修士学位論文

題　　目

目的関数ベースのRough Membership C-Means クラスタリングに基づく協調フィルタリング

Collaborative Filtering Based on Objective Function-Based Rough Membership C-Means Clustering

主査	本多　克宏	教授
副査	宇野　裕之	教授
副査	能島　裕介	教授
副査	生方　誠希	准教授

令和　4　年　（　2022年　）度修了

(No. 2210104023)　　KIM　HAERANG

大阪府立大学大学院工学研究科

電気・情報系専攻　知能情報工学分野

目的関数ベースの Rough Membership C-Means クラスタリングに基づく協調フィルタリング

Collaborative Filtering Based on Objective Function-Based Rough Membership C-Means Clustering

大阪府立大学大学院　工学研究科　電気・情報系専攻　知能情報工学分野
人間情報システム研究グループ　2210104023　KIM HAERANG

Abstract: Collaborative filtering is a technique to recommend items by user's preference. Since users' preference is based on human sensitivity, methods that can deal with uncertainty like Rough clustering can be effective. However, since most of the rough clustering methods are heuristic, there is a problem that it is hard to discuss the cluster validity. Therefore, in this study, we introduced two types of Objective function-based Rough Membership C-Means clustering which is possible to discuss cluster validity to Collaborative Filtering. Furthermore, we verified the recommendation performance by adapting our proposed methods, namely, user-based RMCM2-CF and item-based RMCM2-CF, to real-world datasets and comparing other collaborative filtering methods.

1　はじめに

協調フィルタリング (Collaborative Filtering, CF) は対象のユーザに対して，他のユーザの嗜好情報に基づいて好ましいアイテムを推薦する手法であり，Amazon などの電子商取引サイトや YouTube などの動画配信サイト等のコンテンツ推薦システムを実現している．協調フィルタリングには，メモリベース協調フィルタリングとモデルベース協調フィルタリングがある．さらにメモリベース協調フィルタリングにはユーザ間の類似度に基づいて推薦を行うユーザベース協調フィルタリング (user-based CF)[1] とアイテム間の類似度に基づいて推薦を行うアイテムベース協調フィルタリング (item-based CF)[2] に分けることができる．モデルベース協調フィルタリグは嗜好情報からモデルを作りそのモデルに基づいて推薦を行う手法であり，Matrix Factorization[3] やクラスタリングベース協調フィルタリングがモデルベース協調フィルタリングに該当する．そのなかでもクラスタリングベース協調フィルタリングは実装が容易であり，効率的な計算ができるおよびクラスター中心のみで推薦器が作れるのでメモリー削減が期待されるというメリットを持つ．

クラスタリングとは，データ内の類似した個体をクラスターとして抽出し，自動的にグループ化する手法である．代表的なクラスタリング手法として，Hard C-Means (HCM; k-means) 法 [4] がある．HCM 法では，各対象は唯一のクラスターに帰属される排他的な分割が行われる．しかし，協調フィルタリングタスクで用いるユーザの嗜好情報は人間の主観的な評価に基づいており，曖昧さや不確実性を含んでいる．したがって，曖昧さや不確実性を取り扱うことができるクラスタリング手法が協調フィルタリングタスクにおいて有効であると考えられる．

ラフ集合理論 [5] に基づいて不確実性を取り扱うことのできるクラスタリング手法として，ラフクラスタリング [6] がある．ラフクラスタリングは，対象のクラスターに対する帰属の確実性･可能性･不確実性を考慮することにより，一つの対象が複数のクラスターへ帰属することを表現できる．ラフクラスタリングのアルゴリズムとしては，Generalized Rough C-Means 法 [7]，Rough Set C-Means 法 [8] など様々な方法が提案されている．その中でも，Rough Membership C-Means (RMCM) 法 [8] は，より詳細な近傍情報を利用することができ，クラスター中心決定の際のウェイト変数を設定する必要がないという利点を持つ．

しかし，これらのラフクラスタリングはヒューリスティックな手法なため，クラスタリングの妥当性検証が困難となっている．本研究では，目的関数を導入することでクラスタリングの妥当性検証および理論的な展開が可能となる目的関数ベースの RMCM 法 (RMCM2)[9] に基づく協調フィルタリング手法としてユーザベースおよびアイテムベースの 2 つのアプローチを考える．まず，類似したユーザのグループを抽出する RMCM2 法に基づく協調フィルタリング (user-based RMCM2-CF) を提案する．さらに，類似したアイテムのグループを抽出するアイテムベースの RMCM2 法に基づく協調フィルタリング (item-based RMCM2-CF) を提案し，各手法を実データに適用することで推薦性能を検証する．また，従来の HCM 法に基づく協調フィルタリング (HCM-CF)，Rough Membership C-Means 法に基

づく協調フィルタリング (RMCM-CF) およびメモリベース協調フィルタリング手法 (user-based CF, item-based CF) との比較を通じて提案法の有効性を検証する．

本論文は以下の 7 章から構成されている．第 2 章では，準備として各クラスタリング手法 (HCM 法, RMCM 法，RMCM2 法) について概説し，第 3 章では，各協調フィルタリング手法 (HCM-CF，RMCM-CF，user-based CF，item-based CF) および提案法である RMCM2-CF を説明する．第 4 章では数値実験の設定を，第 5 章では結果を示す．最後に，第 6 章で考察を，第 7 章で本論文のまとめを述べる．

2 準備

2.1 HCM 法

代表的な非階層的クラスタリング手法である HCM 法は，クラスター中心の算出と対象のクラスター割り当てを繰り返すことでクラスターを抽出する．HCM 法の目的関数を以下に示す．ただし，u_ci は対象 i のクラスター c に対するメンバシップ，d_{ci} は $||\boldsymbol{x}_i - \boldsymbol{b}_c||$ である．

$$
\begin{aligned}
J_{\mathrm{HCM}} &= \sum_{c=1}^{C}\sum_{i=1}^{n} u_{ci} d_{ci}^2. \\
\text{s.t.}\quad & u_{ci} \in \{0,1\}, \quad \forall c, i, . \\
& \sum_{c=1}^{C} u_{ci} = 1, \quad \forall i.
\end{aligned}
\tag{1}
$$

HCM 法では，各対象は唯一のクラスターに帰属されるため，複数のクラスターへの帰属を表すことができない．よって，HCM 法はデータに内在する曖昧性・不確実性を取り扱うことが不可能となる．

HCM 法のアルゴリズムを以下に示す．

Step 1 クラスター数 C を設定する．

Step 2 C 個の初期クラスター中心 $\boldsymbol{b}_c$ を対象空間 $U = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_n\}$ の中から非復元抽出により決定する．

Step 3 対象 i のクラスター c に対するメンバシップ u_{ci} を最近隣割り当てによって求める．

$$
u_{ci} = \begin{cases} 1 & \left(c = \underset{1 \le l \le C}{\arg\min} ||\boldsymbol{x}_i - \boldsymbol{b}_l|| \right), \\ 0 & (\text{otherwise}) . \end{cases}
\tag{2}
$$

Step 4 クラスター中心 $\boldsymbol{b}_c$ を計算する．

$$
\boldsymbol{b}_c = \frac{\sum_{i=1}^{n} u_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} u_{ci}}.
\tag{3}
$$

Step 5 u_{ci} に変化がなくなるまで **Step 3-4** を繰り返す．

2.2 RMCM 法

RMCM 法は，HCM 法をラフ集合理論によって拡張した手法の一つであり，対象の全体集合 U について，二項関係 $\mathcal{R} \subseteq U \times U$ によって粒状化された空間においてクラスタリングを実行する．RMCM 法では，最近隣割り当てによる暫定クラスターを算出した上で，対象の $\mathcal{R}$ による近傍内でのクラスター比率を表すラフメンバシップをクラスターメンバシップとして利用することで帰属の不確実性を取り扱う．これより，RMCM 法はクラスター中心の決定のためのウェイト変数を設定する必要がなく，他のラフクラスタリング手法より詳細な近傍情報を用いてクラスタリングを行うことができる．

RMCM 法のアルゴリズムを以下に示す．

Step 1 クラスター数 C および二項関係 $\mathcal{R}$ を設定する．

$$
R_{it} = \begin{cases} 1 & (||\boldsymbol{x}_t - \boldsymbol{x}_i|| \le \delta) , \\ 0 & (\text{otherwise}) . \end{cases}
\tag{4}
$$

ここで，近傍半径 δ は対象間距離分布の τ-パーセンタイルによって決定する．

Step 2 初期クラスター中心 $\boldsymbol{b}_c$ を決定する．

Step 3 対象 i のクラスター c に対するメンバシップ u_{ci} を (2) 式の最近隣割り当てによって求める．

Step 4 ラフメンバシップ $\mu_{ci}^{\mathcal{R}}$ を計算する．

$$
\mu_{ci}^{\mathcal{R}} = \frac{\sum_{t=1}^{n} R_{it} u_{ct}}{\sum_{t=1}^{n} R_{it}}.
\tag{5}
$$

Step 5 クラスター中心 $\boldsymbol{b}_c$ を計算する．

$$
\boldsymbol{b}_c = \frac{\sum_{t=1}^{n} \mu_{ci}^{\mathcal{R}} \boldsymbol{x}_i}{\sum_{i=1}^{n} \mu_{ci}^{\mathcal{R}}}.
\tag{6}
$$

Step 6 u_{ci} に変化がなくなるまで **Step 3-5** を繰り返す．

2.3 RMCM2 法

RMCM2 法はヒューリスティックなアルゴリズムである RMCM 法に目的関数を設定することで，クラスタリングの妥当性検証やさらなる理論的展開を可能とすることを目的とした手法である．RMCM2 法の目的関数は RMCM 法のクラスター中心の更新式（(6) 式）が導出されるように設計されている．RMCM2 法の目的関数を以下に示す．

$$
\begin{aligned}
J_{\mathrm{RMCM2}} &= \sum_{c=1}^{C}\sum_{i=1}^{n} \mu_{ci}^{\mathcal{R}} d_{ci}^2. \\
\text{s.t.}\quad & u_{ci} \in \{0,1\}, \quad \forall c,i,. \\
& \sum_{c=1}^{C} u_{ci} = 1, \quad \forall i.
\end{aligned}
\tag{7}
$$

この目的関数を最小化することが RMCM2 法の最適化問題となる．この問題をラグランジュ未定乗数法を用いて解くことにより，クラスター中心の更新式（(6) 式）とメンバシップの更新式（(8) 式）が求められる．

$$
u_{ci} = \begin{cases} 1 & \left(c = \underset{1\le l\le C}{\arg\min} \sum_{k=1}^{n} \frac{R_{ki}}{\sum_{t=1}^{n} R_{kt}} d_{lk}^2 \right), \\ 0 & (\text{otherwise}). \end{cases}
\tag{8}
$$

RMCM 法では，最近隣割り当てによる暫定クラスターを算出するが，RMCM2 法では目的関数から導出されるメンバシップの更新式により，近傍情報を用いて近傍全体を含めてクラスター中心に近いかを判定することで暫定クラスターを算出する．また，RMCM2 法は原理上，どの個体も含まない空クラスターが発生しやすい．

RMCM2 法のアルゴリズムを以下に示す．

Step 1 クラスター数 C および二項関係 $\mathcal{R}$ を (4) 式で設定する．

Step 2 初期クラスター中心 $\boldsymbol{b}_c$ を決定する．

Step 3 対象 i のクラスター c に対するメンバシップ u_{ci} を (8) 式より求める．

Step 4 ラフメンバシップ $\mu_{ci}^{\mathcal{R}}$ を (5) 式で計算する．

Step 5 クラスター中心 $\boldsymbol{b}_c$ を (6) 式で計算する．

Step 6 u_{ci} に変化がなくなるまで **Step 3-5** を繰り返す．

3 協調フィルタリング

3.1 メモリベース協調フィルタリング

メモリベース協調フィルタリングは，ユーザ間またはアイテム間の類似度を基にアイテムを推薦する手法である．各メモリベース協調フィルタリング手法について以下で説明する．

3.1.1 user-based CF

user-based CF では，ユーザ間の類似度に基づいて評価値を計算し，嗜好の類似したユーザが高く評価しているアイテムを高く推薦し，類似しないユーザが高く評価しているアイテムを低く推薦する．user-based CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{uv}\}$ を基に，ユーザ同士 (u と v) の類似度 w_{uv} を計算する．用いられる類似度はピアソンの積率相関係数，コサイン類似度，Jaccard 係数などがある．以下にピアソンの積率相関係数の計算式を示す．ただし，$\bar{r}_u$ はユーザ u の平均評価値，$\bar{r}_v$ はユーザ v の平均評価値を意味する．

$$
w_{uv} = \frac{\sum_k (r_{uk} - \bar{r}_u)(r_{vk} - \bar{r}_v)}{\sqrt{\sum_k (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_k (r_{vk} - \bar{r}_v)^2}}
\tag{9}
$$

Step 2 以下の式に従ってユーザ u に対するアイテム i の推薦度を計算する．また，U はアイテム i と j を両方評価したユーザの集合となる．

$$
\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v\in U} w_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v\in U} |w_{uv}|}
\tag{10}
$$

3.1.2 item-based CF

item-based CF ではアイテム間の類似度を基づいて評価値を計算し，ユーザが高く評価しているアイテムと似ているアイテムを高く推薦する．item-based CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{ij}\}$ を基に，アイテム同士の類似度を計算する．以下にピアソンの積率相関係数の計算式を示

す．ただし，$\bar{r}_i$ はアイテム i の平均評価値，$\bar{r}_j$ はアイテム j の平均評価値を意味する．

$$w_{i,j} = \frac{\sum_{u \in U}(r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{ui} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{uj} - \bar{r}_j)^2}} \tag{11}$$

Step 2 以下の式に従ってユーザ u に対するアイテム j の推薦度を計算する．ただし，N はユーザ u が評価したアイテムの集合となる．

$$\hat{r}_{ui} = \frac{\sum_{n \in N} r_{un} w_{in}}{\sum_{n \in N} |w_{in}|} \tag{12}$$

実際の推薦システムにおいては，user-based CF よりも item-based CF の方が使用される場合が多い．user-based CF と比べての item-based CF のメリットを以下に示す．

- user-based CF では新しいユーザが入った時，そのユーザに対する情報不足よりアイテムの推薦が難しくなる cold-start 問題が生じる．それに対して item-based CF ではアイテム間の類似度を用いて推薦を行うため cold-start 問題に対処することができる．
- item-based CF ではなぜそのアイテムが推薦されたかをユーザに説明できる．
- 一般的に item-based CF は user-based CF より高いスケーラビリティおよび推薦性能を持つ．

3.2 HCM-CF

3.2.1 user-based HCM-CF

user-based HCM-CF は，HCM 法によって嗜好の類似したユーザのクラスターを抽出し，クラスター内で嗜好度の高いコンテンツを推薦する手法である．

user-based HCM-CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{ij}\}$ に対して HCM 法を適用し，ユーザクラスター構造 (メンバシップ u_{ci} とクラスター中心 b_{cj}) を求める．ここで，n はユーザ数，m はアイテム数である．

Step 2 ユーザ i に対するアイテム j の推薦度 $\hat{r}_{ij}$ を計算する．

$$\hat{r}_{ij} = \sum_{c=1}^{C} u_{ci} b_{cj}. \tag{13}$$

Step 3 閾値 $\eta \in [\min\{\hat{r}_{ij}\}, \max\{\hat{r}_{ij}\}]$ 以上の推薦度を持つアイテムを推薦する．

$$\tilde{r}_{ij} = \begin{cases} 1 & (\hat{r}_{ij} \geq \eta), \\ 0 & (\text{otherwise}). \end{cases} \tag{14}$$

3.2.2 item-based HCM-CF

item-based HCM-CF では，ユーザではなくアイテムに注目し，HCM 法によって類似したアイテムのクラスターを抽出することで，クラスター内のアイテムを高く評価（または所有）しているユーザへの推薦度を高くする．アイテム空間で計算を行うことで，ユーザ数よりアイテム数が少ない場合においてメモリ・時間削減が期待できる．

item-based HCM-CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{ij}\}$ の転置行列に対して HCM 法を適用し，アイテムクラスター構造 (メンバシップ u_{cj} とクラスター中心 b_{ci}) を求める．

Step 2 ユーザ i に対するアイテム j の推薦度 $\hat{r}_{ij}$ を計算する．

$$\hat{r}_{ij} = \sum_{c=1}^{C} u_{cj} b_{ci}. \tag{15}$$

Step 3 閾値 $\eta \in [\min\{\hat{r}_{ij}\}, \max\{\hat{r}_{ij}\}]$ 以上の推薦度を持つアイテムを式 (14) により推薦する．

3.3 RMCM-CF

RMCM-CF は，評価値行列 X に RMCM 法を適用することで人間の感性に起因する不確実性を考慮することができる協調フィルタリング手法である．

3.3.1 user-based RMCM-CF

user-based RMCM-CF は，評価値行列 X に RMCM 法を適用することでユーザのクラスターを抽出し，

クラスター内で嗜好度の高いコンテンツを推薦する手法である．

user-based RMCM-CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{ij}\}$ に対して RMCM 法を適用し，ユーザクラスター構造 (ラフメンバシップ $\mu_{ci}^{\mathcal{R}}$ とクラスター中心 b_{cj}) を求める．

Step 2 ユーザ i に対するアイテム j の推薦度 $\hat{r}_{ij}$ を計算する．

$$\hat{r}_{ij} = \sum_{c=1}^{C} \mu_{ci}^{\mathcal{R}} b_{cj}. \tag{16}$$

Step 3 閾値 $\eta \in [\min\{\hat{r}_{ij}\}, \max\{\hat{r}_{ij}\}]$ 以上の推薦度を持つアイテムを (14) 式のように推薦する．

3.3.2 item-based RMCM-CF[10]

item-based RMCM-CF は，評価値行列の転置行列 X^T に RMCM 法を適用することで類似したアイテムのクラスターを抽出し，クラスター内のアイテムを高く評価（または所有）しているユーザへの推薦度を高くする手法である．item-based RMCM-CF の手順を以下に示す．

Step 1 $n \times m$ の評価値行列 $X = \{r_{ij}\}$ の転置行列に対して RMCM 法を適用し，アイテムクラスター構造 (ラフメンバシップ $\mu_{cj}^{\mathcal{R}}$ とクラスター中心 $\boldsymbol{b}_{ci}$) を求める．

Step 2 ユーザ i に対するアイテム j の推薦度 $\hat{r}_{ij}$ を計算する．

$$\hat{r}_{ij} = \sum_{c=1}^{C} \mu_{cj}^{\mathcal{R}} b_{ci}. \tag{17}$$

Step 3 閾値 $\eta \in [\min\{\hat{r}_{ij}\}, \max\{\hat{r}_{ij}\}]$ 以上の推薦度を持つアイテムを (14) 式のように推薦する．

3.4 提案法：RMCM2-CF

RMCM2-CF は，RMCM2 法を用いることで不確実性を考慮し，目的関数を導入することで今後さらなる理論的展開を可能とするベースになる手法である．

また，予備実験により，RMCM2 法は空クラスターが発生しやすいという問題 (6. 考察) を確認している．空クラスターが発生すると，指定した数のクラスターが抽出できないことに加えて，クラスター中心計算の際のゼロ除算によりアルゴリズムが停止する問題がある．したがって，本研究の実験においては，空クラスターが発生した場合はその空クラスターを除外することで空クラスターが推薦性能に及ぼす影響を軽減している．

3.4.1 user-based RMCM2-CF

user-based RMCM2-CF では，評価値行列 X に RMCM2 法を適用することで嗜好の類似したユーザのクラスターを抽出し，クラスター内で嗜好度の高いコンテンツを推薦する．

user-based RMCM2-CF の手順は user-based RMCM-CF の手順と同様であり，評価値行列 X に RMCM2 法を適用する部分だけが異なる．

3.4.2 item-based RMCM2-CF

item-based RMCM2-CF は，評価値行列の転置行列 X^T に RMCM2 法を適用することで類似したアイテムのクラスターを抽出し，クラスター内のアイテムを高く評価（または所有）しているユーザへの推薦度を高くする．

item-based RMCM2-CF の手順は item-based RMCM-CF の手順と同様であり，評価値行列の転置行列 X^T に RMCM2 法を適用する部分だけが異なる．

4 数値実験

実データ (NEEDS-SCAN/PANEL データおよび MovieLens-100k データ) に対して提案法 (user-based RMCM2-CF および item-based RMCM2-CF) を適用し，ラフさを調節するパラメータ τ やクラスター数 C による推薦性能の変化を検証した．評価指標としては ROC-AUC 指標を用いた．

4.1 実験データ

4.1.1 NEEDS-SCAN/PANEL

NEEDS-SCAN/PANEL データは日本経済新聞社が収集した，996 世帯の 18 個の製品に対する所有の

有無を表すデータ (要素数 17,928 個) である．

評価値 r_{ij} は世帯 i が製品 j を所有している時 1，所有していない時 0 となる．

このデータの中で 1000 個をテストデータとし，テストデータに対する評価値を未評価値として 0 に置き換えたデータをトレーニングデータとした．

4.1.2 MovieLens-100k

MovieLens-100k データは GroupLens Research (`https://grouplens.org/`) が収集した，943 人のユーザが 1,682 個の映画に対して行った 100,000 個の評価値のデータである．そのうちの 10%(10,000 個) をテストデータとし，テストデータに対する評価値を未評価値の値に置き換えたデータをトレーニングデータとした．

評価値は $r_{ij} \in \{0, 1, 2, 3, 4, 5\}$ となり，$r_{ij} = 0$ の時が未評価値となる．本実験においては，user-based のクラスタリングベース協調フィルタリングを適用する場合は未評価値をユーザの平均評価値に，item-based のクラスタリングベース協調フィルタリングを適用する場合はアイテムの平均評価値に置き換えている．各未評価値の処理を行ったときのデータの一部を表 1，2 に示す．

4.2 評価指標

推薦性能は ROC-AUC 指標によって評価した．ROC (Receiver Operating Characteristic) は偽陽性率 (false positive rate; FPR) に対する真陽性率 (true positive rate; TPR) であり，AUC (Area Under the ROC Curve) は ROC 曲線の下側の面積である．ここで偽陽性率は実際陽性でないものが陽性であると判断された割合，真陽性率は実際陽性であるものが陽性であると判断された割合のことである．推薦が完全にランダムに行われた場合 AUC は 0.5 となり，AUC が 1 に近いほど推薦性能が良いといえる．

混同行列を表 3 に示す．

表 3　混同行列

実際	予測	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

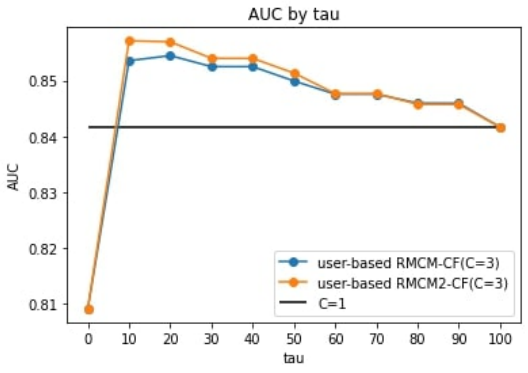

図 1　NEEDS-SCAN/PANEL データの $C = 3$ の時の τ による AUC の変化 (user-based)

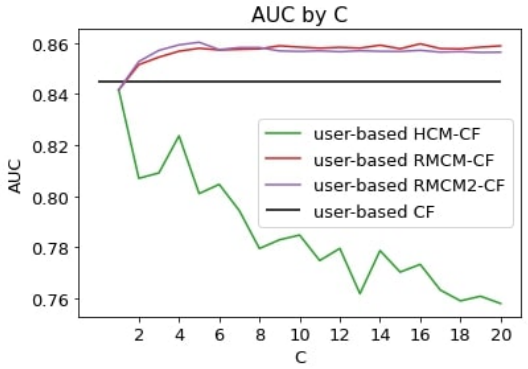

図 2　NEEDS-SCAN/PANEL データのクラスター数による AUC の変化 (user-based)

真陽性率および偽陽性率は以下の式のように計算される．

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \quad (18)$$

5 実験結果

5.1 NEEDS-SCAN/PANEL データにおける実験結果

5.1.1 user-based 手法の実験結果

user-based 手法 (user-based HCM-CF, user-based RMCM-CF, user-based RMCM2-CF, user-based CF) の結果を示す．まず，クラスター数 C が 3 の時の τ による AUC の変化を図 1 に示す．各 AUC の値は 10 回の試行の平均であり，$\tau = 0$ のとき，HCM-CF と同様の結果を示す．また，クラスター数 C が 1 の時は各製品の所有有無の平均値がすべての世帯における推薦度となり，AUC は 0.8416 である．図 1 から，HCM-CF より提案法が 0.039 ほど高い AUC を示すことが確認できる．また，最も高い AUC を示

表 1　未評価値をユーザの平均評価値にした MovieLens-100k の評価値行列 X

	1	2	3	⋯	1861	1862
1	5	3	4	⋯	3.61	3.61
2	4	3.75	3.75		3.75	3.75
⋮		⋮		⋱	⋮	
942	4.27	4.27	4.27	⋯	4.27	4.27
943	3.42	5	3.42		3.42	3.42

表 2　未評価値をアイテムの平均評価値にした MovieLens-100k の評価値行列 X

	1	2	3	⋯	1861	1862
1	5	3	4	⋯	3	3
2	4	3.20	3.04		3	3
⋮		⋮		⋱	⋮	
942	3.88	3.20	3.04	⋯	3	3
943	3.88	5	3.04		3	3

表 4　NEEDS-SCAN/PANEL データにおける最も高い AUC を持つ τ (user-based)

τ	0	10	20	30	40	50
回数	0	11	4	4	4	0
τ	60	70	80	90	100	
回数	0	0	0	0	0	

す τ は RMCM-CF の場合は $\tau = 20$，RMCM2-CF の場合は $\tau = 10$ となった．

クラスター数 C を 1 から 20 まで変化させたときの AUC の変化を図 2 に示す．RMCM-CF および RMCM2-CF の AUC は，τ を種々変化させたときの最大値を採用している．各 C において最も高い AUC を持つ τ の値を表 4 に示す．

表から，$\tau = 10$ の時 AUC が最も高くなる回数が多かったことが分かる．ただし，C を 17 以上にすると τ が 30, 40 の時の AUC が最も高くなった．

図 2 から，クラスター数に関わらず，RMCM-CF および提案法の RMCM2-CF が HCM-CF より高い AUC 値を持つことが確認できる．また，user-based CF の AUC は 0.8451 となり，RMCM-CF および RMCM2-CF の方が高い推薦性能を持つことができた．C の変化に注目すると，C を大きくすると HCM-CF の推薦性能は落ちる反面，RMCM-CF および RMCM2-CF の場合は安定した推薦性能をもつことも確認できる．

5.1.2　item-based 手法の実験結果

item-based 手法 (item-based HCM-CF, item-based RMCM-CF, item-based RMCM2-CF, item-based CF) の結果を示す．

クラスター数 $C = 3$ の時の τ による AUC の変化を図 3 に示す．$\tau = 100$ の時の結果は $C = 1$ の時の結果，つまり各世帯の所有有無の平均値がすべての製品における推薦度と同様となり，AUC は約 0.5013 となる．まず，$\tau = 100$ の時の AUC($C = 1$ の時の結果) が他の τ の AUC より低いことから，クラスタリングを行うことによって推薦性能が向上していることが分かる．また，RMCM-CF および RMCM2-CF が HCM-CF($\tau = 0$ の時の結果) より高い推薦性能を持つことが確認でき，その差は約 0.033 となった．item-based 手法においては RMCM-CF と RMCM2-CF の結果にほぼ差がなく，最も高い AUC を持つのも $\tau = 50$ の時で一致していることがわかる．

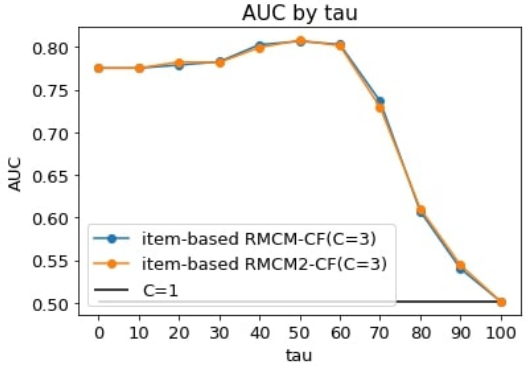

図 3　NEEDS-SCAN/PANEL データの $C = 3$ の時の τ による AUC の変化 (item-based)

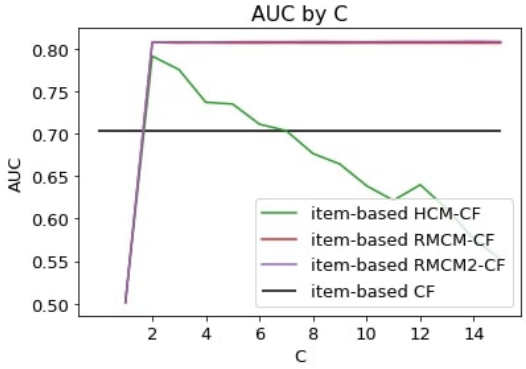

図 4　NEEDS-SCAN/PANEL データにおける最も高い AUC を持つ τ (item-based)

表 5　NEEDS-SCAN/PANEL データにおける最も高い AUC を持つ τ (item-based)

τ	0	10	20	30	40	50
回数	0	0	0	0	0	14
τ	60	70	80	90	100	
回数	0	0	0	0	0	

次に，$C = 1 \sim 15$ による AUC の変化を図 4 に示す．item-based 手法においては，NEEDS-SCAN/PANEL データのアイテム数が 18 ということから C を 15 までとしている．RMCM-CF および RMCM2-CF の AUC は，τ を種々変化させたときの最大値を採用している．各 C において最も高い AUC を持つ τ の値を表 5 に示す．

表から，すべての C において $\tau = 50$ の時の AUC が最も高かったことが分かる．

図 4 から，user-based 手法の結果と同様，RMCM-CF および RMCM2-CF が HCM-CF より高い推薦性能を持ち，item-based CF の AUC(0.7031) よりも高い AUC を持つことが確認できる．また，C を大きくすると HCM-CF の推薦性能は急激に落ちる反面，RMCM-CF および RMCM2-CF においてはほぼ一定で高い AUC(0.80 付近) を維持していることが分かる．

5.1.3　user-based 手法と item-based 手法との比較

AUC の順に並べると，user-based RMCM2-CF> user-based RMCM-CF>user-based CF>user-based HCM-CF>item-based RMCM2-CF≥item-based RMCM-CF>user-based CF となり，全体的に user-based 手法が item-based 手法より高い推薦性能を示した．

5.2　MovieLens-100k データにおける実験結果

5.2.1　user-based 手法の実験結果

$C = 3$ の時の τ による AUC の変化を図 5 に示す．各 AUC の値は 10 回の試行結果の平均である．また，$\tau = 100$ の時は各アイテムの平均評価値がすべてのユーザへの推薦度となり，AUC は 0.7143 となる．図 5 から，まず $\tau = 100$ の時の AUC が他の τ の AUC より低いことから，クラスタリングを行うことによって推薦性能が向上していることが確認できる．また，$\tau = 10 \sim 50$ の時は RMCM-CF と RMCM2-CF より HCM-CF が高い推薦性能を示すが，$\tau = 60$ 以上とすると AUC が急激に向上することがわかる．最も高い AUC を示す τ は RMCM-CF の場合 $\tau = 80$，RMCM2-CF の場合 $\tau = 90$ となり，HCM-CF との差は約 0.035 ほどとなった．

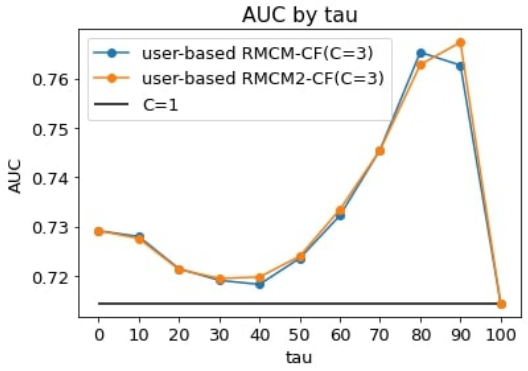

図 5　MovieLens-100k データの $C = 3$ の時の τ による AUC の変化 (user-based)

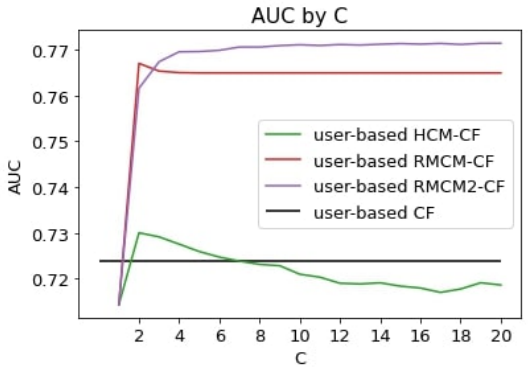

図 6　MovieLens-100k データのクラスター数による AUC の変化 (user-based)

表 6　MovieLens-100k データにおける最も高い AUC を持つ τ (user-based)

τ	0	10	20	30	40	50
回数	0	0	0	0	0	0
τ	60	70	80	90	100	
回数	0	0	1	18	0	

クラスター数 C を 1 から 20 まで変化させたときの AUC の変化を図 6 に示す．ここで，RMCM-CF および RMCM2-CF の AUC は，τ を種々変化させたときの最大値を採用している．各 C において最も高い AUC を持つ τ の値を表 6 に示す．

$C = 2$ の時を除外すると，すべての C において $\tau = 90$ の時 AUC が最大となった．

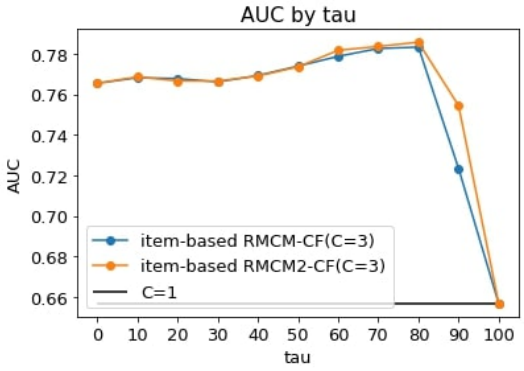

図 7　MovieLens-100k データの $C = 3$ の時の τ による AUC の変化 (item-based)

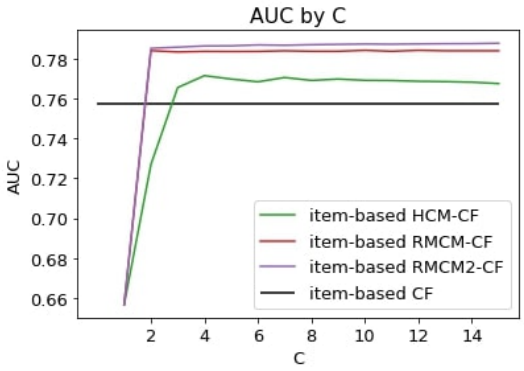

図 8　MovieLens-100k データのクラスター数による AUC の変化 (item-based)

図 6 から，RMCM-CF および RMCM2-CF が HCM-CF よりかなり高い AUC を持ち，その中でも RMCM2-CF が 0.77 付近の最も高い AUC を持つことが確認できる．user-based CF の AUC は 0.7240 となり，RMCM-CF および RMCM2-CF より低い推薦性能となった．また C を大きくすると HCM-CF の推薦性能は急激に落ちる反面，RMCM-CF および RMCM2-CF においては高い AUC を維持していることが分かる．

5.2.2　item-based 手法の実験結果

$C = 3$ の時の τ による AUC の変化を図 7 に示す．各 AUC の値は 10 回の試行結果の平均である．また，$\tau = 100$ の時は各ユーザの平均評価値がすべてのアイテムへの推薦度となり，AUC は 0.6568 となる．図から，$\tau = 10 \sim 80$ の時，RMCM-CF および RMCM2-CF が HCM-CF より高い推薦性能を持つことが確認できる．また，τ を 90 以上にすると AUC は急激に落ちることになる．AUC が最大となるのは $\tau = 80$ のときであり，HCM-CF との差は 0.0202 となる．

表 7　MovieLens-100k データにおける最も高い AUC を持つ τ (user-based)

τ	0	10	20	30	40	50
回数	0	0	0	0	0	0
τ	60	70	80	90	100	
回数	0	0	0	19	0	

クラスター数を 1 から 10 まで変化させたときの AUC の変化を図 8 に示す．ここで，RMCM-CF の AUC は，τ を種々変化させたときの最大値を採用している．各 C において最も高い AUC を持つ τ の値を表 7 に示す．

すべての C において，$\tau = 80$ の時 AUC が最も高くなるという結果となった．

図 8 から，これまでの結果と同様，RMCM-CF および RMCM2-CF が HCM-CF より高い推薦性能を持ち，C に関わらず高い推薦性能を維持していることが確認できる．その中でも最も高い AUC を持つのは RMCM2-CF となり，RMCM-CF より約 0.003 ほど高い AUC を持つことになった．また item-based CF の AUC は 0.7572 となり，クラスタリングベース手法より高い AUC を持つことになった．

5.2.3　user-based 手法と item-based 手法との比較

AUC が高い順に並べると，item-based RMCM2-CF≥item-based RMCM-CF>user-based RMCM2-CF>user-based RMCM-CF>item-based HCM-CF>user-based HCM-CF>item-based CF>user-based CF となり，全体的に item-based 手法が user-based 手法より高い推薦性能を示した．

5.2.4　MovieLens-100k データの視覚化

MovieLens-100k データの評価値行列 X に PCA を用いて次元圧縮を行うことで 2 次元に視覚化した結果を図 9 に示す．ただし，未評価値はユーザの平均評価値とした．ここで，$C = 3, \tau = 0$ とし，各データポイントの色は次元圧縮を行わない元のトレーニングデータに user-based クラスタリングを行ったときのラフメンバシップの値を表す．

図 9 から，データがはっきり 3 つに分かれていることが確認できる．また，PCA の結果として得ら

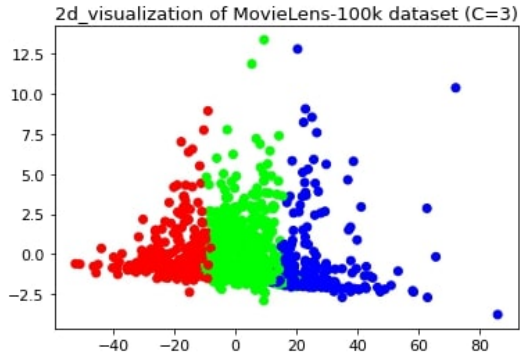

図 9　MovieLens-100k データを 2 次元に視覚化

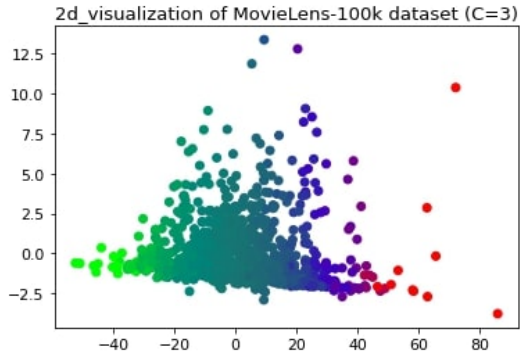

図 10　MovieLens-100k データを 2 次元に視覚化

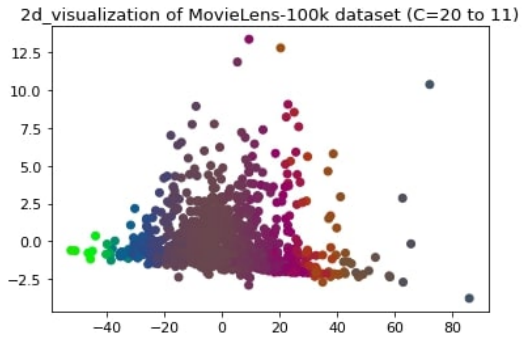

図 11　MovieLens-100k データを 2 次元に視覚化

れる第一主成分 (x 軸) はユーザの平均評価値，第二主成分 (y 軸) はユーザが与えた評価値の分散であることが確認できた．x 軸を基準にクラスタリングが行われていることからユーザの平均評価値によってデータがグループ分けされているといえる．AUC が最大になる $\tau = 80$ とすると，ラフメンバシップは図 10 のように変化する．図より，データポイントが多く集まっている部分が似たようなラフメンバシップ値を持つようになっていることが確認できる．さらに $C = 20$ とし，得られたメンバシップを 3 次元に次元圧縮することで RGB 色として表すと，図 11 となる．初期クラスター数は 20 だったが，$\tau = 80$ ということから空クラスターが多く発生し，最終的なクラスター数は 11 となった．$C = 3$ の時に得ら

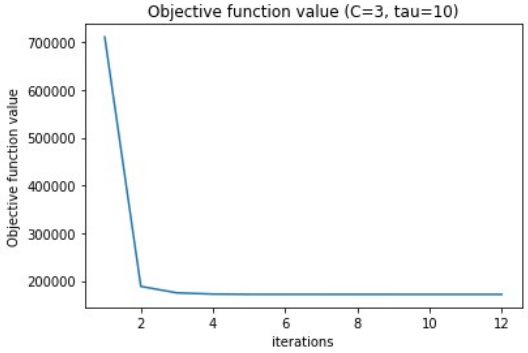

図 12　MovieLens-100k データの目的関数値の変化

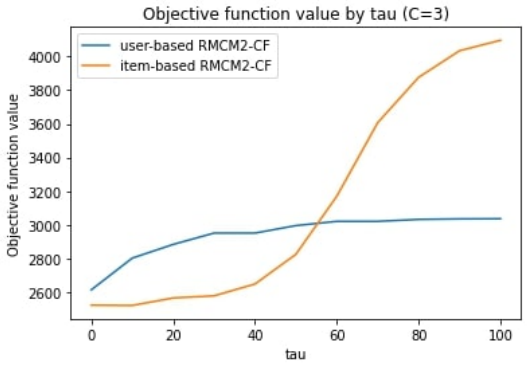

図 13　NEEDS-SCAN/PANEL データの $C = 3$ の時の τ による目的関数値の変化

れるラフメンバシップ値の分布と似たような分布を持ち，C に関わらず似たようなクラスターが取れていると考えられる．

5.3　目的関数値

MovieLens-100k データにおける user-based RMCM2-CF の目的関数値の変化を図 12 に示す．ただし，$C = 3, \tau = 10$ とし，目的関数値は 1 回の試行の結果となる．図から，目的関数値が単調減少していることが確認できる．

次に，C や τ による目的関数値を観察した．まず，C を固定して τ を種々変化させた時の各試行の収束後の目的関数値を図 13 に示す．また，目的関数値は 10 回の試行の平均である．図から，データに関わらず user-based と item-based の両方の手法において τ を大きくすると目的関数値も単調増加することが確認できる．

次に，τ を固定して C を変化させた時の目的関数値を図 14 に示す．図から，データに関わらず user-based と item-based の両方の手法において C を大

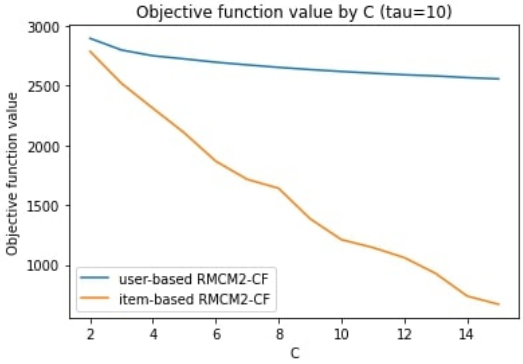

図 14　NEEDS-SCAN/PANEL データの $\tau = 10$ の時の C による目的関数値の変化

表 8　NEEDS-SCAN/PANEL データの最小目的関数値

	NEEDS-SCAN/PANEL	
	user-based	**item-based**
C	50	15
τ	0	0
目的関数値	1491.0892	244.0

きくすると目的関数値は単調減少することが確認できる．

これより，C が大きいほど，そして τ が小さいほど ($\tau = 0$) 目的関数値が小さくなると考えられる．これらを確認するため，Python の Optuna ライブラリを用いて目的関数値が最小となる C および τ を探した．C の範囲は 2 ～ 50(NEEDS-SCAN/PANEL データの item-based RMCM2-CF の場合はデータサイズの関係上 2 ～ 15)，τ の範囲は 0 ～ 100 とした．また，iteration は 500 とした．

これより，C が最大で，τ が最小 ($\tau = 0$) のとき目的関数値が最小となることが確認できる．

次に，user-based RMCM2-CF と item-based RMCM2-CF の各手法においてハイパーパラメータ (C, τ) を固定した上でクラスター中心の初期値に対する AUC と目的関数値の関係を観察した．その関係を散布図で表した結果を図 15~26 に示す．試行回数は 500 とした．ここで空クラスターが発生すると C の値が変化し C に依存されるため，空クラスターが発生しなかった試行のみを採用する．次に，目的関数値と AUC の相関係数を表 9 および 10 にまとめた．

図 15~20 及び表 9 から，NEEDS-SCAN/PANEL データの場合，$C = 3, \tau = 30$ の試行を除くと，全体的に負の相関があることが確認できる．

図 21~26 から，MovieLens-100k データの場合，

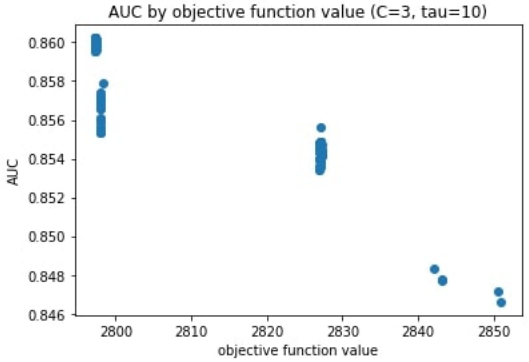

図 15　NEEDS-SCAN/PANEL データの $C = 3$, $\tau = 10$ の時の AUC と目的関数値 (user-based RMCM2-CF)

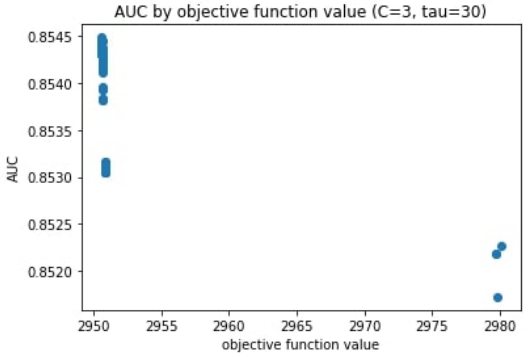

図 16　NEEDS-SCAN/PANEL データの $C = 3$, $\tau = 30$ の時の AUC と目的関数値 (user-based RMCM2-CF)

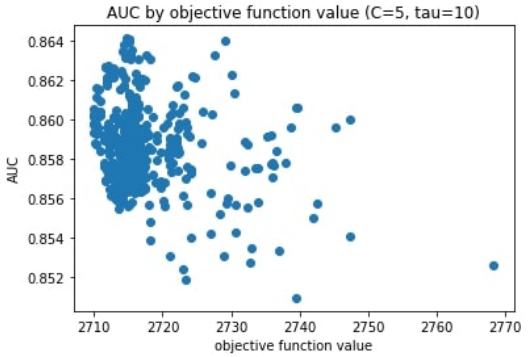

図 17　NEEDS-SCAN/PANEL データの $C = 5$, $\tau = 10$ の時の AUC と目的関数値 (user-based RMCM2-CF)

表 9　NEEDS-SCAN/PANEL データにおいての目的関数値と AUC の相関係数

	user-based	**item-based**
$C = 3, \tau = 10$	-0.8072	-0.6567
$C = 3, \tau = 30$	-0.3583	-0.0579
$C = 5, \tau = 10$	-0.2134	-0.6595

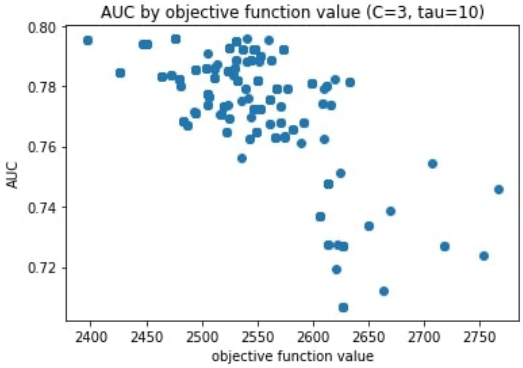

図 18　NEEDS-SCAN/PANEL データの $C = 3$, $\tau = 10$ の時の AUC と目的関数値 (item-based RMCM2-CF)

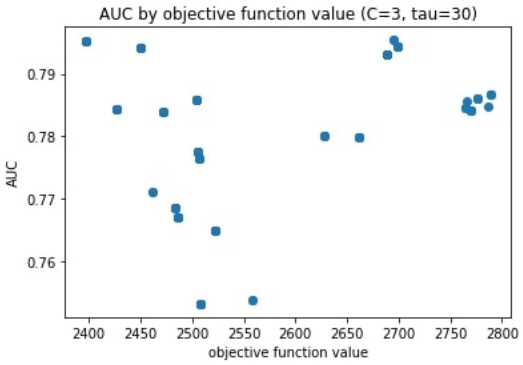

図 19　NEEDS-SCAN/PANEL データの $C = 3$, $\tau = 30$ の時の AUC と目的関数値 (item-based RMCM2-CF)

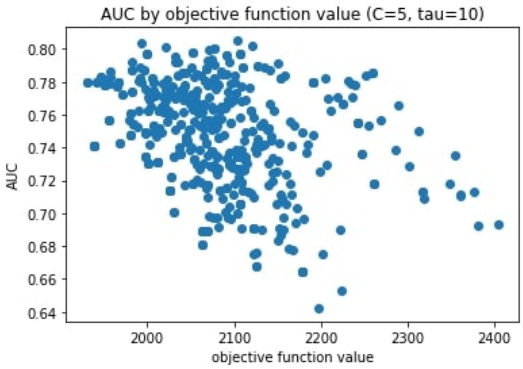

図 20　NEEDS-SCAN/PANEL データの $C = 5$, $\tau = 10$ の時の AUC と目的関数値 (item-based RMCM2-CF)

複数の極小値が存在し，その間は負の相関が成立する傾向が見られる．表 10 からも，C や τ を変更しても -1 に近い相関関係を維持していることが確認できる．

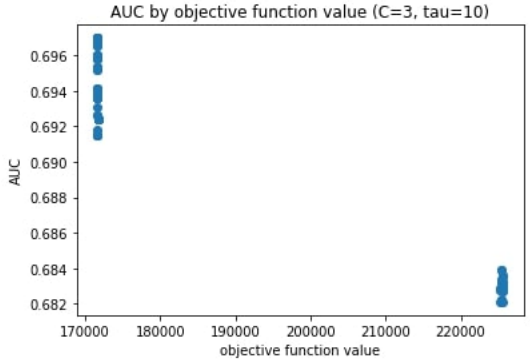

図 21　MovieLens-100k データの $C = 3$, $\tau = 10$ の時の AUC と目的関数値 (user-based RMCM2-CF)

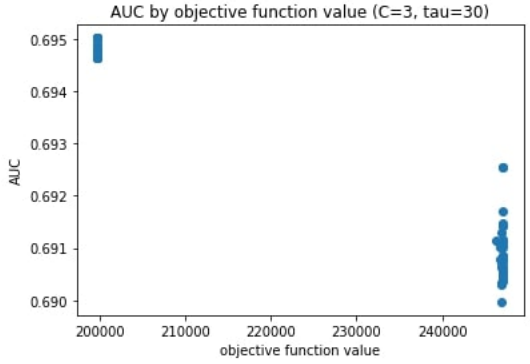

図 22　MovieLens-100k データの $C = 3$, $\tau = 30$ の時の AUC と目的関数値 (user-based RMCM2-CF)

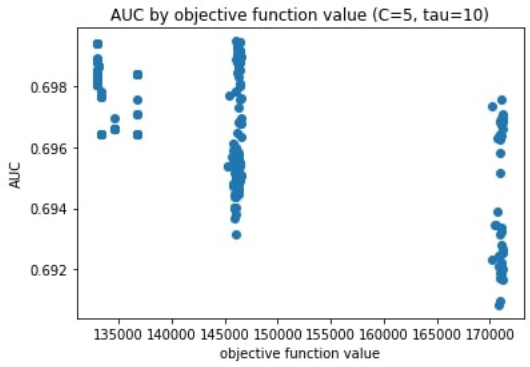

図 23　MovieLens-100k データの $C = 5$, $\tau = 10$ の時の AUC と目的関数値 (user-based RMCM2-CF)

5.4　空クラスターの発生

RMCM 法および RMCM2 法では，アルゴリズムの繰り返し過程でクラスター中心が一致すると，対象が一つも含まれない空クラスターが発生する．特にアルゴリズムの原理上，RMCM2 法は $\tau = 100$ のときは必ず空クラスターが発生する．RMCM-CF の

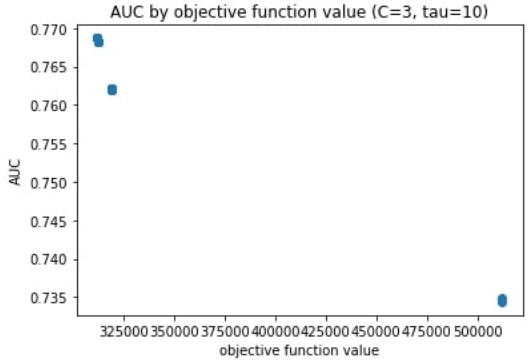

図 24　MovieLens-100k データの $C = 3$, $\tau = 10$ の時の AUC と目的関数値 (item-based RMCM2-CF)

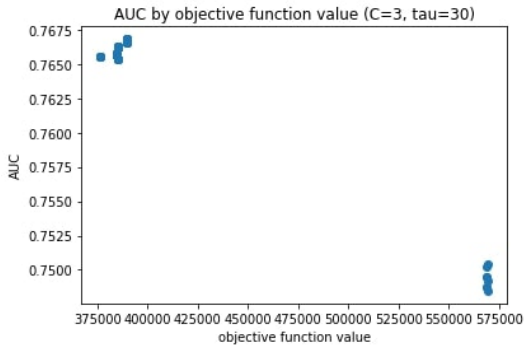

図 25　MovieLens-100k データの $C = 3$, $\tau = 30$ の時の AUC と目的関数値 (item-based RMCM2-CF)

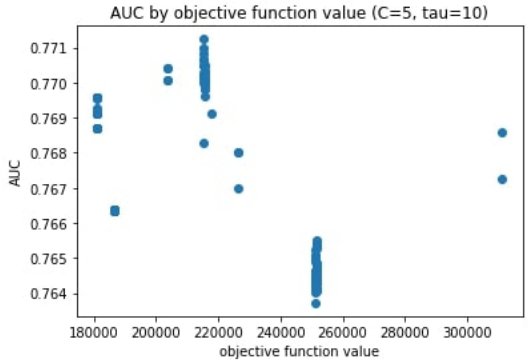

図 26　MovieLens-100k データの $C = 5$, $\tau = 10$ の時の AUC と目的関数値 (item-based RMCM2-CF)

表 10　MovieLens-100k データにおいての目的関数値と AUC の相関係数

	user-based	**item-based**
$C = 3, \tau = 10$	-0.7782	-0.8027
$C = 3, \tau = 30$	-0.9763	-0.8980
$C = 5, \tau = 10$	-0.5911	-0.7493

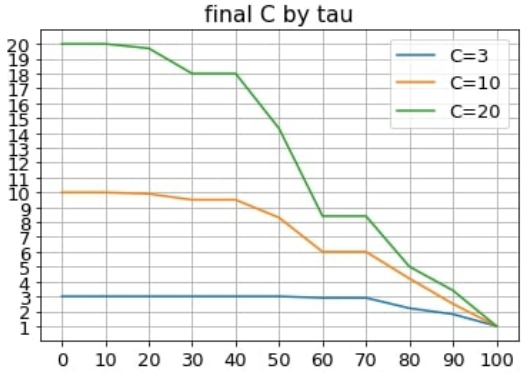

図 27　NEEDS-SCAN/PANEL データにおける τ による最終クラスター数

場合は空クラスターが発生しても推薦性能に大きな影響を及ぶことはなかったが，RMCM2-CF の場合空クラスターが発生すると推薦性能が低下することが確認できている．その影響を除去するため本実験では空クラスターを除去しており，初期クラスター数と更新を終えた最終クラスター数は一致しない場合が多い．

また，空クラスターの発生頻度は τ および C に影響を受ける．NEEDS-SCAN/PANEL データにおける τ による最終クラスター数を図 27 に示す．また，最終クラスター数は 10 回の試行の平均を取っている．

図から，空クラスターは τ が大きくなるほど発生する傾向があり，$\tau = 100$ の時はどの C においても最終クラスター数は 1 に収束することが確認できる．また，C を大きくすることで小さい τ でも空クラスターが発生しやすい傾向があることが確認できる．

各 C における最終クラスター数の平均を以下に示す．NEEDS-SCAN/PAENL データでの最終クラスター数の平均を図 28 に，MovieLens-100k データの最終クラスター数の平均を図 29 に示す．ここで最終クラスター数の平均とは，各 C に対して τ を $\{0, 10, \cdots, 90, 100\}$ としたとき得られる最終クラスター数の平均を意味する．

図 28 を見ると，item-based RMCM2-CF の方が空クラスターが発生しやすいことが確認でき，図 29 を見ると user-based RMCM2-CF の方が空クラスターが発生しやすいことが確認できる．NEEDS-SCAN/PANEL データの大きさは 996×18 であり，MovieLens-100k データの大きさは 943×1682 であることを考慮すると，データ数が多い場合において空クラスターが発生しにくいことが読み取れる．また，二値データ $(0, 1)$ の NEEDS-SCAN/PANEL データ

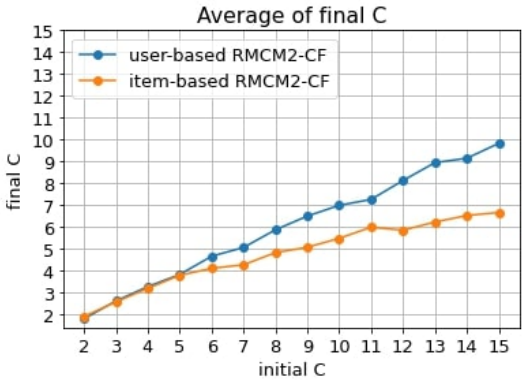

図 28　NEEDS-SCAN/PANEL データにおいての最終クラスター数の平均

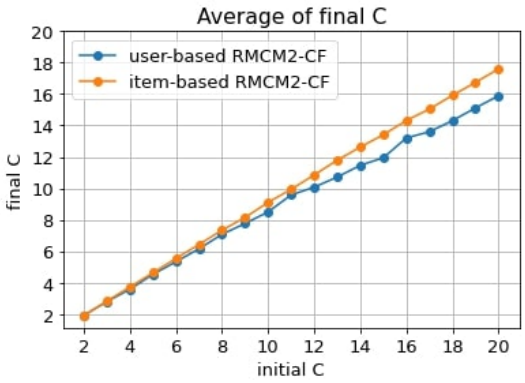

図 29　MovieLens-100k データにおいての最終クラスター数の平均

がそうでない MovieLens-100k データより空クラスターが発生しやすいことも確認できる．

6　考察

6.1　未評価値について

MovieLens-100k データは未評価値が全体の約 94% を占める非常に疎なデータであり，未評価値の影響を大きく受ける．本実験では，MovieLens-100k データの未評価値を user-based 手法の場合ユーザの平均評価値に，item-based 手法の場合アイテムの平均評価値に置き換えている．先行研究 [10] では，未評価値がユーザの平均評価値のデータに対して item-based RMCM-CF を適用していたが，その時の AUC は 0.73 付近となり，未評価値をアイテムの平均評価値にした時の AUC(約 0.78) よりかなり低くなった．また，item-based HCM-CF との差も少なく，RMCM 法による推薦性能の向上が見られなかった．この原因として，データの未評価値をユーザの平均評価値にすることで，似たような評価値を持つアイテムが非常に多くなる．つまり既にクラスタリングされているようなデータになると考えられる．その結果，HCM-CF と RMCM-CF との差が少なかったと考えられる．逆の場合 (未評価値をアイテムの平均評価値に代入したデータに対して user-based 手法を適用した場合) も同様のことが考えられ，どの手法を用いるかによって未評価値の処理を合わせる必要があると考えられる．

6.2　空クラスターについて

本実験では，RMCM-CF および RMCM2-CF で空クラスターが発生した場合はその空クラスターを除去している．これは，RMCM2-CF の場合に空クラスターの発生が AUC に悪影響を与えるようになるためである．例えば，NEEDS-SCAN/PANEL データに user-based RMCM2-CF を適用した場合，$C = 2, \tau = 80$ のとき空クラスターが発生しなかった試行では AUC が 0.8456 となるが，空クラスターが発生した場合は AUC が最低 0.6857 となり大きな推薦性能の低下が見られた．その原因としては，値の更新が行われてなかったことが理由として挙げられる．RMCM2 法で空クラスターが発生する時点を観察してみた結果，すべての試行において iteration が 0 のとき，つまり更新が一回も行われなかった時点で空クラスターが発生することを確認した．このようにクラスター中心の更新が行われず，初期化されたままのクラスター中心を用いて AUC の計算を行ったため AUC が収束しなくなり，RMCM2 法では空クラスターが出現すると AUC が低くなったと考えられる．

提案法の場合，$\tau = 100$ の時クラスター数は 1 に収束する．その時のクラスター割り当て状況を見ると，クラスター中心から各対象までの距離の総和（これから距離の総和とする）が最小になるクラスターがすべての対象に対してメンバシップが 1 となり，それ以外のクラスターのメンバシップはすべて 0 になる．空クラスターが発生し始めるとき ($\tau = 50$ 付近) の試行でのメンバシップ行列およびクラスター中心行列を確認すると，距離の総和が 2 番目になるクラスターから空クラスターとなることが分かった．これより，RMCM2 法では τ が大きくなるほど距離の総和が小さい順から距離の総和が最小になるクラスターに吸収されるようになると考えられる．試行によって空クラスターの発生する τ が変わることもこ

の距離の総和の差が関わるためであると考えられる．

空クラスターの発生頻度はクラスター数 C および τ の影響を受けることが確認でき，τ を大きくしたり C を大きくすることで空クラスターの発生頻度が高くなった．τ が大きくなると，固体数の少ないクラスターのラフメンバシップは小さくなる．これより，クラスター中心は他のクラスターに近くなり，繰り返すと最終的に他のクラスター中心と一致してしまい，空クラスターが発生するのではないかと考えられる．同じ理由で，C が大きくなると，固体数の少ないクラスターが多くなるため，空クラスターが発生しやすくなる．

また，空クラスターを除去することによって最適なクラスター数を見つけることができるかもしれないという期待があった．実際，τ が大きければ異なる C に対しても最終クラスター数がほぼ同様になる試行があることを確認できている．しかし，これは C が非常に大きい時 (データサイズの半分以上など) に見られる現象であり，全体的には空クラスター数は C が大きくなるほど増加する傾向を見せ，一定のクラスター数になることはなかった．これより，空クラスターの除去によって最適なクラスター数を見つけることは難しいと考えられる．ただし，C による AUC の変化を見ると，RMCM2-CF は C に関わらず高い AUC を維持していることから，推薦性能だけがを見れば最適なクラスター数を見つけることの重要性は比較的に低いと考えられる．

6.3 AUC について

MovieLens-100k データの場合，クラスター数 C を 10 以上にすると，$\tau = 80$ のときより $\tau = 90$ のとき AUC が最大となる場合が多くなった．これは，クラスター数が大きくなると近傍外のクラスターの影響が小さくなり，オーバーラップが大きい MovieLens-100k データにおいては τ を大きくする必要があったためであると考えられる．

NEEDS-SCAN/PANEL データおよび MovieLens-100k データのクラスター数による AUC の変化 (図 2，図 4，図 6，図 8) を見ると，クラスター数に関わらず HCM-CF より RMCM-CF および RMCM2-CF で安定した結果を出すことが分かる．これは，RMCM-CF および RMCM2-CF では τ を調節することでデータの近傍情報を用いることができ，遠い個体も計算に入れるためクラスター数の影響が小さくなると考えられる．

item-based RMCM2-CF の場合，user-based RMCM2-CF より C による AUC の変化が小さいことが確認できた．今回のデータの場合は τ が比較的に大きい時に AUC が最大となったため，結果として得られる最終クラスター数は初期クラスター数と比べて大幅に減少している可能性が高い．これより，小さい C での結果と同じ AUC を持つようになり，さらに比較的小さい C で高い AUC を出すことができたため，C による AUC の影響が小さくなったと考えられる．

NEEDS-SCAN/PANEL データでは user-based RMCM2-CF の場合 $\tau = 10 \sim 40$ のとき，item-based RMCM2-CF の場合 $\tau = 50$ のときに AUC が最大となり，オーバーラップが小さくクラスター構造が比較的に明確なデータであると考えられる．それに対して，MovieLens-100k データの場合，τ が 80 または 90 のとき AUC が最大となり，オーバーラップが大きくクラスター構造が明確でないデータであると考えられる．

item-based 手法の場合，$C = 1(\tau = 100)$ のときの AUC が他の C のときの AUC よりかなり低いことが確認できる．これは，item-based 手法の場合，$C = 1$ の時のクラスター中心 b_{ci} はユーザがアイテムを高く評価する傾向があれば高く，そうでなければ低くなり，嗜好度とは関係が薄くなるためである．特に NEEDS-SCAN/PANEL では user-based 手法と item-based 手法の $C = 1$ での AUC の差が大きい（user-based の場合は 0.8 付近，item-based の場合は 0.5 付近）が，これは NEEDS-SCAN/PANEL データが家電を含む製品の所有有無を表すためだと考えられる．例えば，誰もが持っている人気製品（洗濯機やエアコンなど）とそうでない製品（衣類乾燥機など）がわりとはっきりしているが，持っている製品数はそこまで大きく変わらない．そのため，user-based の $C = 1$ では高い AUC を持つが，item-based の場合は低い AUC を持つことになる．また，MovieLens-100k データの item-based 手法の場合，$C = 1$ のとき AUC が約 0.66 となるが，これは未評価値の前処理による推薦性能の向上だと考えられる．実際，未評価値の前処理を行わなかった場合の AUC は 0.5 を下回った．

メモリベース協調フィルタリングの中では，一般的に item-based CF が user-based CF より高い推

薦性能を持つ傾向があると知られている．ただし，NEEDS-SCAN/PANEL データでは user-based 手法が高い推薦性能を持つ結果となった．これは，今回用いた NEEDS-SCAN/PANEL データのサイズが 996×18 であり，ユーザ空間と比べるとアイテム空間の情報が非常に少ないことが原因であると考えられる．

6.4 目的関数値について

本実験においては，NEEDS-SCAN/PANEL データおよび MovieLens-100k データの両方で目的関数値は C が大きい時，τ が小さい時小さくなる傾向が見られた．まず，C が大きいと目的関数値が小さくなる理由としては，C が大きくなるにつれ各クラスターに含まれる個体数が少なくかつクラスター中心との距離の合計も小さくなるためだと考えられる．また，τ が小さいと目的関数値が小さくなる理由としては，τ が大きくなるにつれより遠くの個体との距離も目的関数値の計算に含まれるためだと考えられる．

目的関数値と AUC の関係 (図 15~26) から，全体的に負の相関があり，目的関数値が小さければ AUC は高くなる傾向が見られる．これより，RMCM2 法の目的関数の設定に妥当性があることが示唆され，多数の試行の中から目的関数値が最小になる解を採用することで，より良いクラスター構造である可能性が高いものを選択可能になると考えられる．ただし，試行によっては目的関数値と AUC の間にこのような関係が成立しない場合もあり，目的関数値が小さくなる複数の試行を検証する必要があると考えられる．

7 おわりに

本研究では，2 種類の目的関数ベースの RMCM 法に基づく協調フィルタリング (user-based RMCM2-CF および item-based RMCM2-CF) を提案し，実データである NEEDS-SCAN/PANEL データおよび MovieLens-100k データに適用し，推薦性能の変化を観察した．

実験結果から，NEEDS-SCAN/PANEL データにおいて提案法の RMCM2-CF が HCM-CF およびメモリベース協調フィルタリングより著しく高い推薦性能を持ち，user-based RMCM2-CF の場合は τ が 10〜20 のとき，item-based RMCM2-CF の場合は τ が 50 のとき AUC が最大になる場合が多かった．また，MovieLens-100k データにおいては提案法の RMCM2-CF が最も高い推薦性能を持ち，user-based RMCM2-CF の場合は τ が 90 のとき，item-based RMCM2-CF の場合は τ が 80 のとき AUC が最も高かった．user-based 手法と item-based 手法を比較すると，MovieLens-100k データのように十分なアイテム情報が与えられた場合において item-based 手法が user-based 手法より高い推薦性能を持つことが確認できた．また，目的関数値と AUC の関係には負の相関が見られる傾向があることを確認した．

これより，ラフ集合理論に基づく不確実性の取り扱いが協調フィルタリングタスクにおいて有効であり，かつ RMCM2 法の目的関数の設定には妥当性があることが示唆された．また，提案法をベースにして様々な機構の導入を試みることでさらに有効な協調フィルタリング手法を提案することも期待できる．

謝辞

本研究は大阪公立大学大学院情報学研究科の生方誠希准教授，本多克宏教授の御指導のもとに行われたものであり，心より感謝の意を表します．

参考文献

[1] J. Breese, D. Heckerman, and C. Kadie : Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 43/52 (1998)

[2] G. Linden, B. Smith, and J. York : Amazon.com recommendations: Item-to-item collaborative filtering, IEEE Internet computing, **7**-11, 76/80 (2003)

[3] Y. Koren, R. Bell, and C. Volinsky : Matrix factorization techniques for recommender systems, Computer, **42**-8, 30/37, (2009)

[4] J. MacQueen : Some methods of classification and analysis of multivariate observations, Proceeding of 5th Berkeley Symposium on Math. Stat. and Prob., 281/297 (1967)

[5] Z. Pawlak : Rough sets, International journal of computer & information sciences, **11**-5, 341/356 (1982)

[6] 生方 誠希 : ラフ集合に基づく C-Means 型クラスタリングの展開, 日本知能情報ファジィ学会誌, **32**-4, 121/127 (2020)

[7] S. Ubukata, A.Notsu, and K. Honda : General formulation or rough C-means clustering, International Journal of Coumputer Sceince and Network Security, **17**-9, 29/38 (2017)

[8] S. Ubukata, H. Kato, A. Notsu and K. Honda : Rough Set-Based Clustering Utilizing Probabilistic Membership, Journal of Advanced Computational Intelligence and Intelligent Informatics, **22**-6, 956/964 (2018)

[9] S. Ubukata, A. Notsu, and K. Honda : Objective function-based rough membership C-means clustering, Information Sciences, **548**, 479/496 (2021)

[10] H. Kim, S. Ubukata, A. Notsu, and K. Honda : Two Types of Collaborative Filtering Membership C-Means Clustering : proceeding of The 22nd International Symposium on Advanced Intelligent Systems, 118/123 (2021)