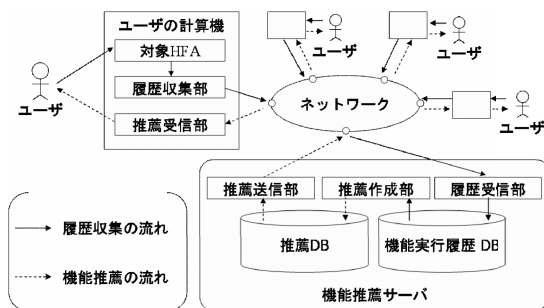


Fig. 2 Architecture of the recommender system for software function discovery.

3. 機能推薦システム

3.1 構成

提案する機能推薦システムは、図 2 に示すように、ユーザの計算機上のクライアントソフトウェア (履歴収集部、推薦受信部) と機能推薦サーバ (推薦送信部、推薦作成部、履歴受信部) から構成される。クライアントソフトウェアは本システムを用いるすべてのユーザの計算機にインストールされる。機能推薦サーバはユーザの計算機とネットワークで接続された環境に 1 台設置され、各ユーザから送信された機能実行履歴を蓄積する「機能実行履歴 DB」と、各ユーザへの推薦 (順位付けされた機能の集合) を蓄積する「推薦 DB」を具備する。

なお、本システムは、著者らが開発した「機能実行履歴を用いたソフトウェア機能発見支援システム¹¹⁾」を機能拡張したものである。システム構成上の最も大きな差異は、「推薦作成部」と「推薦DB」がサーバに追加された点である。これにより、サーバからユーザへは、機能実行履歴ではなく、推薦作成部が作成し、推薦DBに蓄積されている推薦(順位付けされた機能の集合)が送信されることになる。ユーザへ送信される推薦は機能実行履歴と比べて著しくサイズが小さいため、文献 11) のシステムと比べて、より多くのユーザによる利用に適している。

3.2 システムの動作

3.2.1 機能実行履歴の収集

機能推薦システムは、まず、複数のユーザからネットワークを介して機能実行履歴を収集する(図2中の実線の矢印)。手順は次のとおりである。

1. ユーザが「対象HFA」を使用している間、ユーザの計算機上で稼動する「履歴収集部」が、ユーザが実行した機能を監視し、機能実行履歴として一時的に記録する。
2. 「履歴収集部」は、ユーザが「対象HFA」を終了した際に、一時的に記録した機能実行履歴をサーバへ送信する。
3. サーバ上で稼動する「履歴受信部」が、各ユーザの計算機から送信されてきた機能実行履歴を受信し「機能実行履歴DB」に蓄積する。

機能実行履歴の例を図3に示す。これは、提案システムを用いて Microsoft Office XP (Word, Excel, PowerPoint) から収集したものである。この機能実行履歴では、1行が、1回の機能実行に対応する。各行に

は、左から、機能のID番号、機能が実行された日時、実行された機能の名称、がそれぞれ記録される。なお、機能実行にショートカットキーが使用された場合は、その名称が右端に記録される。たとえば、図3の先頭行の場合、ID番号7024の「クリア->すべて(&C)」が、2002年9月6日の14時26分3秒にショートカットキー Del によって実行されたことを表す。

3.2.2 推薦の作成

図2の機能推薦サーバ上の「推薦作成部」において、各ユーザに対する推薦(順位付けされた機能の集合)を協調フィルタリングにより作成し、推薦DBへ蓄積する。推薦作成のタイミングは、新たな機能実行履歴を受信したときである。つまり、機能実行履歴DBが更新されるたびに、全ユーザに対する推薦を再算出し、推薦DBを更新する。

推薦の作成手順を図4に示す。この図では、システムの全ユーザを表す集合を $U = \{u_1, u_2, \dots, u_m\}$ とし、ユーザ u_a に対する推薦が次の3つの処理を経て作成されている。なお、全機能を表す集合を $F = \{f_1, f_2, \dots, f_n\}$ 、 u_i による機能 f_j の実行頻度を $q_{i,j}$ とする。

処理1. 各ユーザの「機能実行履歴」中に含まれる各機能の実行頻度を次式で計算し、「機能実行頻度

```

7024 2002/09/06 14:26:3 クリア->すべて(&C) Del
3517 2002/09/06 14:26:24 書式設定->囲み線(&C)
128 2002/09/06 14:26:26 編集->元に戻す(&U) Ctrl+Z
3517 2002/09/06 14:26:32 書式設定->囲み線(&C)
3518 2002/09/06 14:26:33 書式設定->文字の網かけ(&C)

```

図3 機能実行履歴の例

Fig. 3 An example of the usage history.

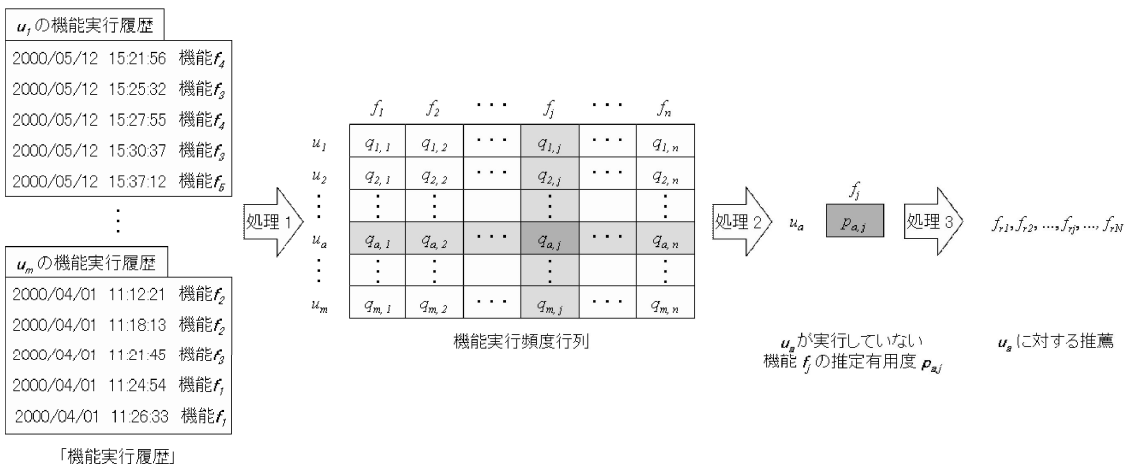


図4 推薦作成の処理手順(ユーザ u_a に対する推薦の作成)

Fig. 4 Procedure of making a recommendation for user u_a .

行列」を作成する．

$$q_{i,j} = \frac{e_{i,j}}{\sum_{f_k \in F_i} e_{i,k}} \quad (1)$$

ただし，ユーザ u_i による機能 f_j の実行回数を $e_{i,j}$ ，ユーザ u_i が過去に実行した機能の集合を F_i と表すとする．

処理 2. 協調フィルタリングのアルゴリズムを適用し，ユーザ u_a が実行していない機能 f_j の有用度 $p_{a,j}$ を推定する（アルゴリズムの詳細は，4 章で説明する）．

処理 3. 処理 2 をユーザ u_a が実行していないすべての機能について行う．有用度で機能を順位付けし，推薦を作成する．

3.2.3 推薦の送信と提示

機能推薦システムは，推薦 DB に蓄積された各ユーザへの推薦をユーザに送信し，提示する（図 2 中の点線の矢印）．手順は次のとおりである．

1. サーバ上の「推薦送信部」は，ユーザからの要求に応じて，推薦 DB に含まれる最新の推薦を取り出し，ユーザの計算機に送信する．
2. ユーザの計算機上で稼動する「推薦受信部」は，サーバから推薦を受信し，ユーザへ提示する．

4. 提案システムにおける推薦アルゴリズム

本章では提案システムで利用可能な 6 つのアルゴリズムについて説明する．既存アルゴリズムを提案システムに応用した 4 つのアルゴリズムを 4.1 から 4.4 節に示す．また，本論文で新たに提案する 2 つのアルゴリズムを 4.5，4.6 節に示す．

4.1 User count アルゴリズム

Terveen ら²¹⁾ の web ページ推薦システム PHOAKS で使用されたアルゴリズムを，提案システムに応用したものである．このアルゴリズムでは，多くのユーザに使用されている機能を優先的に推薦する．

ユーザの集合を U ，ユーザ数を $|U|$ ，ユーザ u_i による機能 f_j の実行頻度を $q_{i,j}$ と表すと， u_a に対する f_j の有用度 $p_{a,j}$ は次式で定義される．

$$p_{a,j} = \frac{\sum_{u_i \in U} \text{binaryOf}(q_{i,j})}{|U|} \quad (2)$$

ただし，

$$\text{binaryOf}(q_{i,j}) = \begin{cases} 0 & (q_{i,j} = 0.0) \\ 1 & (q_{i,j} > 0.0) \end{cases} \quad (3)$$

式 (3) において，関数 $\text{binaryOf}(q_{i,j})$ は， u_i が f_j

を 1 回も実行していない場合は 0，1 回以上実行した場合は 1 になる．式 (2) の右辺は， f_j を実行したユーザの割合と等しくなる．結果として，より多くのユーザに使用されている機能ほど，有用度 $p_{a,j}$ が大きな値となる．

4.2 Average アルゴリズム

Shardanand ら²⁰⁾ の楽曲推薦システム Ringo で使用されたアルゴリズムを提案システムに応用したものである．全ユーザ間で平均的に高い頻度で実行されている機能を優先的に推薦するアルゴリズムである． u_a に対する f_j の有用度 $p_{a,j}$ は次式で定義される．

$$p_{a,j} = \frac{\sum_{u_i \in U} q_{i,j}}{|U|} \quad (4)$$

式 (4) の右辺は， f_j の実行頻度をすべてのユーザについて加算し，ユーザ数 $|U|$ で除算することを示す．結果として，より平均的に高い頻度で実行されている機能ほど，有用度 $p_{a,j}$ が大きな値となる．

4.3 Correlation coefficient アルゴリズム

Resnick ら¹⁸⁾ の GroupLens で使用されたアルゴリズムを提案システムに応用したものである．まず，推薦対象のユーザ u_a とその他の各ユーザ u_i について，ユーザ間の類似度 $\text{sim}(u_a, u_i)$ を次式で求める．

$$\begin{aligned} \text{sim}(u_a, u_i) &= \frac{\sum_{f_k \in F_a \cap F_i} (q_{a,k} - \bar{q}_a)(q_{i,k} - \bar{q}_i)}{\sqrt{\sum_{f_k \in F_a \cap F_i} (q_{a,k} - \bar{q}_a)^2} \sqrt{\sum_{f_k \in F_a \cap F_i} (q_{i,k} - \bar{q}_i)^2}} \end{aligned} \quad (5)$$

ただし， u_i が実行した機能の集合を F_i ， F_i に含まれる機能の数を $|F_i|$ と表す．また， u_i の実行頻度の平均値 \bar{q}_i は次式で定義される．

$$\bar{q}_i = \frac{1}{|F_i|} \sum_{f_l \in F_i} q_{i,l} = \frac{1}{|F_i|} \left(\because \sum_{f_l \in F_i} q_{i,l} = 1 \right) \quad (6)$$

ユーザ u_a に対する機能 f_j の有用度 $p_{a,j}$ は次式で定義される．

$$p_{a,j} = \bar{q}_a + \frac{\sum_{u_i \in U} \text{sim}(u_a, u_i)(q_{i,j} - \bar{q}_i)}{\sum_{u_i \in U} \text{sim}(u_a, u_i)} \quad (7)$$

式 (5) は， u_a ，ならびに， u_i による f_k の実行頻度 $q_{a,k}$ と $q_{i,k}$ の相関係数を計算する式であり，計算結果は $[-1, 1]$ の範囲の値をとる．また，式 (7) の第 2

項は、 u_i による f_j の実行頻度から、 u_i の実行頻度の平均値を減算したものを、類似度 $sim(u_a, u_i)$ によって重み付けして、全ユーザについて総和をとったものである。結果として、類似度が高いユーザが高い頻度で実行している機能ほど、有用度 $p_{a,j}$ が大きな値となる。

4.4 Binary correlation アルゴリズム

Breese ら³⁾ がテレビ番組推薦システムで使用したアルゴリズムを、提案システムに応用したものである。まず、推薦対象のユーザ u_a とその他の各ユーザ u_i について、ユーザ間の類似度 $sim(u_a, u_i)$ を次式で求める。

$$sim(u_a, u_i) = \frac{\sum_{f_k \in F_a \cup F_i} \{binaryOf(q_{a,k}) \times binaryOf(q_{i,k})\}}{\sqrt{\sum_{f_k \in F_a \cup F_i} binaryOf(q_{a,k})} \sqrt{\sum_{f_k \in F_a \cup F_i} binaryOf(q_{i,k})}} \tag{8}$$

ただし、 u_i が実行した機能の集合を F_i と表すとし、関数 $binaryOf(q_{i,j})$ は式 (3) で定義したものを使用する。ユーザ u_a に対する機能 f_j の有用度 $p_{a,j}$ は式 (7) で定義される。

4.5 Rank correlation アルゴリズム

本論文で新たに提案するアルゴリズムで、類似度を順位相関係数によって求める。4.3 節に示した Correlation coefficient は相関係数で類似度を計算するため、実行頻度に異常値（他と大きく離れた値）が存在する場合、類似度が極端に大きくなるという問題がある⁹⁾。GroupLens など、従来の推薦システムでは、vote 値（1 から 5 の整数値など）が用いられるため、異常値は発生しない。ここでは、スピアマンの順位相関係数⁹⁾ を用いて類似度計算を行うことで、この問題を回避する。

まず、推薦対象のユーザ u_a とその他の各ユーザ u_i について、ユーザ間の類似度 $sim(u_a, u_i)$ を次式で求める。

$$sim(u_a, u_i) = \frac{\left\{ \sum_{f_k \in F_a \cap F_i} rankOf(q_{a,k})^2 + \sum_{f_k \in F_a \cap F_i} rankOf(q_{i,k})^2 - d^2 \right\} \times \frac{n}{2} - T^2}{\sqrt{n \sum_{f_k \in F_a \cap F_i} (rankOf(q_{a,k})^2 - T^2)} \times \sqrt{n \sum_{f_k \in F_a \cap F_i} (rankOf(q_{i,k})^2 - T^2)}} \tag{9}$$

ただし、 u_i が実行した機能の集合を F_i と表すと

し、 n を $F_a \cap F_i$ に含まれる機能の数とする。また、 $T = n(n+1)/2$ とし、 $d^2 = \sum_{f_k \in F_a \cap F_i} (rankOf(q_{a,k}) - rankOf(q_{i,k}))^2$ とする。 $rankOf(q_{i,k})$ は、 $F_a \cap F_i$ に含まれる機能の実行頻度の順位（整数値）を求める関数である。ユーザ u_a に対する機能 f_j の有用度 $p_{a,j}$ は式 (7) で定義される。

4.6 Sequence-based アルゴリズム

本論文で新たに提案するアルゴリズムである^{13)~15)}。4.5 節に示した Rank correlation アルゴリズムでは、実行頻度に基づく順位付けが全機能を通じて行われ、ユーザ間の類似度評価に用いられる。実行頻度の似通った機能が多数存在すると、その周辺では、実行頻度のわずかな違いが順位の大きな違いとなり、類似度を大きく低下させることになる。ここでは、Yao²²⁾ が提案した関数 dpm を用いてユーザ間の類似度を計算する。まず、推薦対象のユーザ u_a とその他の各ユーザ u_i について、ユーザ間の類似度 $sim(u_a, u_i)$ を次式で求める。なお、式 (10) の右辺第 2 項は、後述する $NDPM$ の定義式 (13) を応用したものである。 $NDPM$ は「ユーザ u_a がシステムに求める理想の推薦 O_a 」と「システムがユーザに行う推薦 O_a' 」の差を評価するものであり、両者の類似度を表す式と考えることができる。ただし、他のアルゴリズムにおける類似度の評価式 (5), (8) の値域が $[-1, +1]$ であるのに対し、式 (13) の値域は $[0, 1]$ である。そこで、値域を同じとするため、 $1 - 2 \times$ (式 (13) 右辺) としている。

$$sim(u_a, u_i) = 1 - 2 \times \frac{\sum_{f_k \in F_a \cap F_i} \sum_{f_l \in F_a \cap F_i} dpm(q_{a,k}, q_{a,l}, q_{i,k}, q_{i,l})}{\sum_{f_k \in F_a \cap F_i} \sum_{f_l \in F_a \cap F_i} dpmNormalizer(q_{a,k}, q_{a,l})} \tag{10}$$

ただし、 F_a, F_i は、それぞれユーザ u_a, u_i が実行した機能の集合を表すとし、

$$dpm(q_{a,k}, q_{a,l}, q_{i,k}, q_{i,l}) = \begin{cases} 0 & \begin{pmatrix} (q_{a,k} > q_{a,l}) \wedge (q_{i,k} > q_{i,l}) \\ \vee (q_{a,k} = q_{a,l}) \wedge (q_{i,k} = q_{i,l}) \\ \vee (q_{a,k} < q_{a,l}) \wedge (q_{i,k} < q_{i,l}) \end{pmatrix} \\ 1 & \begin{pmatrix} (q_{a,k} = q_{a,l}) \wedge (q_{i,k} \neq q_{i,l}) \\ \vee (q_{a,k} \neq q_{a,l}) \wedge (q_{i,k} = q_{i,l}) \end{pmatrix} \\ 2 & \begin{pmatrix} (q_{a,k} > q_{a,l}) \wedge (q_{i,k} < q_{i,l}) \\ \vee (q_{a,k} < q_{a,l}) \wedge (q_{i,k} > q_{i,l}) \end{pmatrix} \end{cases} \tag{11}$$

$$dpmNormalizer(q_{a,k}, q_{a,l}) = \begin{cases} 1 & (q_{a,k} = q_{a,l}) \\ 2 & (q_{a,k} \neq q_{a,l}) \end{cases} \quad (12)$$

であるとする。

式 (11) に示す関数 dpm では、 $F_a \cap F_i$ に含まれる機能の組 (f_k, f_l) について、ユーザ u_a による (f_k, f_l) の実行頻度 $(q_{a,k}, q_{a,l})$ と、 u_i による (f_k, f_l) の実行頻度 $(q_{i,k}, q_{i,l})$ の大小関係を比較する。 $(q_{a,k}, q_{a,l})$ と $(q_{i,k}, q_{i,l})$ の大小関係が同じ場合 $dpm = 0$ である。両者の大小関係が逆の場合 $dpm = 2$ である。 $(q_{a,k}, q_{a,l})$ および $(q_{i,k}, q_{i,l})$ のどちらか一方だけに大小関係がない場合 $dpm = 1$ である。このように、 dpm は 2 つの機能間で実行頻度の大小を評価するだけである。実行頻度に異常値 (他と大きく離れた値) が存在しても、類似度が極端に大きくなることはない。また、全機能を通じての順序付けが行われないため、実行頻度のわずかな違いから生じた大きな順位差が類似度を大きく低下させることもない。

式 (12) に示す関数 $dpmNormalizer$ は、式 (10)、右辺第 2 項の分子値を $[0, 1]$ の範囲とするための関数である。この関数によって、式 (10) は、式 (5) と同様に $[-1, 1]$ の範囲に正規化される。ユーザ u_a に対する機能 f_j の有用度 $p_{a,j}$ は式 (7) で定義される。

5. 精度評価実験

5.1 精度評価基準

提案システムの精度を実験的に評価するための基準として、Yao²²⁾ が提案した $NDPM$ (*Normalized Distance-based Performance Measure*) を用いた。 $NDPM$ はユーザ u_a がシステムに求める理想の推

薦 O_a と、システムがユーザに行う推薦 O_a' との差異を表す値であり、協調フィルタリングや情報検索などのアルゴリズムの精度 (正しく推定する性能) 評価に広く用いられている^{1),2),17)}。 $NDPM$ は $[0, 1]$ の範囲の実数値をとり、値が小さいほど O_a と O_a' の差異が小さい、すなわち、推薦の精度が良いことを表す。また、中間の値である 0.5 はランダムに推薦を作成した場合 (ランダム推薦) の $NDPM$ の理論値である。ランダム推薦の $NDPM$ は、未知有用機能の候補をすべて提示する場合の $NDPM$ と理論的に等価であり、推薦アルゴリズムが最低限備えるべき精度である。

$NDPM$ の定義は次のとおり。

$$NDPM(O_a, O_a') = \frac{\sum_{o_k \in O_a \cup O_a'} \sum_{o_l \in O_a \cup O_a'} dpm(o_{a,k}, o_{a,l}, o_{a,k}', o_{a,l}')}{\sum_{o_k \in O_a \cup O_a'} \sum_{o_l \in O_a \cup O_a'} dpmNormalizer(o_{a,k}, o_{a,l})} \quad (13)$$

ただし、 $o_{a,k}$ は O_a における機能 k の推薦順位 (rank) を表し、 $o_{a,k}'$ は O_a' における機能 k の推薦順位を表す。また、関数 dpm と関数 $dpmNormalizer$ はそれぞれ式 (11)、(12) を用いる。

5.2 実験データ

実験に用いた機能実行履歴を表 2 に示す。履歴を収集したユーザ (被験者) の半数は、著者らが所属する大学の研究室の学生および教官であり、残りの半数は、ソフトウェア開発企業の社員である。大学の研究室では、MS Word を学術論文の作成に、MS PowerPoint を口頭発表スライドの作成に、MS Excel を実験データの統計処理に、主に用いていた。ソフトウェア開発企業では、MS Word をソフトウェア仕様書の作成に、

表 2 評価実験に使用した機能実行履歴
Table 2 Usage histories for experimental evaluation.

	実験 1		実験 2	
	Word 2000	Word 2002	PowerPoint 2002	Excel 2002
対象 HFA	Word 2000	Word 2002	PowerPoint 2002	Excel 2002
ユーザ数	6	23	21	20
最大収集期間	22 ヶ月	14 ヶ月	10 ヶ月	14 ヶ月
最小収集期間	3 ヶ月	1 ヶ月未満	1 ヶ月未満	1 ヶ月
平均収集期間	12.0 ヶ月	5.0 ヶ月	4.6 ヶ月	5.5 ヶ月
最大実行機能数	108	143	162	77
最小実行機能数	62	16	29	15
平均実行機能数	90.0	67.3	90.9	42.0
実行された機能の種類	209	227	254	132

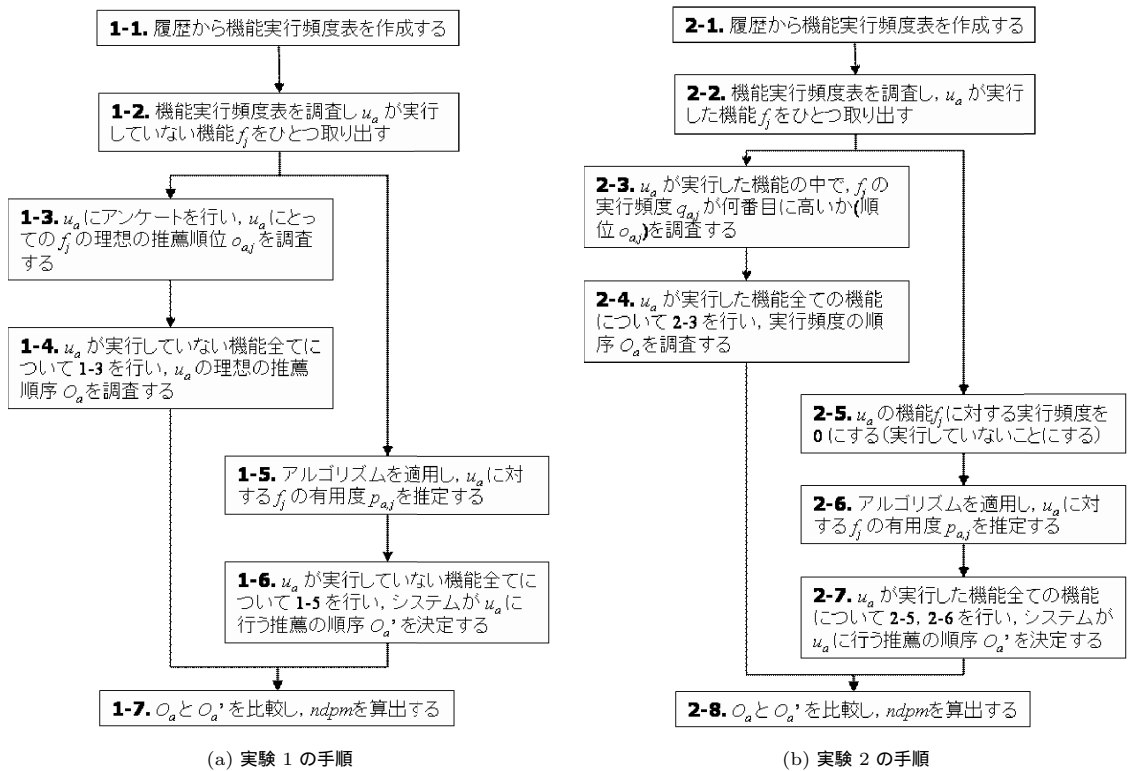


図 5 精度評価実験の手順 (システムがユーザ u_a に行う推薦の精度を評価する例)
 Fig. 5 An experimental protocol in evaluating a recommendation for user u_a .

MS PowerPoint を社内会議で使用する場合の口頭発表スライドの作成に、MS Excel を仕様書の作成に、主に用いていた。

5.3 実験手順

5.3.1 実験 1 (インタビューに基づく評価)

ユーザ u_a の理想の推薦 O_a を、各ユーザへのインタビューにより作成した。これを、システムの推薦 O_a' と比較し、NDPM を算出した。 O_a の作成にインタビューを用いる評価方法は、各ユーザが真に望む推薦をシステムの評価に反映できるという利点がある。

表 2 に示すとおり、対象 HFA は MS Word 2000、被験者は 6 名である。評価したアルゴリズムは、User count, Average, Correlation coefficient, Sequence-based の 4 つである。

実験 1 における NDPM の算出手順を図 5 (a) に示す。図中、 F は機能実行履歴に含まれる機能の集合を、 F_a は u_a が過去に実行した機能の集合を表す。この手順では、理想の推薦 O_a をユーザへのインタビューにより作成するが、その基準は次のとおり。

- u_a が機能 f_j を知らない場合、まず、実験者からその機能の目的や働きについての説明を受ける。次に、 f_j がシステムから推薦されたことと仮定し、今

後予想される実行頻度 (今後どのくらいの頻度でその機能を使うとユーザ自身が感じているか) によって順位付けを行う。実行頻度が同程度になると予想される機能は、同じ順位を付けることができる。

- u_a が f_j を知っており、かつ、履歴収集以前に使用したことのある場合、最下位の順位とする。
- u_a は f_j を知っているが、使用したことはない場合、最下位から 2 番目の順位とする。

5.3.2 実験 2 (履歴に基づく評価)

ユーザ u_a が過去に実行した機能を頻度順に並べたものを理想の推薦 O_a と見なし、システムの推薦 O_a' と比較し、NDPM を算出した。ユーザが過去に実行しなかった機能についての精度は評価できないが、 O_a の作成にインタビューが不要であるため、より多くのユーザによる実験が可能である。この方法は、従来、協調フィルタリングのアルゴリズムの精度評価に広く用いられてきた^{3), 18)~20)}。

表 2 に示すとおり、対象アプリケーションは Word 2002, PowerPoint 2002, Excel 2002 であり、被験者は 20~23 名である。評価したアルゴリズムは、実験 1 で用いたアルゴリズムに Binary correlation, Rank

表 3 実験 1 (インタビューに基づく評価) における *NDPM* の値
Table 3 Experimental result 1 (interview based evaluation).

被験者	アルゴリズム			
	User count	Average	Correlation coefficient	Sequence-based
#1	0.2701	0.2702	0.2654	0.2811
#2	0.5868	0.5868	0.5553	0.4268
#3	0.3358	0.3371	0.3288	0.3373
#4	0.4388	0.4401	0.4179	0.3873
#5	0.3664	0.3532	0.3533	0.3454
#6	0.4243	0.3894	0.3778	0.3497
平均	0.4037	0.3961	0.3831	0.3546

表 4 実験 2 (履歴に基づく評価) における平均 *NDPM* の値
Table 4 Experimental result 2 (history based evaluation).

アルゴリズム	Word 2002	PowerPoint 2002	Excel2002
Rank correlation	0.2842	0.2345	0.2618
Sequence-based	0.2858	0.2345	0.2619
Binary correlation	0.2883	0.2366	0.2638
Average	0.2891	0.2374	0.2666
Correlation coefficient	0.2910	0.2376	0.2664
User count	0.3109	0.2758	0.2912

correlation を加えた 6 つである。実験 1 では、ショートカットキーによって実行された機能の履歴を収集していなかったが、実験 2 では収集している。実験 2 における *ndpm* の算出手順を図 5 (b) に示す。

5.4 実験結果

実験 1 における、各アルゴリズムの各被験者に対する *NDPM* の値、および、その平均値を表 3 に示す。いずれのアルゴリズムにおいても、ランダム推薦 (*NDPM* = 0.5) よりも *NDPM* の平均値が小さい、すなわち、平均的に良い推薦が得られた (有意水準 5% で有意差あり)。ユーザに注目すると、5 名のユーザはすべてのアルゴリズムにおいてランダム推薦より良い推薦が得られたが、ユーザ 2 に限り、Sequence-based 以外のアルゴリズムにおいてランダム推薦より悪い推薦が得られた。また、アルゴリズムに注目すると、User count → Average → Correlation coefficient → Sequence-based の順に、より良い推薦 (より低い平均 *NDPM*) が得られた。ただし、標本数が少ないこともあり、Sequence-based と他のアルゴリズムの間に有意差は確認されなかった。

実験 2 における、各アルゴリズムの *NDPM* の平均値を表 4 に示す。いずれのアルゴリズムにおいても、ランダム推薦よりも平均的に良い推薦が得られた (有意水準 5% で有意差あり)。また、いずれのソフトウェア、および、いずれのアルゴリズムにおいても、全ユーザがランダム推薦より良い推薦を得た。いずれのソフトウェア (Word2002, Excel2002, PowerPoint2002) においても、User count → Correlation coefficient → Average → Binary correlation → Sequence-based → Rank correlation の順に、より良い推薦 (より低い平均 *NDPM* 値) が得られた。なお、表 4 中の各セルの平均 *NDPM* 値が、その直下のセルの平均 *NDPM* の値よりも (有意水準 5% で) 有意に小さい場合、そのセルの平均 *NDPM* 値を斜体として強調表示している。

6. 考察

実験 1, 2 を通して、いずれのアルゴリズムを用いた場合でも、ランダム推薦より平均的に良い推薦が得られた。今後、フィールドテストを行い、システム運用上の問題を解決していく必要はあるものの、未知有

用機能の候補をより厳選してユーザに提示するという本論文の目的に対して、協調フィルタリングに基づいて機能を推薦するという本システムのアプローチの効果が確認できた。以降では、各アルゴリズムの差異について考察する。

ユーザ間の類似度に基づいて推薦を作成する4つのアルゴリズムのうち、Binary correlation, Sequence-based, Rank correlationの3つは、類似度計算を行わない2つのアルゴリズム (User count, Average) よりも高い精度を示した。このような差が生じた原因は、後者は全ユーザに同じ推薦を作成するのに対し、前者は、各ユーザの機能実行履歴に応じて、ユーザごとに異なる推薦を作成することにあると考えられる。5.2節で述べたように、各ユーザは、必ずしも同じ作業目的にHFAを使用していないため、作業の類似(すなわち、過去の機能実行履歴の類似)を考慮した推薦を行うことが望ましい。ただし、類似度計算を行わないアルゴリズムは計算時間が短く、スケーラビリティの点では優れている。

Correlation coefficientは、ユーザ間の類似度に基づいて推薦を行うにもかかわらず、実験2においては、類似度計算を行わないUser count, Averageと同程度の精度しか得られなかった。この原因は、Correlation coefficientにおける類似度が、実行頻度の相関係数に基づいて計算されることにある。相関係数は異常値(大きく離れた値)に大きく影響を受けるという特徴がある⁹⁾。HFAの機能には、著しく高い頻度で実行される機能(「上書き保存」、「元に戻す」など)が存在し、他の機能が類似度にほとんど反映されなくなる傾向にある(「上書き保存」、「元に戻す」などの実行頻度だけが類似度計算に影響を与える)。類似度計算を行う他のアルゴリズム(Binary correlation, Sequence-based, Rank correlation)は、機能の実行頻度を類似度計算に直接用いないため、異常値の影響を受けない。

7. 関連研究

協調フィルタリングを用いたものではないが、HFAの機能を推薦するシステムとして、Owen¹⁶⁾のDYK (Did You Know)システムや、Horvitzら⁷⁾の“Tip of the Day”がある。これらのシステムは、ユーザがソフトウェアを起動した直後に、HFAが持つ機能の1つをランダムに提示する。ヘルプシステムとは異なり、ユーザが検索語を入力する必要はなく、その存在を想像すらできない機能であってもユーザが発見できる可能性がある。ただし、提案システムとは異なり、個々のユーザにとって有用な機能を厳選し、有用度が

高い順に並べて推薦することはできない。

HFAの機能実行履歴を収集し、機能推薦を行うシステムとしては、Fischerら⁴⁾のActive Helpシステムや、Horvitzら⁸⁾のMicrosoft Office Assistantがある。これらのシステムは、機能実行履歴に基づいてユーザが現在行っている作業を推定し、その作業に対して有用な機能を推薦する。一方、提案システムは、これらのシステムと異なり、より長期間の作業における機能の有用度を推定し、「覚えておくに役立つ」機能を推薦する。提案システムと、Active HelpシステムやMicrosoft Office Assistantは推薦を行うタイミングが異なるため、両者を並行して使用することも可能である。提案システムと、これらのシステムの連携について、さらなる議論が必要である。

8. まとめ

本論文では、HFAの機能の推薦に、協調フィルタリングを用いるシステムを提案した。提案システムは、複数のユーザから機能実行履歴を自動収集し、それらに協調フィルタリングのアルゴリズムを適用することで、各ユーザに対して、有用度が高いと推定される順に未使用機能を推薦できる。6つのアルゴリズムを提案システムに用いた場合の精度をNDPMにより評価した結果、すべてのアルゴリズムが、ランダムに推薦順序を決定した場合より精度が高くなり、未知有用機能の候補を厳選してユーザに提示できることを確認した。

各アルゴリズムの精度については、ユーザ間の類似度計算を行う3つのアルゴリズム(Rank correlation, Sequence-based, Binary correlation)が、類似度計算を行わない2つのアルゴリズム(User count, Average)よりも高い精度を示した。特に、各ユーザが実行した機能の順序(実行頻度の大きな順番に並べたもの)に基づいて類似度計算を行うアルゴリズム(Rank correlation, Sequence-based)が、最も高い精度を示した。また、Correlation coefficientはユーザ間の類似度計算を行うが、異常値(実行頻度の著しく高い機能)の影響を強く受けたため、類似度計算を行わないアルゴリズムと同程度の精度しか得られなかった。

提案システムにおける推薦アルゴリズムでは、ソフトウェア機能に対するユーザのvote値は用いず、代わりに、ユーザによるソフトウェア機能実行履歴、特に、機能ごとの実行頻度を用いた。これにより、推薦作業の自動化が可能となったが、同時に、実行頻度の高い機能ほど有用度も高いと推定するシステムとなっている。しかし、現実には、「たまにしか使わない(実

行頻度は低い)が,非常に役に立つ機能」が HFA には存在し,そのような機能を重点的に知りたいとするユーザもいる.実行頻度以外の情報を利用した推薦アルゴリズムの検討も重要であるが,たとえば,「ユーザ間の類似度評価」は実行頻度に基づいて行いが,「未使用機能の有用度推定」を実行頻度そのものではなく,機能を実行したかどうかだけに着目して行えば,「実行頻度は低い」が,多数の類似ユーザが利用している機能」ほど有用度を高く推定することも可能である.推薦アルゴリズムの中心となる「ユーザ間の類似度評価」と「未使用機能の有用度推定」をうまく組み合わせ,より多様な機能推薦方式を検討することも今後の課題の1つである.

参 考 文 献

- 1) Balabanovic, M. and Shoham, Y.: Fab: Content-based Collaborative Recommendation, *Comm. ACM*, Vol.40, No.3, pp.66-72 (1997).
- 2) Balabanovic, M.: An Adaptive Web Page Recommendation Service, *Proc. 1st International Conference on Autonomous Agents (Agents '97)*, pp.378-385 (1997).
- 3) Breese, J.S., Heckerman, D. and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, pp.43-52 (1998).
- 4) Fischer, G., Lemke, A.C. and Schwab, T.: Knowledge-based Help Systems, *Proc. CHI '85 Conference on Human Factors in Computing Systems*, pp.161-167 (1985).
- 5) Fischer, G.: User Modeling in Human-Computer Interaction, *User Modeling and User-Adapted Interaction (UMUAI)*, Vol.11, No.1/2, pp.65-86 (2001).
- 6) Goldberg, D., Nichols, D., Oki, B.M. and Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry, *Comm. ACM*, Vol.35, No.12, pp.61-70 (1992).
- 7) Horvitz, E.: Agents with Beliefs: Reflections on Bayesian Methods for User Modeling, *Proc. 16th Intl. Conference on User Modeling (UM '97)*, pp.441-442 (1997).
- 8) Horvitz, E., Breese, J., Heckerman, D., Hovel, D. and Rommelse, K.: The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users, *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, pp.256-265 (1998).
- 9) 池田 央: 統計ガイドブック, 新曜社 (1984).
- 10) 森崎修司, 門田暁人, 松本健一, 井上克郎, 鳥居宏次: 機能実行履歴を用いたソフトウェア利用知識の共有, *情報処理学会論文誌*, Vol.41, No.10, pp.2770-2781 (2000).
- 11) 森崎修司, 白石裕美, 大和正武, 門田暁人, 松本健一, 鳥居宏次: 機能実行履歴を用いたソフトウェア機能の発見支援システム, *電子情報通信学会論文誌 D-I*, Vol.J84-D-I, No.6, pp.755-767 (2001).
- 12) 森田昌宏, 速水治夫: 情報フィルタリングシステム, *情報処理*, Vol.37, No.8, pp.751-758 (1996).
- 13) Ohsugi, N., Monden, A. and Matsumoto, K.: A Recommendation System for Software Function Discovery, *Proc. 9th Asia Pacific Software Engineering Conference (APSEC2002)*, pp.248-257 (2002).
- 14) Ohsugi, N., Monden, A. and Morisaki, S.: Collaborative Filtering Approach for Software Function Discovery, *Proc. 2002 Intl. Symposium on Empirical Software Engineering (ISESE2002)*, Vol.2, pp.29-30 (2002).
- 15) 大杉直樹, 門田暁人, 松本健一, 森崎修司: ソフトウェア機能の推薦システムのための協調フィルタリング, *ソフトウェアシンポジウム 2002 論文集*, pp.15-24 (2002).
- 16) Owen, D.: Answers First, Then Questions, *User-Centered System Design, New Perspectives on Human-Computer Interaction*, pp.361-375, Lawrence Erlbaum Associates, Inc. (1986).
- 17) Pretschner, A. and Gauch, S.: Ontology Based Personalized Search, *Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence*, pp.391-398 (1999).
- 18) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94)*, pp.175-186 (1994).
- 19) Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms, *Proc. 10th Intl. World Wide Web Conference (WWW10)*, pp.285-295 (2001).
- 20) Shardanand, U. and Maes, P.: Social Information Filtering: Algorithms for Automating 'Word of Mouth', *Proc. CHI '95 Conference on Human Factors in Computing Systems*, pp.210-217 (1995).
- 21) Terveen, L., Hill, W., McDonald, D. and Creter, J.: PHOAKS: A System for Sharing Recommendations, *Comm. ACM*, Vol.40, No.3, pp.59-62 (1997).

- 22) Yao, Y.Y.: Measuring Retrieval Effectiveness Based on User Preference of Documents, *J. Am. Soc. Inf. Sci.*, Vol.46, No.2, pp.133-145 (1995).

(平成 15 年 5 月 16 日受付)
(平成 15 年 11 月 4 日採録)



大杉 直樹 (学生会員)

平成 13 年奈良工業高等専門学校専攻科電子情報工学専攻修了。平成 15 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。現在、同大学院情報科学研究科博士後期課程在籍。修士 (工学)。エンピリカルソフトウェア工学, 特に, ソフトウェア信頼性/開発コスト予測, ソフトウェアプロセス改善, ヒューマンインタフェースに興味を持つ。電子情報通信学会, IEEE 各会員。



門田 暁人 (正会員)

平成 6 年名古屋大学工学部電気学科卒業。平成 10 年奈良先端科学技術大学院大学博士後期課程修了。同年同大学情報科学研究科助手。平成 15 年 Auckland 大学客員研究員。博士 (工学)。電子情報通信学会, 日本ソフトウェア科学会, IEEE, ACM 各会員。ソフトウェアメトリクス, ソフトウェアセキュリティ, ヒューマンインタフェースの研究に従事。



森崎 修司 (正会員)

平成 13 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年株式会社インターネットイニシアティブ入社。博士 (工学)。インターネットを介したシームレスな知識共有, 脆弱性を意識した情報システム開発, ヒューマンインタフェースに興味を持つ。



松本 健一 (正会員)

昭和 60 年大阪大学基礎工学部情報工学科卒業。平成元年同大学院博士課程中退。同年同大学基礎工学部情報工学科助手。平成 5 年奈良先端科学技術大学院大学助教授。平成 13 年同大学教授。工学博士。エンピリカルソフトウェア工学, 特に, プロジェクトデータ収集/利用支援, ソフトウェアプロセス, ウェブユーザビリティ等の研究に従事。電子情報通信学会, IEEE, ACM 各会員。