

# ハイブリッド型メタ学習モデルによるコールドスタート問題の緩和

佐藤 立弥<sup>†</sup>    伊東 聖矢<sup>††</sup>    大原 剛三<sup>††</sup>

<sup>†</sup> 青山学院大学大学院理工学研究科理工学専攻知能情報コース    〒 252-5258 神奈川県相模原市中央区淵野辺 5-10-1

<sup>††</sup> 青山学院大学理工学部情報テクノロジー学科    〒 252-5258 神奈川県相模原市中央区淵野辺 5-10-1

E-mail: <sup>†</sup>c5621192@aoyama.jp, <sup>††</sup>{s.ito,ohara}@it.aoyama.ac.jp

あらまし 推薦システムには評価履歴が少ないユーザやアイテムの推薦精度が低下するコールドスタート問題がある。近年、この問題を緩和する手法として、少数データのみでモデルの調整を可能とするメタ学習を推薦システムに適用した手法が提案されている。しかし、従来のメタ学習を用いた推薦システムでは、データをユーザ単位で分割するユーザタスクまたはアイテム単位で分割するアイテムタスクのいずれかのみを用いているため、ユーザコールド、アイテムコールドのいずれかしか適切に緩和できない。そこで本研究では、ユーザタスクとアイテムタスクのそれぞれでメタ学習したモデルを統合することで、いずれのコールドスタート問題も緩和可能な手法を提案する。

キーワード ハイブリッド型推薦, セッションベース推薦, メタ学習, コールドスタート問題, 深層学習モデル

## 1 はじめに

多くの推薦システムは蓄積された評価履歴を基にユーザの嗜好に合ったアイテムを推薦している。しかし、そのような推薦システムでは、評価履歴が十分にないユーザに対する推薦精度が低くなるコールドスタート問題がある [1]。特に、評価履歴が少数であるユーザに対する推薦精度の低下をユーザコールド、評価履歴が少数であるアイテムに対する推薦精度の低下をアイテムコールドと呼ぶ。そして、評価履歴が少数のユーザは新規ユーザである場合が多く、推薦が有用でないと新規ユーザはサービスから離れてしまい、サービスの利用者数が増えない。そのため、新規ユーザへの適切な推薦は重要となる。また、同様に新規アイテムの適切な推薦も重要となる。このような理由から、コールドスタート問題を緩和する研究が行われている。

コールドスタート問題を緩和する研究として、商品の説明テキストや推薦対象アイテムとは異なるドメインにおける評価履歴などの豊富な補助情報を利用する研究がある [2, 3]。しかし、補助情報を利用する研究は特定のデータセットに特化しており、推薦対象によって利用できる特徴が異なることや、他ドメインの情報を利用できるとは限らないため、それらの手法を任意のドメインに適用することは困難である。

そのため、ユーザとアイテムの基本的な属性情報と評価履歴のみを用いた場合でもコールドスタート問題を緩和できる手法が提案されている [4, 5]。その中で、近年ではメタ学習を用いた推薦手法が良い結果を出している [6]。メタ学習は複数のタスクから事前に学習の仕方を獲得することで、新規タスクを少数データで学習することを可能とする。そのため、評価履歴が少数の場合にもモデルを最適化することが可能となり、コールドスタート問題を緩和することができる。メタ学習を最初に推薦システムに適用した Lee ら [7] の研究では、代表的なメタ学習手法である Model Agnostic Meta Learning (MAML) [8]

を推薦システムに適用している。また Dong ら [9] は、メタ学習によって獲得する学習の仕方が 1 つだけという問題点に対して、ユーザの特徴によって学習の仕方を変化させる手法を提案している。しかし、メタ学習を用いた推薦システムでは、ユーザ単位またはアイテム単位にデータを分割してメタ学習を行っており、主にユーザコールドまたはアイテムコールドのいずれかしか緩和されないという問題点がある。

そこで本研究では、メタ学習を用いた推薦システムにおいて、ユーザコールドとアイテムコールドの影響を同時に緩和することを目的とする。具体的には、ユーザ単位またはアイテム単位でそれぞれメタ学習したモデルを組み合わせた手法を提案する。これにより、各データ分割方法でメタ学習したモデル単体ではユーザコールドまたはアイテムコールドのどちらかのみを緩和可能であっても、両モデルを組み合わせることで両者の特徴を備えることでコールドスタート問題全体の緩和を可能にする。また、実際に推薦の評価に用いられるベンチマークデータセットを用い、評価履歴が少数であるコールドの場合と十分あるウォームの場合の精度を比較し、ユーザコールドとアイテムコールドのそれぞれの緩和度合から、提案手法の有効性を実験的に評価する。

## 2 関連研究

### 2.1 深層学習を用いた推薦システム

推薦システムは、協調フィルタリングとコンテンツベースフィルタリング、2 つを組合わせたハイブリッド手法の 3 種類に大別できる [10]。基本的には、協調フィルタリングとコンテンツベースフィルタリングそれぞれの短所を相互に補強するハイブリッド手法が多用される。また、深層学習の発展に伴い、ハイブリッド手法とともにコールドスタート問題に対して深層学習モデルが広く活用されている。その中でも、単純な構造を

持つことから様々な手法の基礎として多く用いられる Neural Collaborative Filtering [11] という手法がある。この手法では、ユーザの潜在表現の獲得、アイテムの潜在表現の獲得、および、2つの潜在表現のかけ合わせに多層パーセプトロンを用いている。これにより、ユーザとアイテムの複雑な相互作用を表現することが可能となる。

## 2.2 コールドスタート問題の緩和に関する研究

コールドスタート問題における問題設定は、レビューテキストや商品の説明テキスト、推薦対象以外のドメインなどの豊富な補助情報が使える設定、および、ユーザとアイテムの属性情報と評価履歴の限られた情報のみができる設定に大別できる。

豊富な補助情報が使える問題設定であるレビューテキストを用いた研究として、Tsai ら [2] は、商品の説明テキストを用いてコールドでないアイテムを基にコールドアイテムの評価履歴に仮想的な履歴を追加する手法を提案している。また、目的と異なるドメインの情報をを用いた手法としてクロスドメインレコメンデーションがある。Wang ら [3] は、推薦対象ではない補助ドメインにおけるユーザ間類似度を利用し、目的ドメインにおけるコールドユーザに対する推薦精度を向上させている。これらの手法では、豊富な補助情報を用いることでコールドスタート問題を緩和することは可能であるが、補助情報がないデータセットには対応できない。そのため、限られた情報のみの問題設定においてコールドスタート問題を緩和する方法が研究されている [4, 5]。

限られた情報のみを用いる研究として、Chae ら [4] は、既知の評価履歴から仮想的な評価履歴を生成し、評価値行列を拡張する手法を提案している。この手法ではデータセット全体の疎性も解消されるため、コールドスタート問題の緩和だけでなく、全体の推薦精度も向上させている。また、推薦システムで用いられるユーザまたはアイテムの埋め込み表現を補強する手法も提案されており、Zhao ら [5] は、コールドアイテムの埋め込み表現を属性情報を用いて補強する手法を提案している。補強された埋め込み表現を推薦モデルの入力に用いることで、コールドアイテムに対する推薦精度の向上に成功している。これらの研究に対して、近年ではメタ学習を用いた推薦手法が良い結果を出している [6]。

## 2.3 メタ学習を用いた推薦システム

メタ学習は、目的のタスクと関連するタスクから学習の仕方を事前に学習することで、少数の学習データに対する性能を高める手法である。具体的には、複数のタスクを用いた事前学習によってメタ知識と呼ばれる事前知識を獲得し、目的のタスクの学習・推論においてそのメタ知識を活用する。メタ学習は推薦システムにおいてコールドスタート問題の緩和のために用いられ、タスクに応じてデータの分割方法が異なる。タスクは、1人のユーザの評価傾向を捉えることを1つのタスクとするユーザタスクと、1つのアイテムの評価傾向を捉えることを1つのタスクとするアイテムタスクがある。つまり、図1に示すように、1つの予測対象に対して2種類のタスクに対するデータ生

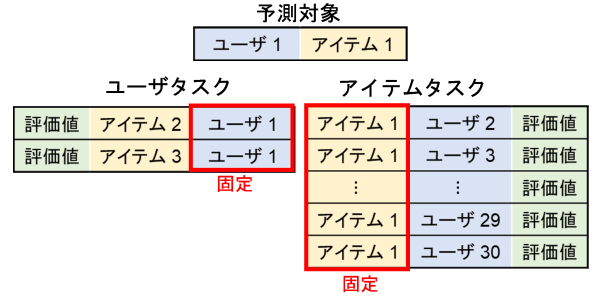


図 1: 2 種類のタスクに対するデータ生成法

成方法をとることができる。具体的には、ユーザ  $u$  の属性情報や評価履歴の潜在表現などの特徴を  $\mathbf{x}_u$ 、アイテム  $i$  の属性情報や評価履歴の潜在表現などの特徴を  $\mathbf{x}_i$ 、 $\mathbf{x}_u$  と  $\mathbf{x}_i$  を結合したベクトルを  $\mathbf{x}_{ui}$ 、ユーザ  $u$  によるアイテム  $i$  の評価値を  $r_{ui}$ 、ユーザ  $u$  により評価済みのアイテム集合を  $I_u$  とすると、ユーザ  $u$  の評価傾向を捉えるためのタスク  $T_u$  に対するデータ集合は  $\{(\mathbf{x}_{ui}, r_{ui}) | i \in I_u\}$  となる。同様に、アイテム  $i$  を評価済みのユーザ集合を  $U_i$  とすると、アイテム  $i$  の評価傾向を捉えるためのタスク  $T_i$  に対するデータ集合は  $\{(\mathbf{x}_{ui}, r_{ui}) | u \in U_i\}$  となる。また、メタ学習は主に Metric-base, Model-base, Optimization-base の 3 つに大別される [14]。それぞれの分類における代表的な手法として、Metric-base の Prototypical Networks [17]、Model-base の Neural Process (NP) [15]、Optimization-base の Model-Agnostic Meta-Learning (MAML) [8] がある。その中でも、メタ学習を用いた推薦システムでは MAML を基礎とした手法が多い。

MAML は様々なタスクに適応しやすいモデルパラメータの初期値を獲得する手法である。初期値の獲得は inner update と outer update の 2 段階のレベルの学習に分かれており、事前に 2 つの学習を繰り返すことでメタ知識を獲得する。Inner update は 1 つのタスクに特化してモデルを更新する学習であり、outer update は複数のタスクからメタ知識を獲得する学習である。ここで、説明変数  $\mathbf{x} \in X$  と目的変数  $y \in Y$  の組み合わせで構成されるデータ数  $N_k$  のタスクを  $T_k = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_k}$  とし、 $M$  個のタスクで構成される訓練用のデータセットを  $D^{train} = \{T_m\}_{m=1}^M$  とする。また、 $D^{train}$  中の各タスクを、データ数  $A_k$  のサポートセット  $S_k = \{(\mathbf{x}_n, y_n)\}_{n=1}^{A_k}$  とデータ数  $N_k - A_k$  のクエリセット  $Q_k = \{(\mathbf{x}_n, y_n)\}_{n=A_k+1}^{N_k}$  に分割する。サポートセットは inner update に、クエリセットは outer update に用いる。また、パラメータ  $\theta$  を持つモデルを  $f_\theta$  としたとき、inner update では通常の教師あり学習で勾配法を用いてモデルを更新するのと同様に、以下の式 (1) で特定のタスクに最適化されたパラメータ  $\theta_{T_k}$  を獲得する。

$$\theta_{T_k} = \theta - \alpha \nabla_{\theta} L(f_{\theta}, S_k) \quad (1)$$

ここで、 $\alpha$  は inner update における学習率を表し、 $L$  は損失関数を表す。推薦システムにおける評価値予測では、損失関数として平均二乗誤差が用いられる。また、パラメータ  $\theta$  は outer update で獲得するモデルパラメータの初期値から学習が始め

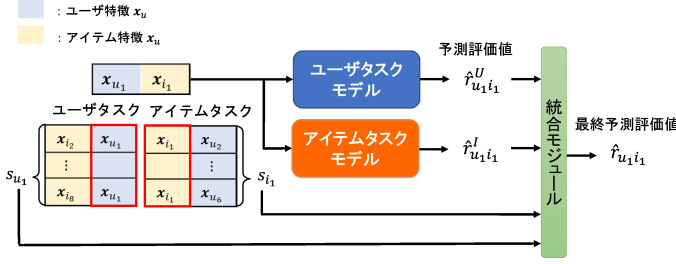


図 2: 提案手法の概要図

られる。Outer update では、複数のタスクのクエリセットに対して、損失が最小となるパラメータの初期値  $\theta^*$  を以下の式 (2) で獲得する。

$$\theta^* = \theta - \beta \nabla_{\theta} \sum_{T_k \sim p(T)} L(f_{\theta T_k}, Q_k) \quad (2)$$

ここで、 $\beta$  は outer update における学習率とし、 $p(T)$  は  $D^{train}$  におけるタスクの分布とする。以上のように式 (1) と式 (2) を各タスクに適用することで、様々なタスクで最適化しやすいモデルパラメータの初期値をメタ知識として獲得する。

メタ学習を最初に推薦システムに適用した研究として、Lee [7] らは、MAML [8] をユーザータスクに適用した Meta-Learned User Preference Estimator (MeLU) という手法を提案している。具体的には、MAML を用いてユーザーの平均的な嗜好を推薦モデルのパラメータ初期値として獲得する。これにより、ランダムな初期値と比較して少ない学習データでモデルのパーソナライズが可能となり、主にユーザーコールドの影響を緩和している。また、Dong ら [9] は、MeLU において全ユーザーに対して 1 つの初期値のみを獲得することを問題と捉えて、MeLU を拡張した手法を提案している。具体的には、まず、主にユーザーの属性情報から得たタスクの特徴を基に計算したタスク間の関連性を外部メモリに保存する。その後、推論時に外部メモリと推論対象のタスクの特徴を照らし合わせることで、タスクの特徴によって初期値のパラメータを変化させる機構を実現している。また、他の代表的なメタ学習手法を推薦システムに適用した研究もあり、Sankar ら [16] は、Prototypical Networks [17] をアイテムタスクに適用することでアイテムコールドの影響を緩和する手法を提案している。また、Lin ら [6] は、NP [15] をユーザータスクに適用することで、Dong ら [9] と同様の問題を解消している。

しかし、これらのメタ学習を用いた手法では、タスクに対するデータの分割方法がユーザー単位またはアイテム単位のいずれかであり、ユーザーコールドとアイテムコールドの影響を同時に緩和できていない。そこで本研究では、2 つのタスクに対してそれぞれメタ学習したモデルを組み合わせることで、この課題を解決する。

### 3 提案手法

#### 3.1 提案手法の概要

提案手法の概要を図 2 に示す。提案手法は、ユーザータスクモ

デルとアイテムタスクモデル、および、統合モジュールの 3 つから構成される。2 つのタスクモデルはそれぞれ事前にメタ学習済みのモデルであり、学習する際のタスクが異なる。ユーザータスクモデルはユーザータスク、アイテムタスクモデルはアイテムタスクでメタ学習する。また、統合モジュールは 2 つのタスクモデルの予測結果のどちらをどのくらい重視するかを調整するモジュールである。統合モジュールが 2 つのタスクモデルの特性を考慮して予測結果を調整することで、いずれのコールドスタート問題も緩和する。

次に、ユーザー  $u$  がアイテム  $i$  につける評価値  $r_{ui}$  を予測する際の入出力の流れを説明する。ここで、ユーザーの特徴を  $\mathbf{x}_u$ 、アイテムの特徴を  $\mathbf{x}_i$ 、ユーザータスクが含むインスタンス数（ユーザータスクサイズ）を  $s_u$ 、アイテムタスクが含むインスタンス数（アイテムタスクサイズ）を  $s_i$  とする。提案手法全体ではモデルに  $\mathbf{x}_u$ 、 $\mathbf{x}_i$ 、 $s_u$ 、 $s_i$  を入力し、予測評価値  $\hat{r}_{ui}$  を出力する。具体的には、ユーザータスクモデルには  $\mathbf{x}_u$  と  $\mathbf{x}_i$  を入力し、予測評価値  $\hat{r}_{ui}^U$  を得る。アイテムタスクモデルも同様に、 $\mathbf{x}_u$  と  $\mathbf{x}_i$  を入力し、 $\hat{r}_{ui}^I$  を得る。次に、 $s_u$ 、 $s_i$ 、および、2 つのタスクモデルから得た  $\hat{r}_{ui}^U$ 、 $\hat{r}_{ui}^I$  を統合モジュールの入力として  $\hat{r}_{ui}$  を得る。

また、提案手法では大きく 3 種類の学習をする。各タスクモデルに対するメタ学習とファインチューニング、統合モジュールに対する学習である。メタ学習で様々なタスクに適応しやすくなるようにモデルを訓練し、ファインチューニングで特定のタスクに適合する。統合モジュールに対する学習では、タスクサイズを用いてどちらのタスクモデルを重視するかを訓練する。

#### 3.2 2 つのタスクモデルの特性

2 つのタスクモデルは異なるタスクが学習対象となるため、それぞれ異なる特性を持つ。それらの特性の違いは、タスクが含むインスタンス数（タスクサイズ）による精度差の分布に、2 つの傾向として表れる。1 つは、一定のタスクサイズまでは、タスクサイズが大きいモデルの方がファインチューニングに用いる学習データが多いため、精度が良くなる傾向である。もう 1 つは、いずれかのタスクサイズが一定の値より大きい場合には、タスクサイズの大小関係に依らずに、いずれかのタスクモデルの方が精度が良くなる傾向である。これらの傾向を実際に予備実験により確認した。予備実験では、ユーザータスクモデルには MeLU、アイテムタスクモデルには MeLU をアイテムタスクに変更したモデルを用いた。また、精度には、映画のレビューデータセットである MovieLens-1M [18] における 5 段階評価の評価値予測の絶対平均誤差 (MAE: Mean Absolute Error) を用いた。図 3 に、結果として得られたタスクサイズごとのタスクモデル間の精度差を表すヒートマップを示す。ヒートマップは赤色が濃いほどアイテムタスクモデルの精度が良く、青色が濃いほどユーザータスクモデルの精度が良いことを表している。また、 $x$  軸と  $y$  軸はタスクサイズごとに範囲を分割し、軸目盛りにはタスクサイズの下限と上限、カッコ内に上限の分位割合を表記している。図 3 に示すように、橙三角内はアイテムタスクの方がタスクサイズが大きくファインチューニングに用いる学習データが多いため、アイテムタスクモデルの方が精度が良

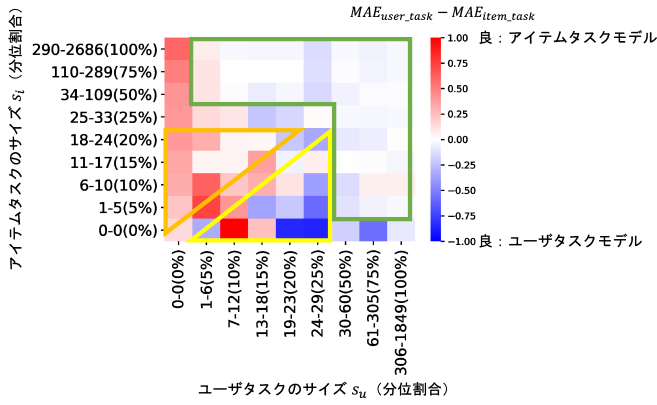


図 3: ユーザタスクモデルとアイテムタスクモデルの精度差

い傾向を確認できる。逆に、若干アイテムタスクモデルが優勢ではあるが黄三角内ではユーザタスクモデルの精度が良い傾向を確認できる。また、いずれかのタスクサイズがある程度大きい緑枠内では、タスクサイズの大小関係に依らず、ユーザタスクモデルの方が精度が良い傾向を確認できる。これらの特性を活かして 2 つのタスクモデルを統合することで、全体の精度を落とさずにユーザコールドとアイテムコールドの影響をどちらも軽減する。

### 3.3 タスクモデルの統合方法

統合モジュールでは、ユーザタスクサイズ  $s_u$  とアイテムタスクサイズ  $s_i$  からどちらのタスクモデルをどのくらい重視するかを表す重みを算出し、その重みを用いて 2 つの予測評価値  $\hat{r}_{ui}^U$  と  $\hat{r}_{ui}^I$  の加重平均をとることで最終的な予測評価値  $\hat{r}_{ui}$  を出力する。具体的には、パラメータを追加したシグモイド関数に  $s_u$  を入力し、ユーザタスクモデルの重視度を表す重み  $w_u$  を式 (3) のように得る。

$$w_u = c^U \cdot \text{Sigmoid}(a^U(s_u + b^U)) = \frac{c^U}{1 + e^{a^U(s_u + b^U)}} \quad (3)$$

同様に、アイテムタスクモデルの重視度を表す重み  $w_i$  は式 (4) のように得る。

$$w_i = c^I \cdot \text{Sigmoid}(a^I(s_i + b^I)) \quad (4)$$

式 (3), (4) 中の  $a$ ,  $b$ ,  $c$  を学習することで図 4 のような関数となることを想定している。このような想定をした理由は主に 3 つある。まず、一定のタスクサイズからはデータ量が増えても精度の変動が小さいため、一定の重みとなる必要がある。そのため、シグモイド関数を用いる。次に、3.2 節で述べた 2 つのタスクモデルの特性から、一定のタスクサイズまではタスクサイズが大きいタスクモデルを重視する必要があるため、その重視度合を  $a$ ,  $b$  で調整する。また、タスクサイズが一定の大きさからは片方のタスクモデルを重視する必要があるため、シグモイド関数の上限を  $c$  で調整する。最後に、算出した  $w_u$  と  $w_i$  を重みとして  $\hat{r}_{ui}^U$  と  $\hat{r}_{ui}^I$  の加重平均をとることで最終的な予測評

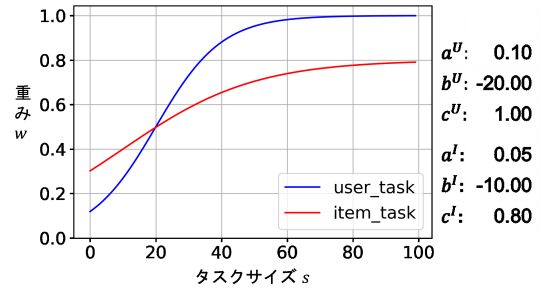


図 4: 改良したシグモイド関数の想定

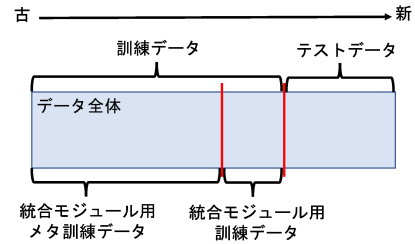


図 5: データセットの分割方法

価値  $\hat{r}_{ui}$  を式 (5) のように得る。

$$\hat{r}_{ui} = \frac{w_u \hat{r}_{ui}^U + w_i \hat{r}_{ui}^I}{w_u + w_i} \quad (5)$$

### 3.4 学習手順の詳細

提案手法では統合モジュールの過学習を避けるために、図 5 に示すように訓練データをさらに分割する。具体的には、統合モジュール用の訓練データと、タスクモデルが統合モジュール用訓練データを推論するためのメタ訓練用データに分割する。統合モジュールを訓練する手順をアルゴリズム 1 に示す。まず、統合モジュール用メタ訓練データを用いて、2 つのタスクモデルをメタ学習する。次に、メタ学習後のタスクモデルを用いて統合モジュール用訓練データに対して予測をする。最後に、その予測評価値を用いて統合モジュールを訓練する。最終的なテストデータに対しての推論には、訓練データでメタ学習したタスクモデルと学習済みの統合モジュールを用いる。

## 4 評価実験

### 4.1 実験設定

評価実験では、推薦の評価に用いられるベンチマークデータセットである MovieLens-1M [18] を使用した。MovieLnes-1M は映画のレビューデータセットであり、ユーザがアイテムにつけた 5 段階評価およびユーザとアイテムの属性情報が含まれる。ユーザの属性情報には性別、年齢、郵便番号、職業があり、アイテムの属性情報にはジャンル、監督、主演、レーティングがある。評価実験では、図 6 に示すように特定時刻で分割したのちに訓練データを削減する前処理を行う。具体的には、特定時刻で分割したユーザの内、訓練データ中にも評価履歴があるユーザの半数のデータを削減する。訓練データを削減する理



---

**Algorithm 1** 統合モジュールの学習
 

---

**Require:**  $D_{\text{integration\_meta\_train}}, D_{\text{integration\_train}}$

```

1: function META_TRAIN( $model, task\_name$ )
2:   for  $outer\_epoch$ 
3:     for  $task\_name \in D_{\text{integration\_meta\_train}}$ 
4:        $support\_set, query\_set \leftarrow \text{Split } task\_name$ 
5:       for  $inner\_epoch$ 
6:         Inner update  $model$  using  $support\_set$ 
7:       Outer update  $model$  using  $query\_set$ 
8:       Initialize  $model$  to updated initial parameters
9:   return  $model$ 
10: end function
11:
12: function TASK_MODEL_PREDICT( $model, \mathbf{x}_{\text{user}}, \mathbf{x}_{\text{item}}, task\_name$ )
13:   Get  $task\_name$  from  $D_{\text{integration\_meta\_train}}$  using
      $\mathbf{x}_{task\_name}$  as key
14:    $task\_size \leftarrow \text{Size of } task\_name$ 
15:   for  $inner\_epoch$ 
16:     Inner update  $model$  using  $task\_name$ 
17:    $\hat{r} \leftarrow model(\mathbf{x}_{\text{user}}, \mathbf{x}_{\text{item}})$ 
18:   Initialize  $model$ 
19:   return  $\hat{r}, task\_size$ 
20: end function
21:
22:  $task\_model^U \leftarrow \text{META\_TRAIN}(task\_model^U, \text{item})$ 
23:  $task\_model^I \leftarrow \text{META\_TRAIN}(task\_model^I, \text{item})$ 
24: for  $\mathbf{x}_u, \mathbf{x}_i, r_{ui} \in D_{\text{integration\_train}}$ 
25:    $\hat{r}_{ui}^U, s_u \leftarrow \text{TASK\_MODEL\_PREDICT}(task\_model^U, \text{user})$ 
26:    $\hat{r}_{ui}^I, s_i \leftarrow \text{TASK\_MODEL\_PREDICT}(task\_model^I, \text{item})$ 
27:    $\hat{r}_{ui} \leftarrow \text{integration\_module}(\hat{r}_{ui}^U, \hat{r}_{ui}^I, s_u, s_i)$ 
28:   update  $\text{integration\_module}$  using  $\hat{r}_{ui}, r_{ui}$ 

```

---

由は、特定時刻で分割した際に、大半が特定時刻をまたがないユーザ（Zero-Shot User）、および、極端に評価履歴が多いユーザ（Warm User）であり、評価に必要なコールドユーザを十分に確保できないためである。前処理として、訓練とテストのデータ数の比率が5対1となる時刻で分割し、削減処理を行ったデータセットは、総評価履歴数 981,217 個、総ユーザ数 6,040 人、総アイテム数 3,702 個となった。評価指標には MAE を用いる。ここで、 $l$  をテストデータ数、 $r$  を正解評価値、 $\hat{r}$  を予測評価値としたとき、MAE は式 (6) で定義できる。

$$MAE = \frac{1}{l} \sum_{j=1}^l |\hat{r}_j - r_j| \quad (6)$$

本実験では、メタ学習を用いた推薦システムの中で基礎となる手法として用いられることが多い MeLU [7] に提案手法を適用し、提案手法の有効性を評価した。評価方法としては、ユーザタスクモデル（UT: User Task model）と、アイテムタスクモデル（IT: Item Task model）、および、提案手法の予測精度を比較する。なお、UT は MeLU [7] と同一のモデルである。

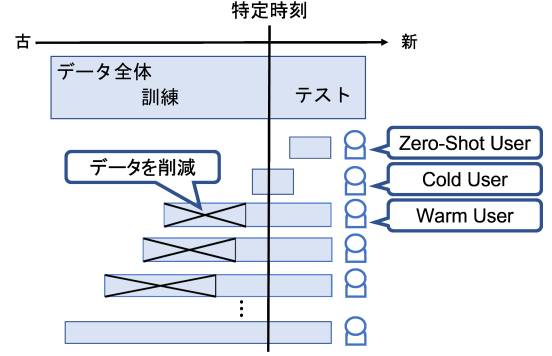


図 6: データセットの前処理

#### 4.2 実験結果と考察

実験結果を表 1 に示す。ここで、U-C は User Cold, U-W は User Warm, I-C は Item Cold, I-W は Item Warm の問題設定を表し、カッコ内はタスクサイズの上限の分位割合を表す。また、太字は各設定における最良の精度を表す。結果として、すべてのテストデータに対して提案手法が最良の精度となった。また、U-C (0%) 以外の設定において提案手法が最良となった。この結果から、提出手法はユーザコールドとアイテムコールドを同時に緩和できているため、MovieLens-1M [18] において提案手法により研究目的を達成できたことが確認できる。また、U-C の設定において、メタ学習の効果を発揮してほしい UT が IT より精度が悪い場合があった。これは、ユーザタスクサイズが極端に小さく、アイテムタスクサイズが十分に大きい訓練データに対しては、IT の方がファインチューニングに用いる学習データが多く、メタ学習の効果が学習データ数の優位性を上回らなかったためだと考えられる。また、I-C の設定において、IT が UT より精度が悪い場合があったことも、同様の理由だと考えられる。

次に、提案手法と従来手法 [7] の精度差を細かく比較し、精度に差が出た要因を考察する。図 7 に、提案手法とユーザタスクモデルの精度差の分布を図 3 と同様のヒートマップを用いて示す。ここでは、赤色が濃いほどユーザタスクモデルの精度が良く、青色が濃いほど提案手法の精度が良いことを表している。結果として、まず、橙三角の範囲において青色が濃い箇所が多いことから、提案手法が従来手法であるユーザタスクモデルより大幅に精度を改善していることがわかる。この範囲は提案手法によって改善することを想定していた範囲であり、大幅に改善したことからタスクサイズによる優位性を考慮した統合が成功したと考えられる。また、ユーザタスクサイズが 0 の範囲でも大幅に精度が改善していることが読み取れる。これは、元々ユーザタスクモデルが極端に低い精度であったため、統合モジュールによりアイテムタスクモデルの予測結果に多少でも引っ張られることで、精度が改善したと考えられる。次に、いずれかのタスクサイズがある程度大きい緑枠の範囲では、微量な精度改善となっていることがわかる。これは、タスクサイズが大きいユーザまたはアイテムは、タスクの分割方法に依らず学習時に頻出なデータであるため、タスクサイズによる優位性

表 1: MovieLens-1M [18] における MAE

設定 (分位割合) タスクサイズ	全体	U-C (0%)	U-C (10%)	U-C (25%)	U-W (75%)	I-C (0%)	I-C (10%)	I-C (25%)	I-W (75%)
		0	1~12	13~29	305~	0	1~10	11~33	286~
UT [7]	0.732	1.450	0.858	0.755	0.721	0.858	0.887	0.823	0.723
IT	0.761	<b>0.863</b>	0.816	0.796	0.749	0.914	0.860	0.837	0.757
提案手法	<b>0.715</b>	0.883	<b>0.790</b>	<b>0.746</b>	<b>0.712</b>	<b>0.827</b>	<b>0.856</b>	<b>0.795</b>	<b>0.709</b>

が小さくなり、タスクモデル間で精度の差が出ずに微量な改善にとどまったのだと考えられる。最後に、紫四角の範囲では一部精度が悪化している。これは、タスクサイズが中途半端であるため学習が困難となってしまったと考えられる。その一方で、同様にタスクサイズが中途半端な黄四角では精度が向上している。これは、紫四角内の訓練データ総数 38 個に比べて黄四角内の訓練データ総数は 265 個であり、紫四角と比べて十分に学習できたため、黄四角の範囲では精度が改善したと考えられる。

#### 4.3 統合モジュールの学習に関する考察

統合モジュールにおけるシグモイド関数の学習結果を想定したものと比較し、想定通りの部分の確認、および、想定と異なる部分の要因を考察する。まず、図 8 に学習後のシグモイド関数のグラフを示す。想定したシグモイド関数のグラフである図 4 と同様に、途中から 2 つのタスクモデル間の重視度が逆転していることが確認でき、概ね想定と同じ傾向であることがわかる。一方、想定より、 $b$  のタスク間の差が小さく、 $c$  のタスク間の差が大きい。このことは、タスクモデルの重視度の分布に違いが表れると考えられるため、次にシグモイド関数による重視度の分布を確認する。

想定したシグモイド関数による重視度の分布を図 9(a) に示し、学習後のシグモイド関数による重視度の分布を図 9(b) に示す。図 9 では、赤色が濃いほどアイテムタスクモデルを重視し、青色が濃いほどユーザタスクモデルを重視していることを表している。まず、図 9(a) から想定したシグモイド関数による傾向を確認する。

図 9(a) では主に 3 つの傾向が確認できる。1 つ目は、ヒートマップの左上または右下へ近づくにつれてタスクサイズの差が大きくなるため、タスクサイズが大きい側の重視度が大きくなる傾向である。次に、橙四角はいずれも小さいタスクサイズであるコールドの範囲を示しており、重視度の均衡が対角線からずれてアイテムタスクモデルが優勢となる傾向が確認できる。また、緑四角はいずれも大きいタスクサイズであるウォームの範囲を示しており、コールドの範囲とは逆転してユーザタスクモデルが優勢となる傾向が確認できる。

次に、図 9(a) と図 9(b) を比較する。まず、図 9(b) では、図 9(a) と同様に、ヒートマップの左上または右下へ近づくにつれてタスクサイズが大きい側の重視度が大きくなっていることがわかる。また、コールドとウォームのいずれの範囲でも想定と同じタスクモデルが優勢となっていることがわかる。しかし、コールドの範囲では想定よりアイテムタスクモデルが優勢でなく、ウォームの範囲では想定よりユーザタスクモデルが優勢で

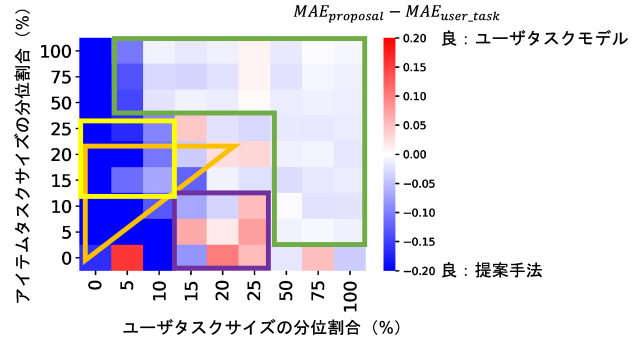


図 7: 提案手法とユーザタスクモデルの精度差

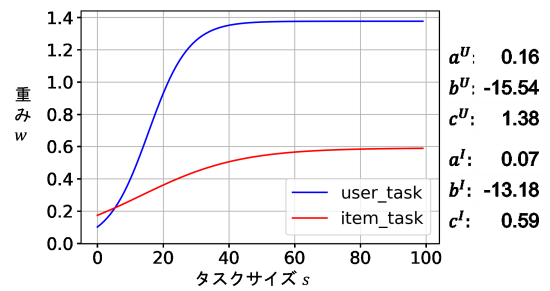


図 8: 学習後のシグモイド関数

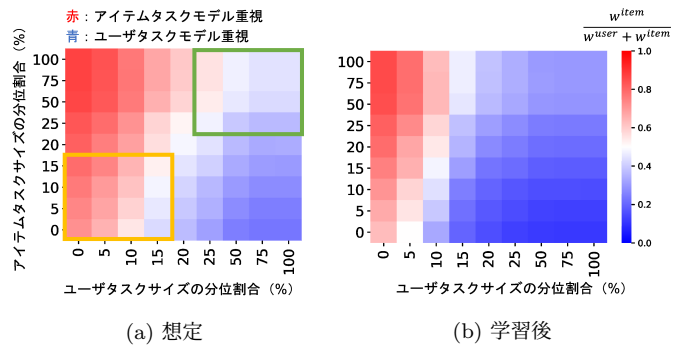


図 9: シグモイド関数の重視度の分布

あることがわかる。これは、訓練データの分布が想定と異なっているためだと考えられる。図 10 に、統合モジュール用訓練データにおける 2 つのタスクモデルの精度差の分布を示す。まず、図 10 におけるコールドの範囲の分布に関しては、図 9 の 2 つの分布のうち図 4 に示した想定したシグモイド関数に基づく図 9(a) の分布に近いことがわかる。一方、ウォームの範囲では、ユーザタスクモデルが想定より優勢であり、図 9(b) に示した学習したシグモイド関数による分布に近いことがわかる。

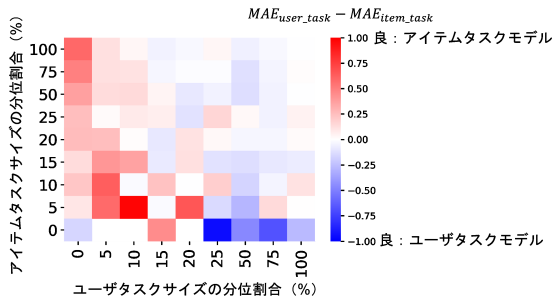


図 10: 訓練データにおける 2 つのタスクモデルの精度差

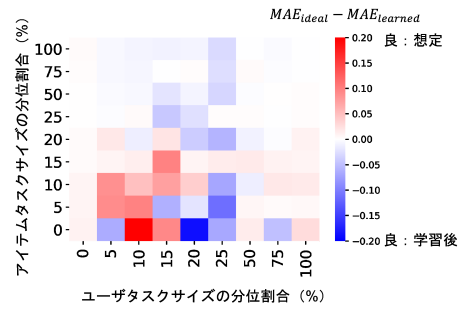


図 11: シグモイド関数の学習後と想定との精度差

これは、ウォームの範囲の訓練データ総数が 20,270 個、コールドの範囲の訓練データ総数が 203 個とデータ数に差があり、ウォームの範囲を重点的に学習したためだと考えられる。そのため、実際に学習したシグモイド関数において、想定の場合より  $b$  のタスク間の差が小さく、 $c$  のタスク間の差が大きい原因は、想定の場合と比べて、統合モジュールが訓練データにおけるウォームの範囲の分布をより再現するように学習したためだと考えられる。

想定と異なった部分の影響を確認するために、学習したシグモイド関数 (図 8) と想定したシグモイド関数 (図 4) をそれぞれ用いた場合の精度差分布を確認する。精度差の分布を図 11 に示す。赤色が濃いほど想定したシグモイド関数を用いた場合の精度が良く、青色が濃いほど学習したシグモイド関数を用いた場合の精度が良いことを表している。ウォームの範囲では、重点的に学習したシグモイド関数の方が精度が良く、逆に、コールドの範囲では想定したシグモイド関数の方が精度が良いことがわかる。これは、学習によって、コールドの範囲の重視度の分布がウォームの範囲の傾向に引っ張られたことにより、コールドの範囲の予測に悪影響を与えたためだと考えられる。この悪影響は、ウォームとコールドの訓練データ数の不均衡を解消することで改善が期待できる。そこで、コールドの訓練データをオーバーサンプリングして、ウォームの範囲のデータ数との不均衡を解消して、統合モジュールを学習した。結果として、不均衡のまま統合モジュールを学習させた場合と比べて、コールドの範囲の重視度の改善とともに精度を平均 0.060 改善することはできたが、想定の場合と比べると平均 0.012 悪い精度であった。そのため、この問題の改善には、単純に不均衡を解消するのみでは十分ではなく、他の方法でもウォームの学習の影響を小さくする必要があると考えられる。

## 5 結 論

本研究では、メタ学習を用いた推薦システムにおいてユーザーコールドとアイテムコールドの影響を同時に緩和することを目的とし、事前にメタ学習したユーザータスクモデルとアイテムタスクモデルを組み合わせる手法を提案した。評価実験から、MovieLens-1M [18] において MeLU [7] に提案手法を適用した場合に、両方のコールドの影響を緩和することができ、全体の精度も向上することを確認した。また、統合モジュールによ

り、タスクサイズを基にした 2 つのタスクモデルの特性を概ね想定通りに学習できていることを確認した。しかし、コールドの範囲では、想定よりアイテムタスクモデルの重視度が優勢でなく、悪い精度であった。

今後の課題として 3 点挙げる。1 つ目は、統合モジュールにおけるウォームの影響を小さくしたコールドの学習である。2 つ目は、2 つのタスクモデルの精度差の分布が異なる場合の提案手法の有効性の検証である。データセットが変われば、2 つのタスクモデルの特性とデータ数の分布が変わることが考えられるため、他のデータセットを用いた実験を通して提案手法の性能がデータセットに依存しないことを示す必要がある。3 つ目は、異なるメタ学習を用いた推薦システムに対する提案手法の有効性の検証である。本論文では MeLU [7] に提案手法を適用した場合の評価にとどまったが、提案手法は、ユーザータスクまたはアイテムタスクに分割する任意の手法に適用することができる。そのため、他の手法やユーザータスクモデルとアイテムタスクモデルで異なる手法に適用した場合を評価することで、メタ学習を用いた推薦システムにおけるアンサンブル手法の選択肢として確立できると考えている。

## 文 献

- [1] Panda, D. K. and Ray, S.: Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review, *Journal of Intelligent Information Systems*, Vol. 59, No. 2, pp. 341–366 (2022).
- [2] Tsai, C.-Y., Chiu, Y.-F. and Chen, Y.-J.: A two-stage neural network-based cold start item recommender, *Applied Sciences*, Vol. 11, No. 9, pp. 42–43 (2021).
- [3] Wang, X., Peng, Z., Wang, S., Yu, P. S., Fu, W., Xu, X. and Hong, X.: CDLFM: Cross-Domain Recommendation for Cold-Start Users via Latent Feature Mapping, *Knowledge and Information Systems*, Vol. 62, No. 5, pp. 1723–1750 (2020).
- [4] Chae, D., Kim, J., Chau, D. H. and Kim, S.: AR-CF: Augmenting Virtual Users and Items in Collaborative Filtering for Addressing Cold-Start Problems, *Proceedings of the 43rd International ACM conference on research and development in Information Retrieval (SIGIR)*, pp. 1251–1260 (2020).
- [5] Zhao, X., Ren, Y., Du, Y., Zhang, S. and Wang, N.: Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder, *Proceedings of the 45th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2595–

2600 (2022).

- [6] Lin, X., Wu, J., Zhou, C., Pan, S., Cao, Y. and Wang, B.: Task-adaptive Neural Process for User Cold-Start Recommendation, *Proceedings of the World Wide Web Conference (WWW)*, pp. 1306–1316 (2021).
- [7] Lee, H., Im, J., Jang, S., Cho, H. and Chung, S.: MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation, *Proceedings of the 25th ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1073–1082 (2019).
- [8] Finn, C., Abbeel, P. and Levine, S.: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Vol. 70, pp. 1126–1135 (2017).
- [9] Dong, M., Yuan, F., Yao, L., Xu, X. and Zhu, L.: MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation, *Proceeding of the 26th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 688–697 (2020).
- [10] 神島敏弘：推薦システムのアルゴリズム (2), 人工知能学会誌, Vol. 23, No. 1, pp. 89–103 (2008).
- [11] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.: Neural Collaborative Filtering, *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pp. 173–182 (2017).
- [12] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.: Extracting and composing robust features with denoising autoencoders, *Proceeding of the 25th International Conference on Machine Learning (ICML)*, Vol. 307, pp. 1096–1103 (2008).
- [13] Mirza, M. and Osindero, S.: Conditional Generative Adversarial Nets, arXiv preprint arXiv:1411.1784 (2014).
- [14] Hospedales, T. M., Antoniou, A., Micaelli, P. and Storkey, A. J.: Meta-Learning in Neural Networks: A Survey, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 44, No. 9, pp. 5149–5169 (2022).
- [15] Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J. and Eslami, S. M. A.: Conditional Neural Processes, *Proceedings of the 35th International Conference on Machine Learning (ICML)*, Vol. 80, pp. 1690–1699 (2018).
- [16] Sankar, A., Wang, J., Krishnan, A. and Sundaram, H.: ProtoCF: Prototypical Collaborative Filtering for Few-shot Recommendation, *Proceeding of the 5th ACM Conference on Recommender Systems (RecSys)*, pp. 166–175 (2021).
- [17] Snell, J., Swersky, K. and Zemel, R. S.: Prototypical Networks for Few-shot Learning, *Proceeding of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4077–4087 (2017).
- [18] Harper, F. M. and Konstan, J. A.: The MovieLens Datasets: History and Context, *ACM Transactions on interactive intelligent systems*, Vol. 5, No. 4 (2015).