

Javawock: A Java Class Recommender System Based on Collaborative Filtering

Masateru Tsunoda

Takeshi Kakimoto

Naoki Ohsugi

Akito Monden

Ken-ichi Matsumoto

In Proceedings of 17th International Conference on Software Engineering and Knowledge Engineering,
pp.491-497, July 2005.

JavaWock: 協調フィルタリングに 基づくJavaクラス推薦システム

Masateru Tsunoda

Takeshi Kakimoto

Naoki Ohsugi

Akito Monden

Ken-ichi Matsumoto

In Proceedings of 17th International Conference on Software Engineering and Knowledge Engineering,
pp.491-497, July 2005.

便利なライブラリを見つけにくい

- 開発者は高品質なソフトウェアを簡単に作るために、ライブラリを利用する
- しかし、提供されているライブラリはあまりにも多い
- 開発者にとって便利なライブラリを見つけにくい



ライブラリの推薦システムが求められる

関連研究：SPARS-J

- 井上ほか
- 過去に開発された類似のソフトウェアの再利用を目的とした検索システム（推薦システムではない）
- キーワードで検索する → 開発者が適切なキーワードを考える必要がある

関連研究：CodeBroker

- YeとFischer
- 現在のタスクと関連のあるJavaライブラリを推薦するシステム
- ソースコード内のコメントからキーワードを自動で抽出
- ただし、適合率が低かった

関連研究：RASCAL

- McCareyほか
- 協調フィルタリングに基づいたJavaのメソッドの推薦システム
- ライブラリを推薦するものではない
- アイテムベースの協調フィルタリングを行っていない

協調フィルタリング

- 多くのユーザの嗜好情報を蓄積し、あるユーザと嗜好の類似した他のユーザの情報を用いて自動的に推論を行う方法論
- 例えばユーザAがXを好み、ユーザBがXとYを好むとき、ユーザAもYを好むのではないか、という推論をコンピュータによって自動的に行う
- Collaborative Filtering (CF)

提案手法

- 協調フィルタリングに基づくJavaクラス推薦システム

アーキテクチャ

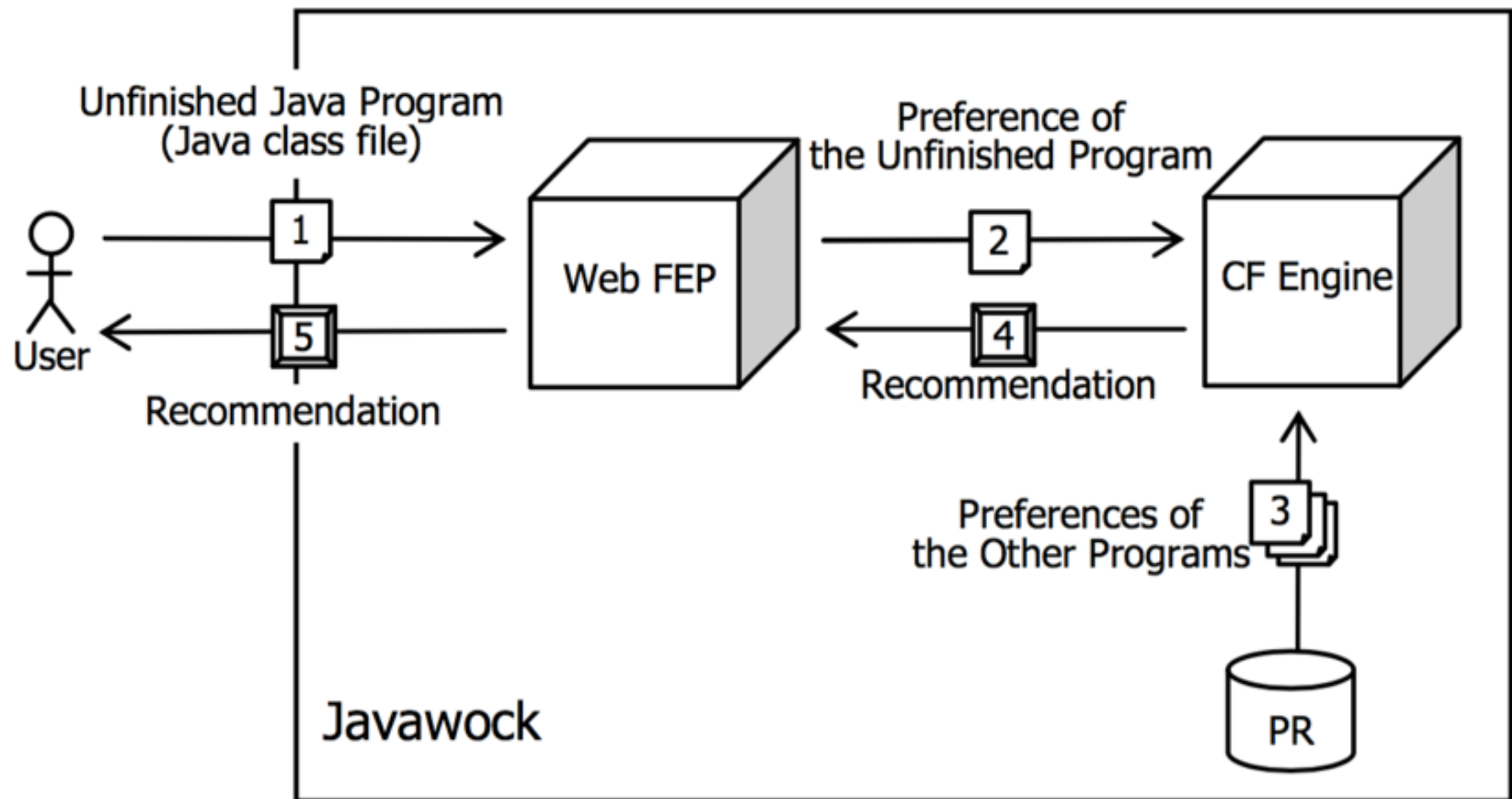


Fig. 2. The architecture of Javawock

CFの手順1

	l_1	l_2	$\dots$	l_j	$\dots$	l_b	$\dots$	l_n
p_1	$u_{1,1}$	$u_{1,2}$	$\dots$	$u_{1,j}$	$\dots$	$u_{1,b}$	$\dots$	$u_{1,n}$
p_2	$u_{2,1}$	$u_{2,2}$	$\dots$	$u_{2,j}$	$\dots$	$u_{2,b}$	$\dots$	$u_{2,n}$
$\dots$	$\dots$	$\dots$		$\dots$		$\dots$		$\dots$
p_i	$u_{i,1}$	$u_{i,2}$	$\dots$	$u_{i,j}$	$\dots$	$u_{i,b}$	$\dots$	$u_{i,n}$
$\dots$	$\dots$	$\dots$		$\dots$		$\dots$		$\dots$
p_a	$u_{a,1}$	$u_{a,2}$	$\dots$	$u_{a,j}$	$\dots$	$u_{a,b}$	$\dots$	$u_{a,n}$
$\dots$	$\dots$	$\dots$		$\dots$		$\dots$		$\dots$
p_m	$u_{m,1}$	$u_{m,2}$	$\dots$	$u_{m,j}$	$\dots$	$u_{m,b}$	$\dots$	$u_{m,n}$

Fig. 3. $m \times n$ table used for recommendation

CFの手順2

$$\text{sim}(p_a, p_i) = \frac{\sum u_{a,j} \times u_{i,j}}{\sqrt{\sum (u_{a,j})^2} \sqrt{\sum (u_{i,j})^2}} \quad (1)$$

$$\text{sim}(l_b, l_j) = \frac{\sum u_{i,b} \times u_{i,j}}{\sqrt{\sum (u_{i,b})^2} \sqrt{\sum (u_{i,j})^2}} \quad (2)$$

コサイン類似度(範囲0~1)を求める

CFの手順3

$$R_{a,b} = \frac{\sum_{K \in k\text{-nearestPrograms}} \text{sim}(p_a, p_K) \times u_{K,b}}{\sum_{K \in k\text{-nearestPrograms}} \text{sim}(p_a, p_K)} \quad (3)$$

$$R_{a,b} = \frac{\sum_{K \in k\text{-nearestLibraryClassFiles}} \text{sim}(l_b, l_K) \times u_{a,K}}{\sum_{K \in k\text{-nearestLibraryClassFiles}} \text{sim}(l_b, l_K)} \quad (4)$$

推薦スコアを求める

評価実験

- J2SE Software Development Kitを対象とする
- よく使われるクラスを371個とそのサブセットのライブラリを331個
選定

実験結果

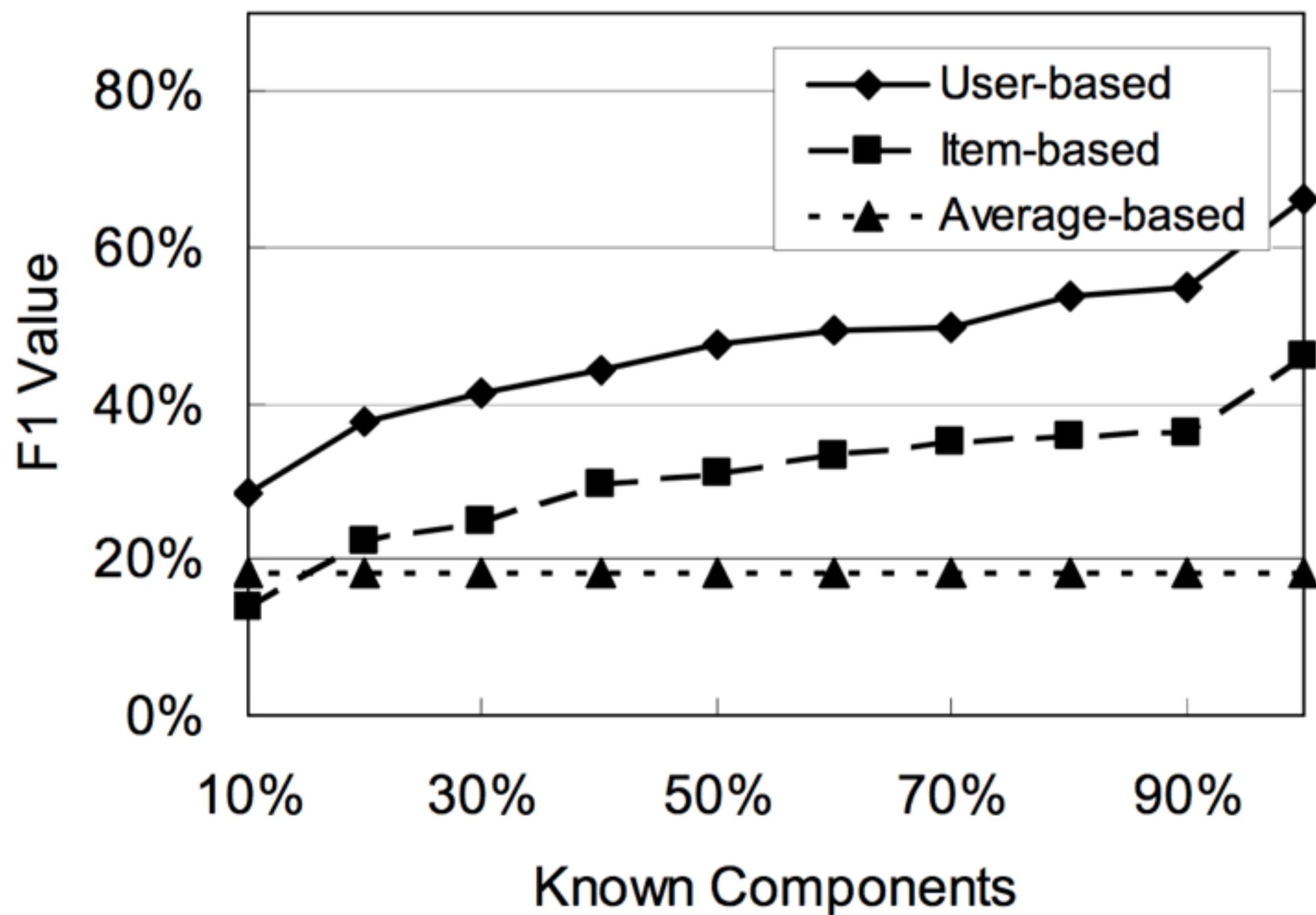


Fig. 6. F1 value of each method

実験結果

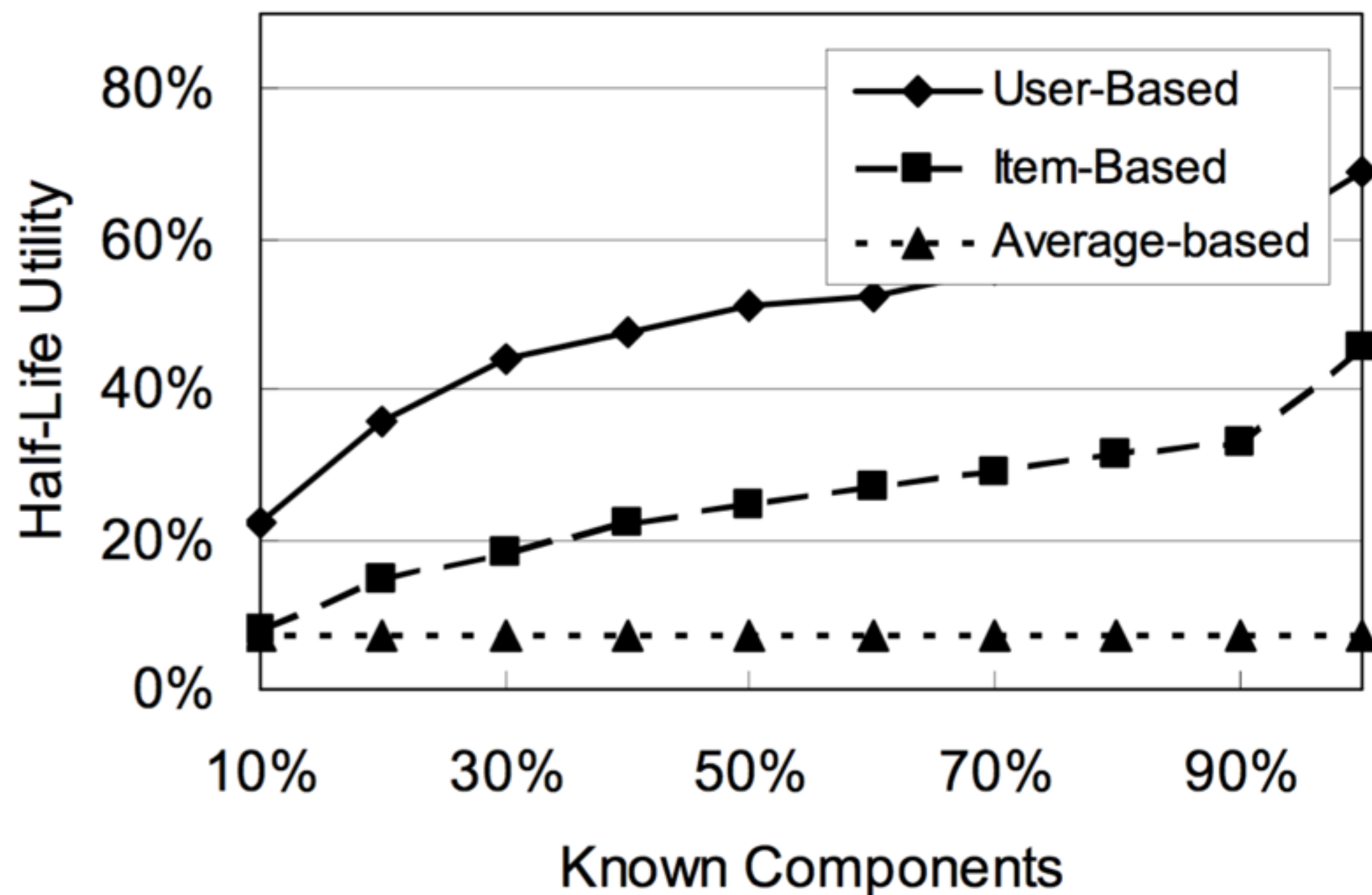


Fig. 7. Half-life utility of each method

実験結果

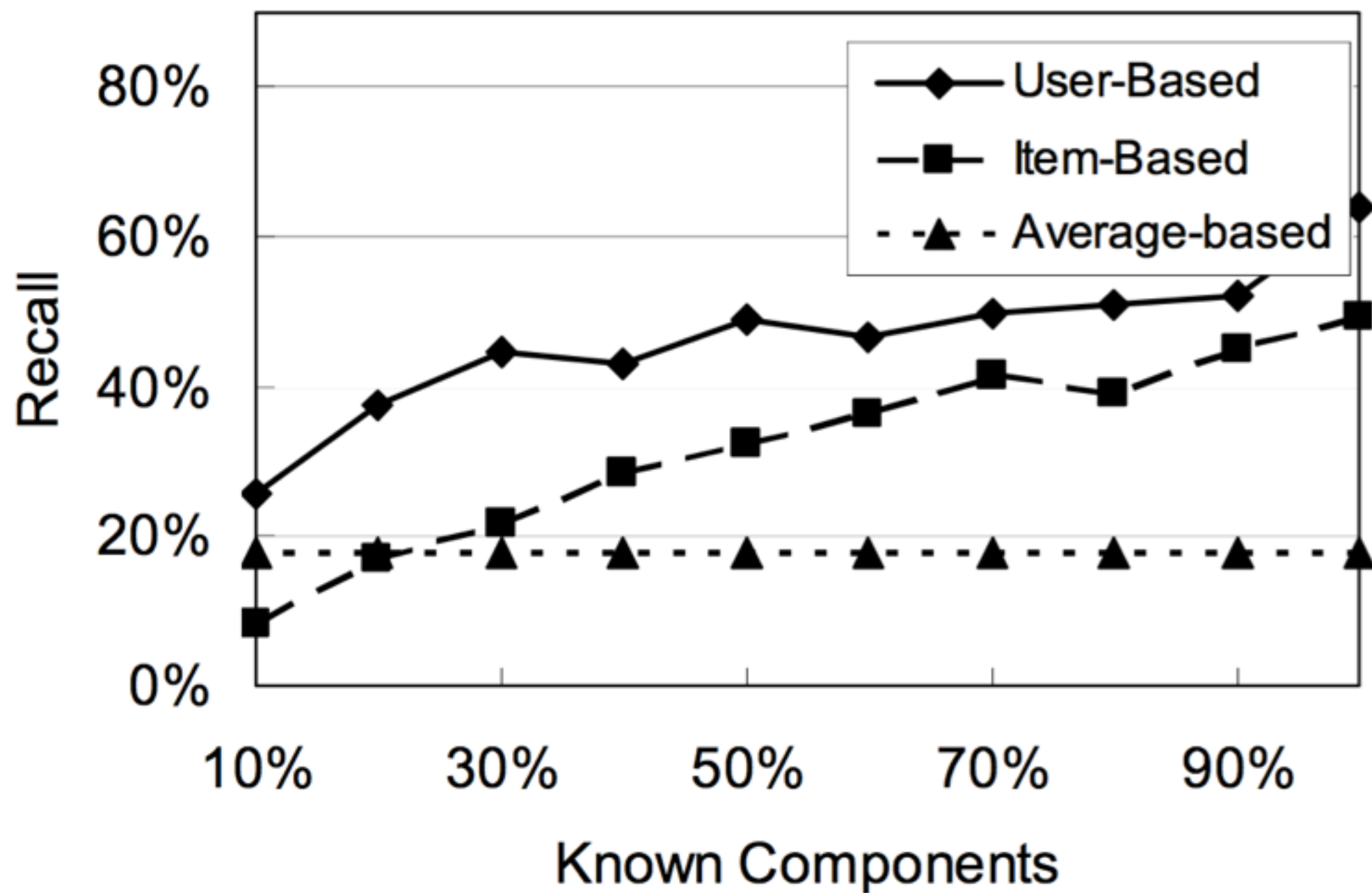


Fig. 8. Precision of each method

実験結果

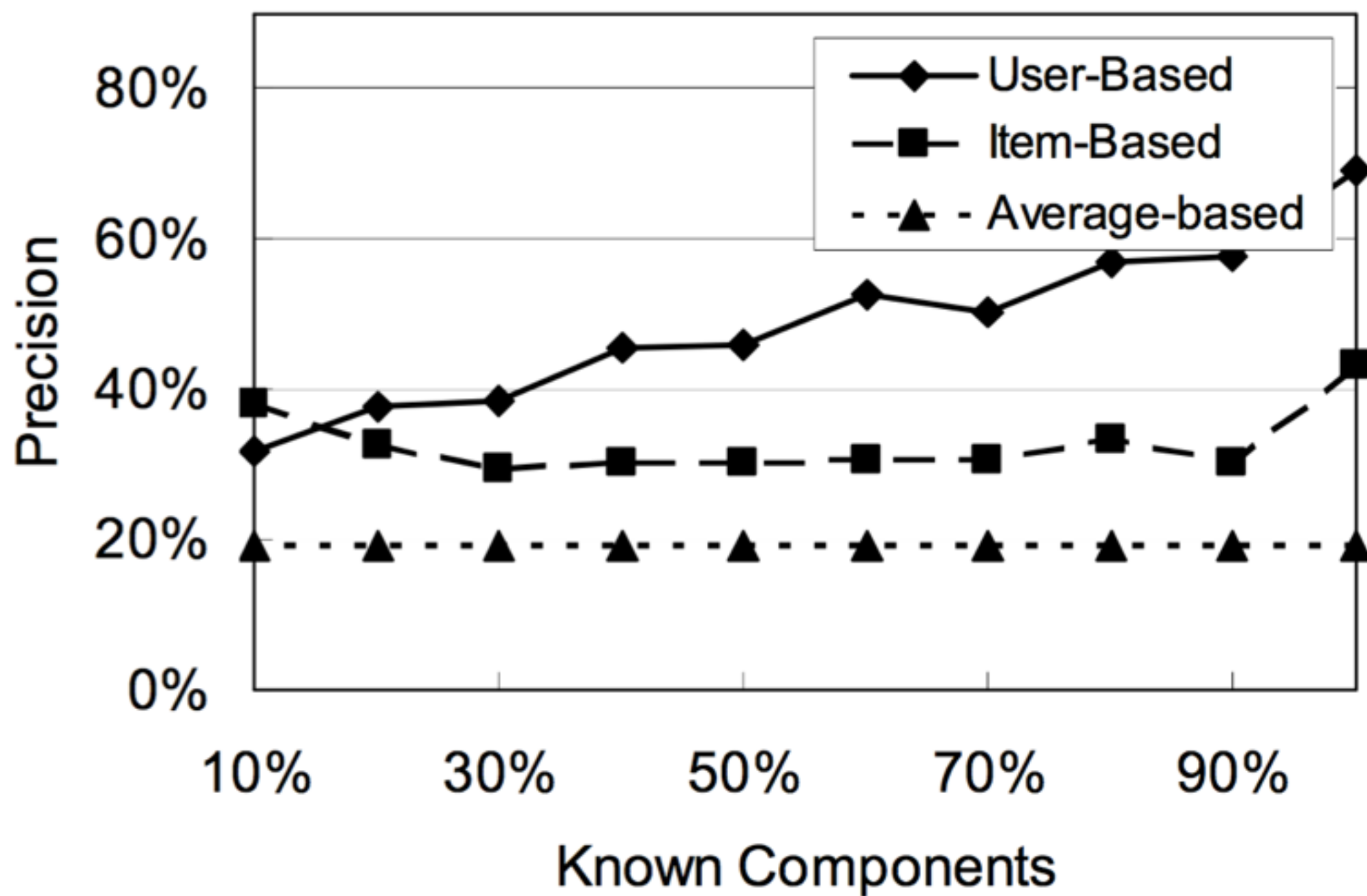


Fig. 9. Recall of each method

実験結果

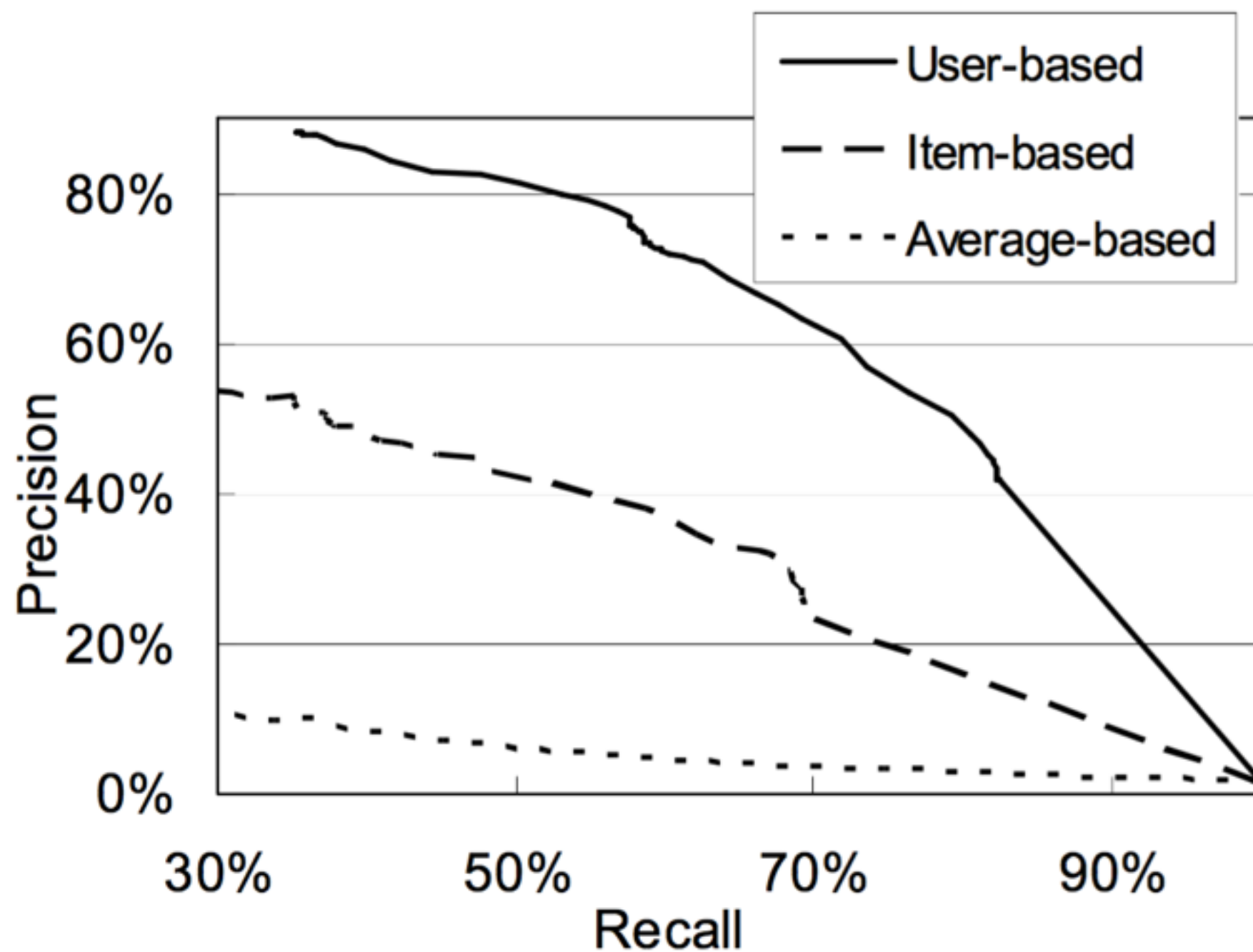


Fig. 10. The relation between recall and precision

まとめ

- 協調フィルタリングに基づくJavaクラス推薦システムを提案した
- 協調フィルタリングを行う方がよりよい結果を得られた