

## ソフトウェアコンポーネント推薦における 協調フィルタリングの効果

亀井 靖 高<sup>†1</sup> 角田 雅 照<sup>†1</sup> 柿 元 健<sup>†2</sup>  
大杉 直 樹<sup>†3</sup> 門 田 暁 人<sup>†1</sup> 松 本 健 一<sup>†1</sup>

本論文では、汎用性の高い/低いコンポーネントに対する、協調フィルタリングを用いたコンポーネント推薦（亀井ら，2006年）の効果を明らかにするために、2つの仮説，(1) 汎用性の高いコンポーネントに対する推薦精度が従来手法（ランダム推薦法，平均値に基づく推薦法）より高い，(2) 汎用性の低いコンポーネントに対する推薦精度が従来手法より高い，を実験的に検証した。推薦精度を調べるために、29件のプロジェクト（使用されているコンポーネントの総数は2,558個）を用いて評価実験を行った。その結果，仮説(2)が支持され，協調フィルタリングを用いたコンポーネント推薦は汎用性の低いコンポーネントに対して特に効果がある（NDPMの中央値が0.55から0.33へ改善した）ことが分かった。

### The Effect of Collaborative Filtering on Software Component Recommendation

YASUTAKA KAMEI,<sup>†1</sup> MASATERU TSUNODA,<sup>†1</sup>  
TAKESHI KAKIMOTO,<sup>†2</sup> NAOKI OHSUGI,<sup>†3</sup>  
AKITO MONDEN<sup>†1</sup> and KEN-ICHI MATSUMOTO<sup>†1</sup>

To clarify the effect of collaborative filtering (CF) on recommending high-generality / low-generality software components, we experimentally verified two hypotheses; (1) the recommendation accuracy of CF for high-generality components is better than that of conventional methods (random algorithm and user average algorithm) and (2) the recommendation accuracy of CF for low-generality components is better than that of the conventional methods. We evaluated recommendation accuracy of CF with a dataset containing 29 open source software development projects (including 2,558 used components). As a result, the hypothesis (2) was supported, and the recommendation accuracy of

CF showed better performance than the conventional methods and the median of NDPM was improved from 0.55 to 0.33 for low-generality components.

#### 1. はじめに

近年，Microsoft や Sun Microsystems などの開発プラットフォームベンダから，数多くのソフトウェアコンポーネント（以降，コンポーネント）が提供されており，開発者はコンポーネントを利用することで，品質の高いソフトウェアを効率良く開発することが可能である<sup>2),3)</sup>。しかし，提供されるコンポーネントの数が膨大となり，有用なコンポーネントを発見しにくいという問題が生じている<sup>2),3)</sup>。

この問題を解決するための手法として，本論文では協調フィルタリングに着目する。協調フィルタリングは，大量に存在するアイテム（書籍や楽曲など）の中からユーザの好みに合うと予測されるアイテムを選出して推薦する手法であり，書籍の推薦などに適用した事例が報告されてきた<sup>1)</sup>。本論文では，「開発状況が似ているプロジェクトどうしは，有用であるコンポーネントも似ているであろう」と仮定し，開発中のプロジェクトで有用であろうコンポーネントの推薦に協調フィルタリングを適用することを考える。具体的には，類似するプロジェクトで使用しており，開発中のプロジェクトで未使用のコンポーネントを推薦する。なお，本論文ではコンポーネントをオブジェクト指向におけるクラスの意味で用いる。

従来の書籍などのアイテムと異なり，コンポーネントには汎用性が高く多くのプロジェクトで用いられるコンポーネントと，特定の用途にのみ用いる汎用性が低いコンポーネントがある。汎用性が低いコンポーネントは，開発者にとっては未知の可能性があり，推薦対象としては汎用性が高いコンポーネントよりも重要であるが，協調フィルタリングによる推薦の効果は従来明らかでない。

そこで本論文では，2つの仮説，(1) 汎用性の高いコンポーネントに対する推薦精度が従来手法（コンポーネントを無作為に推薦するランダム推薦法，および，平均値に基づく推薦手法）より高い，(2) 汎用性の低いコンポーネントに対する推薦精度が従来手法より高い，

<sup>†1</sup> 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

<sup>†2</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information and Science Technology, Osaka University

<sup>†3</sup> 株式会社 NTT データ  
NTT Data Corporation

を実験的に検証する\*1。推薦精度を調べるために、29件のプロジェクト（使用されているコンポーネントの総数は2,558個）それぞれに対して、あるバージョンの開発状況（コンポーネントの使用回数）を基に、次期バージョンで用いられようコンポーネントを推薦し、実際に用いられているかどうかを調べた。

## 2. 協調フィルタリングを用いたコンポーネント推薦方法

協調フィルタリング推薦法では、書籍などのアイテムを協調フィルタリングによって推薦する考え方をコンポーネント推薦に適用し、「開発状況が似ているプロジェクトどうしは、有用とするコンポーネントも似ているであろう」と仮定する。すなわち、類似するプロジェクトで使用しており、開発中のプロジェクトで未使用なコンポーネントを推薦する。

表1は協調フィルタリング推薦法で用いるデータ形式であり、表1の $p_i$ は $i$ 番目のプロジェクト（従来の協調フィルタリングでは、ユーザ）を、 $c_j$ は $j$ 番目のコンポーネント（書籍）を表す。また、 $r_{i,j}$ はプロジェクト $p_i$ で使用されているコンポーネント $c_j$ の使用回数（書籍に対するユーザの評価）を表す。

協調フィルタリング推薦法では、現在開発中のプロジェクトでのコンポーネントの使用回数を基に、未使用のコンポーネントを次の3つの手順により推薦する。

手順1 現在開発中のプロジェクト $p_a$ と、他のプロジェクト $p_i$ との間の類似度をコンポーネントの使用回数から算出し、 $k$ 個の類似プロジェクトを求める。プロジェクト間の類似度計算には、Adjusted Cosine Similarity法<sup>4),9)</sup>を用いた。Adjusted Cosine Similarity法は、各プロジェクトにおけるコンポーネントの使用回数をベクトルの要素とし、2つのベクトルのなす角のコサインを類似度とする方法である。

表1 推薦に用いるデータ形式  
Table 1 Data format used for recommendation.

	$c_1$	$c_2$	...	$c_j$	...	$c_n$
$p_1$	$r_{11}$	$r_{12}$	...	$r_{1j}$	...	$r_{1n}$
$p_2$	$r_{21}$	$r_{22}$	...	$r_{2j}$	...	$r_{2n}$
...	...	...	...	...	...	...
$p_i$	$r_{i1}$	$r_{i2}$	...	$r_{ij}$	...	$r_{in}$
...	...	...	...	...	...	...
$p_m$	$r_{m1}$	$r_{m2}$	...	$r_{mj}$	...	$r_{mn}$

\*1 本論文は、筆者らの研究会原稿<sup>9)</sup>を基に実験の再検討と実施を行い、論文としてまとめたものである。

手順2 その $k$ 個の類似プロジェクトで使用されており、現在開発中のプロジェクトで未使用のコンポーネントの使用回数を見積もる。使用回数を見積りには、Weighted Sum法<sup>4),9)</sup>を用いた。Weighted Sum法は、類似度を重みとして、類似プロジェクトにおけるコンポーネントの使用回数の加重平均を見積り値とする方法である。

手順3 プロジェクト $p_a$ で使用されていないコンポーネントすべてに対して予測値を算出し、見積もった使用回数が多い順に次期開発で有用であろうコンポーネントとして推薦する。

## 3. 評価実験

### 3.1 実験の概要

本論文では、協調フィルタリング推薦法に対して2つの仮説、(1)汎用性の高いコンポーネントに対する推薦精度が従来手法（ランダム推薦法、平均値に基づく推薦法）より高い、(2)汎用性の低いコンポーネントに対する推薦精度が従来手法より高い、を実験的に検証する。仮説が正しいか否かを確かめるために、推薦されたコンポーネントが次期バージョンでどの程度用いられているかどうかを調べた。本論文において汎用性の低いコンポーネントは、全プロジェクト中、20%以下のプロジェクト（29件中5プロジェクト以下）でしか使用されていないコンポーネントとした。残りのコンポーネントは汎用性が高いと見なした。

類似性に基づくコンポーネント推薦方法<sup>9)</sup>の比較対象として、ランダム推薦法<sup>4)</sup>と平均値に基づく推薦法<sup>4),5)</sup>を用いた。ランダム推薦法は、各コンポーネントを無作為に推薦するアルゴリズム<sup>4)</sup>である。一方、平均値に基づく推薦は、他プロジェクトで平均的に高い頻度で使用されているコンポーネントを優先的に推薦するアルゴリズム<sup>4),5)</sup>である。

### 3.2 データセット

実験に使用したデータは、オープンソースソフトウェア開発環境を提供しているSourceForge.net<sup>\*2</sup>において、Java言語を用いて開発されているプロジェクトである。SourceForge.netにおいてプロジェクトの活発さを表す指標であるActivityの上位30件のプロジェクトのうち、複数のバージョンが公開されているプロジェクト29件を用いた。29件のプロジェクト、旧/新バージョンで用いられているコンポーネントのうち、以下の条件を満たすコンポーネントを除く2,558個のコンポーネントを用いた。

- 全プロジェクトにおいて使用頻度が著しく高く、開発者にとって既知であると考えられ

\*2 <http://sourceforge.net/>

るコンポーネント（全プロジェクトで使用されているコンポーネント）

- 全プロジェクトにおいて使用頻度が著しく低く、開発者にとって用いる機会が少ないと考えられるコンポーネント（1つのプロジェクトでしか使用されていないコンポーネント）

### 3.3 実験の手順

本実験において表1の  $p_i \in p_1, p_2, \dots, p_m$  は29件の新バージョンのプロジェクトを,  $c_j \in c_1, c_2, \dots, c_n$  は旧/新プロジェクトで使用されている2,558個のコンポーネントを示す。実験の手順を以下に示す。

手順1 推薦対象のプロジェクトとして,  $i$  番目のプロジェクト  $p_i$  を選択し,  $p_i$  を旧バージョンに置き換える。

手順2 旧バージョンのプロジェクト  $p_i$  の  $j$  番目のコンポーネント  $c_j$  の使用回数  $r_{i,j}$  を未使用と仮定し,  $c_j$  に対する予測値  $\hat{r}_{i,j}$  を他の新バージョンのプロジェクトを用いて算出する。

手順3 手順2をプロジェクト  $p_i$  のすべてのコンポーネント  $c_j$  で実行する。

手順4 手順1, 2, 3をすべてのプロジェクトにおいて実行する。

### 3.4 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値として, Yao が提案した NDPM (Normalized Distance-based Performance Measure)<sup>6)</sup> 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推薦などのアルゴリズムの精度評価に広く用いられている。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, 推薦の精度が高いことを示す。推薦の精度は, 旧バージョンにおけるコンポーネントの使用回数を基に推薦されたコンポーネントが, 新バージョンで実際に用いられているか否かで求める。推薦結果を次期開発に活かせるかという観点から評価するため, 本論文における理想の推薦  $O_a$  は, 新バージョンで実際に用いられていたコンポーネントの使用回数が大きい順に推薦されていた場合とした。

### 3.5 結果と考察

汎用性の高いコンポーネントにおいて各手法（ランダム推薦法, 平均値に基づく推薦法, 協調フィルタリング推薦法）による推薦を行った場合の NDPM を図1に示す。グラフの縦軸は NDPM を示す。協調フィルタリング推薦法は, NDPM の最大値では平均値に基づく推薦より大きな値であったものの, それ以外の値（最小値, 第1四分位点, 中央値, 第3四

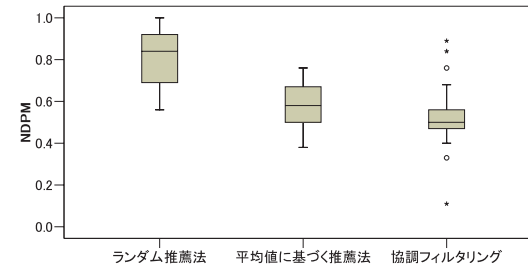


図1 実験結果（汎用性の高いコンポーネント）

Fig. 1 Experimental result for high-generality components.

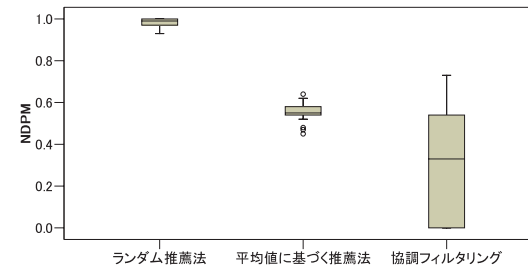


図2 実験結果（汎用性の低いコンポーネント）

Fig. 2 Experimental result for low-generality components.

分位点) では他の2つの手法より小さい値であった。ただし, 有意水準5%で Wilcoxon の符号付順位検定により平均値の差を検定した結果, 協調フィルタリング推薦法の NDPM の平均値は, ランダム推薦法と比べて有意に小さかったものの ( $p = 0.000$ ), 平均値に基づく推薦法と比べて有意に小さくならなかった ( $p = 0.058$ )。仮説(1)「汎用性の高いコンポーネントに対する推薦精度が従来手法より高い」は, 支持されなかった。

平均値に基づく推薦方法は, 使用回数が多い順にコンポーネントを推薦するため, 使用回数が多いコンポーネントが多く含まれる推薦では, 協調フィルタリング推薦法とほぼ同程度の性能が得られたと考える。

汎用性の低いコンポーネントにおいて推薦を行った場合の NDPM を図2に示す。汎用性の高いコンポーネントの場合と比べて従来手法との中央値の差が大きく, 平均値に基づく推薦の NDPM よりも中央値が0.22小さかった。また, 平均値の差の検定を行った結果, 協

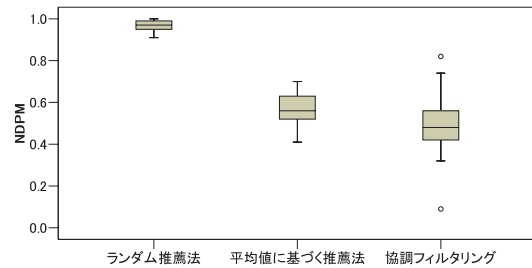


図3 実験結果(全コンポーネント)

Fig. 3 Experimental result for all components.

協調フィルタリング推薦法の NDPM の平均値は、他の 2 つの手法それぞれと比べて有意に小さかった ( $p = 0.000$ ,  $p = 0.000$ ). 仮説 (2) 「汎用性の低いコンポーネントに対する推薦精度が従来手法より高い」は、支持された。

一般的に、汎用性が低いコンポーネントは、有用となるドメインが限定されており、その推薦のためには各プロジェクトの個別性を考慮することが望ましい。協調フィルタリング推薦法は、類似するプロジェクトを求め、そのプロジェクトで使用されているコンポーネントが推薦されるため、プロジェクトの個別性が考慮され推薦精度が高かったと考えられる。協調フィルタリング推薦法は、汎用性の低いコンポーネントの推薦に対して特に有用であると期待できる。

最後に、コンポーネント全体において推薦を行った場合の NDPM を図 3 に示す。最大値以外では NDPM が小さく、従来手法と比べて高い精度が得られることが分かった。平均値の差の検定を行った結果、協調フィルタリング推薦法の NDPM の平均値は、他の 2 つの手法それぞれと比べて有意に小さかった ( $p = 0.000$ ,  $p = 0.003$ )。

#### 4. 関連研究

協調フィルタリングを用いたコンポーネント推薦の手法がいくつか提案されている<sup>3),7),8)</sup>。たとえば、McCarey ら<sup>3)</sup> は、メソッドの使用順序を基に、次に用いるであろうメソッドを推薦する手法を提案している。その効果を確認するために、Java の描画ライブラリ Swing

と AWT のメソッド (2090 メソッド) を対象とした評価実験を行っている。

一方で、有用なコンポーネントを発見する手段として、SPARS-J<sup>2)</sup>、Google Code Search<sup>\*1</sup>、Koders<sup>\*2</sup>、Jarhoo<sup>\*3</sup>といったコンポーネント検索エンジンが多数提案されている。たとえば、SPARS-J は Java ソースコード検索システムであり、ソースコードを静的解析して抽出した索引語を用いてキーワード検索を行うことができる。

上記のように、コンポーネント推薦やコンポーネント検索エンジンは今までにいくつか存在するが、本論文は、協調フィルタリングを用いたコンポーネント推薦の評価、特に汎用性の高い/低いコンポーネントに対する推薦精度を実験的に評価した点が従来と異なる。

#### 5. おわりに

本論文では、汎用性の高い/低いコンポーネントに対して協調フィルタリングを用いたコンポーネント推薦方法<sup>9)</sup>が効果があるかを明らかにするために、29 件のプロジェクト (使用されているコンポーネントの総数は 2,558 個) を用いて評価実験を行った。その結果、協調フィルタリングを用いたコンポーネント推薦は汎用性の低いコンポーネントに対して特に効果があることが分かった。

謝辞 本研究の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。また、特別研究員奨励費 (課題番号: 20009220) の研究助成を受けて行われた。

#### 参考文献

- 1) Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T.: Evaluating collaborative filtering recommender systems, *ACM Trans. Inf. Syst.*, Vol.22, No.1, pp.5-53 (2004).
- 2) Inoue, K., Yokomori, R., Yamamoto, T., Matsushita, M. and Kusumoto, S.: Ranking Significance of Software Components Based on Use Relations, *IEEE Trans. Softw. Eng.*, Vol.31, No.3, pp.213-225 (2005).
- 3) McCarey, F., Cinnéide, M.O. and Kushmerick, N.: Knowledge reuse for software reuse, *Web Intelligence and Agent Systems*, Vol.6, No.1, pp.59-81 (2008).
- 4) Ohsugi, N.: A Framework for Software Function Recommendation based on Collaborative Filtering, Ph.D. Thesis, Nara Institute of Science and Technology (2004).
- 5) Shardanand, U. and Maes, P.: Social information filtering: algorithms for automating "word of mouth", *Proc. CHI Conf. on Human Factors in Computing Systems*, pp.210-217 (1995).

\*1 <http://www.google.com/codesearch>\*2 <http://www.koders.com/>\*3 <http://www.jarhoo.com/>

- 6) Yao, Y.: Measuring retrieval effectiveness based on user preference of documents, *Journal of the American Society for Information Science*, Vol.46, No.2, pp.133-145 (1995).
- 7) Ye, Y. and Fischer, G.: Supporting reuse by delivering task-relevant and personalized information, *Proc. Int'l Conf. on Software Engineering*, pp.513-523 (2002).
- 8) 市井 誠, 山本哲男, 横森励士, 井上克郎: ソフトウェア部品推薦のための協調フィルタリング手法の提案と実現, 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス, Vol.104, No.243, pp.7-12 (2004).
- 9) 亀井靖高, 角田雅照, 柿元 健, 大杉直樹, 門田暁人, 松本健一: 進行中のプロジェクトに有用なソフトウェアコンポーネントの推薦方法, 電子情報通信学会技術報告. SS, ソフトウェアサイエンス研究会, Vol.106, No.16, pp.25-30 (2006).

(平成 20 年 8 月 21 日受付)

(平成 20 年 12 月 5 日採録)



亀井 靖高 (学生会員)

平成 17 年関西大学総合情報学部卒業. 平成 19 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, 同大学博士後期課程在籍. 平成 20 年日本学術振興会特別研究員 (DC2) 採用, 現在に至る. 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の研究に従事.



角田 雅照 (正会員)

平成 9 年和歌山大学経済学部卒業. 平成 19 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. 同年同大学特任助教. 博士 (工学). ソフトウェアメトリクス, コピキタスコンピューティングの研究に従事.



柿元 健 (正会員)

平成 15 年神戸市立工業高等専門学校専攻科電気電子工学専攻修了. 平成 20 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. 同年大阪大学情報学部特任助教. 博士 (工学). ソフトウェア信頼性/開発工数予測等の研究に従事.



大杉 直樹 (正会員)

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 16 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. 同年同大学研究員. 平成 19 年株式会社 NTT データ入社, 技術開発本部勤務. 博士 (工学). エンピリカルソフトウェア工学, コスト見積手法, 要件定義書レビュー方法の研究に従事.



門田 暁人 (正会員)

平成 6 年名古屋大学工学部卒業. 平成 10 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. 同年同大学助手. 平成 16 年同大学助教授. 平成 19 年同大学准教授. 平成 15~16 年 Auckland 大学客員研究員. 博士 (工学). 定量的ソフトウェア開発支援, ソフトウェアプロテクションの研究に従事.



松本 健一 (正会員)

昭和 60 年大阪大学基礎工学部卒業. 平成元年同大学大学院博士課程中退. 同年同大学助手. 平成 5 年奈良先端科学技術大学院大学情報科学研究科助教授. 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/利用支援の研究に従事.