

# 評価値推定タスクにおける推定評価値の評価に関する検討

田中 恒平<sup>†</sup> 小林 亜樹<sup>††</sup>

<sup>†</sup> 工学院大学工学研究科 電気・電子工学専攻 〒163-8677 東京都新宿区西新宿 1-24-2

<sup>††</sup> 工学院大学情報学部情報通信工学科 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: <sup>†</sup>cm16029@ns.kogakuin.ac.jp, <sup>††</sup>aki@cc.kogakuin.ac.jp

あらまし 推薦システムの評価には、一般的に RMSE や MAE といった統計的な指標で評価する 경우가多い。推薦システムは、推定評価値が高いアイテムを推薦する。このため、推薦システムの評価は、評価値が高いアイテムに対して正確に行えていけばよい。しかし、現在の評価方法は推薦される可能性が低いアイテムに対しても正確に評価を行ってしまい、利用シーンとの乖離が生じている。本稿では、評価値が高いアイテムのみを評価対象とし、評価値推定手法の推薦精度を比較する。評価値推定手法は、深層学習の枠組みを用いたオートエンコーダや Matrix Factorization といった手法とし、実データを用いて評価を行った。

キーワード 深層学習, 協調フィルタリング, 情報推薦, 評価値推定

## 1. はじめに

アイテムを購入する場合には、EC サイトを利用するという方法がある。EC サイトを利用してアイテムを購入する場合には、推薦システムによりアイテムが推薦されることがあり、これを購入の参考にする場合もある。

推薦システムがアイテムを推薦するためのアルゴリズムとして広く用いられているのが協調フィルタリングであり、商品の購買履歴や、評価履歴、EC サイト内のアイテム閲覧履歴などの情報から類似したアイテムやユーザを発見し、商品の推薦を行う。協調フィルタリングの中でも特に有効性が高いことで知られているアルゴリズムが Matrix Factorization であり、嗜好データの次元削減と欠損値が存在していた場合にも補完を行うことなく評価値の推定が行えることが特徴である。

これに対して、画像認識や音声認識といった分野で応用が進んでいる Deep Learning を情報推薦に利用しようとする動きがある。Deep Learning は機械学習の一手法であり、入力されたデータの特徴量を得ることが可能なオートエンコーダが、情報推薦分野の特に評価値推定タスクにおいて利用されている。

上述したような手法を利用して推薦システムの推定精度向上を目的とした研究は多く存在する [1] [2]。しかし、現在の RMSE などの統計的な指標を用いた推定精度の評価方法では、実際に推薦される可能性が高いアイテム、すなわち推薦システムにより高い評価値であると推定されたアイテムが、うまく推定できているかどうかは不明である。さらに、推薦システムを利用するユーザにとって、推薦される可能性が低いアイテムの評価は意味のないものであると考えられ、現在の評価方法は推薦システムの利用シーンと乖離した評価を行ってしまっているといえる。

そこで、本稿では、多段階の嗜好データセットを利用して、深層学習の枠組みを利用したオートエンコーダや Matrix Factorization の評価値別の推定精度を比較検討し、それぞれの手法の特徴について考察する。

本稿の構成は以下の通りである。第 2 章で協調フィルタリングの概要について言及し、関連研究を紹介する。第 3 章では、深層学習を用いた推薦の方法と、Matrix Factorization について説明する。第 4 章では実データを用いて実験を行い、結果について考察を行う。第 5 章では、本稿のまとめと、今後の課題について言及する。

## 2. 協調フィルタリング

協調フィルタリングは、推薦システムの利用者が閲覧したアイテムの履歴や、アイテムに付与した評価情報といった情報からアイテムを推薦する技術である。評価情報は、表 1 のような評価値行列により表現でき、この例はユーザがアイテムに対して 5 段階評価を行った例である。このとき評価値が書き込まれていない要素は、ユーザがアイテムを認知していない、あるいは購入したがアイテムに評価値を付与していないなどの理由で欠損値となっている。この欠損部分の評価値を推定することが評価値推定タスクの目的である。評価値を推定するためには、評価値行列の行要素同士あるいは列要素同士の類似度をピアソン相関係数やコサイン類似度といった指標で測る方法や、評価値行列の特徴量を最適化する回帰問題として取り扱う方法がある。

表 1 評価値行列

	アイテム 1	アイテム 2	アイテム 3	アイテム 4
ユーザ 1	1	4		2
ユーザ 2	2		3	2
ユーザ 3	5	4		
ユーザ 4			3	3

協調フィルタリングは、アイテムベース協調フィルタリングとユーザベース協調フィルタリングの 2 つに大別できる。

アイテムベース協調フィルタリングは、様々なユーザによって似たような評価値を付与されるアイテムは類似したアイテムであり、あるユーザが好むアイテムと類似したアイテムも好む

という仮定のもとアイテムの推薦を行う。アイテムベース協調フィルタリングは、表1のような評価値行列の列要素、すなわちアイテム同士の類似度を測りアイテムを推薦する。

一方、ユーザベース協調フィルタリングは、様々なアイテムに対して似たような評価をしているユーザ同士の嗜好が類似していると考え、嗜好が類似しているユーザであれば同様のアイテムを好むという仮定のもとアイテムの推薦を行う。ユーザベース協調フィルタリングは、評価値行列の行要素、すなわちユーザ同士の類似度を測りアイテムを推薦する。

情報推薦分野で用いる嗜好データは、評価値に偏りが生じていると言われている。評価値に偏りが存在したまま嗜好データの学習を行った場合には、学習に悪影響を及ぼす可能性がある。そのため、事前に評価値の偏りを排除することが望ましい[3]。例えば、図1で示す MovieLens-100k データセットでは、ユーザによってアイテムに低い評価値が付与される場合は少なく、評価値定義域の中央値よりも大きな値が付与される場合に偏っている。評価値の偏りに対しては、あるユーザ  $i$  の評価値  $r_i$  から、ユーザ  $i$  がアイテムに付与したすべての評価値の平均値  $\bar{r}_i$  をあらかじめ引いておくことにより、評価値の偏りによる影響を排除することができる。

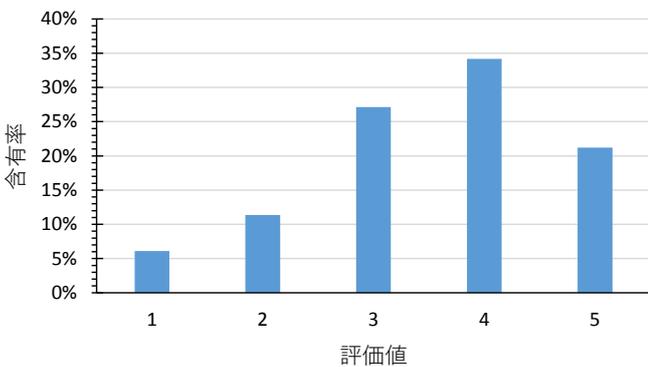


図1 MovieLens-100k 評価値分布

協調フィルタリングの応用として、商品の購買順序情報を考慮した推薦を高速かつ高精度に行うために、複数のマルコフモデルを最大エントロピー原理で統合する手法[4]や、共評価されているアイテム群をトピックとみなし、複数のトピックからアイテムを選定し推薦することで Novelty の高い推薦を可能する手法[5]などがある。

協調フィルタリングの手法の一つに Matrix Factorization がある。Matrix Factorization は評価値推定タスクにおいて特に有効性が高いことで知られている。Matrix Factorization の応用として、Web サイトのユーザビリティテストにおいて、ユーザが未閲覧の web ページの評価値を Matrix Factorization により推定する研究[6]や、グループ推薦において、メンバーの多様性を捉え、さらに欠損値にも対応するために、Matrix Factorization を非線形に拡張を行う[7]などの研究がある。

人工知能技術である深層学習も情報推薦分野に応用されている。ユーザがアイテムを閲覧するなどの時間的な特徴を捉えるためにリカレントニューラルネットワークを利用した研

究[8]や、欠損値を含む嗜好データをオートエンコーダで学習するための方法についての研究[9]などがある。

本稿では、深層学習の枠組みを用いたオートエンコーダや Matrix Factorization で評価値別の推定精度を比較し、それぞれの手法の特徴について考察を行うことが目的である。

### 3. 比較手法

#### 3.1 深層学習の枠組みを用いたオートエンコーダ

嗜好データの学習には DNN の枠組みであるオートエンコーダを用いる。オートエンコーダによる学習にはフォワードプロパゲーションとバックプロパゲーションの2つの段階があり、本節ではそれぞれの計算手順について述べる。さらに、嗜好データに欠損値が存在していた場合においても、欠損値であるという情報を入力可能にし、学習可能な手法である部分次元法についても言及する。

##### 3.1.1 フォワードプロパゲーション

フォワードプロパゲーションにおける目的は、評価値ベクトル  $\mathbf{r}$  を入力し、隠れ層の重み行列を用いて出力ベクトル  $\mathbf{y}$  を求めることである。

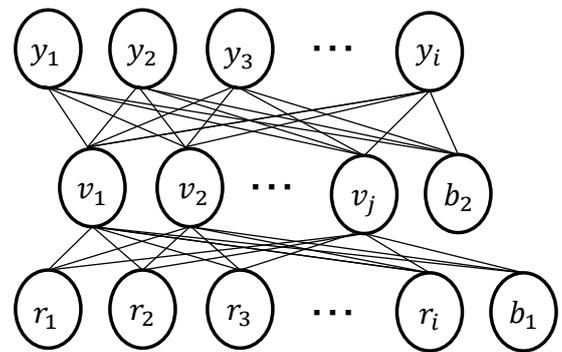


図2 オートエンコーダ

手順としては、まず隠れ層の出力ベクトル  $\mathbf{v}$  を求めるために、入力となる評価値ベクトル  $\mathbf{r}$  と入力層-隠れ層間 (エンコード) の重み行列  $\mathbf{W}$  との積を求める。

$$\mathbf{v} = f(\mathbf{W}\mathbf{r} + b_1) \quad (1)$$

ここで  $f$  は活性化関数であり、シグモイド関数や tanh といった関数がこれに該当する。  $b_1$  はバイアスユニットの値である。次に出力層における出力ベクトル  $\mathbf{y}$  を求めるために、隠れ層の出力ベクトル  $\mathbf{v}$  と隠れ層-出力層間 (デコード) の重み行列  $\mathbf{W}'$  との積を求める。

$$\mathbf{y} = f(\mathbf{W}'\mathbf{v} + b_2) \quad (2)$$

ここでデコードの重み行列  $\mathbf{W}'$  はエンコードの重み行列  $\mathbf{W}$  を転置したものである。

$$\mathbf{W}' = \mathbf{W}^T \quad (3)$$

$b_2$  はバイアスユニットの値である。

### 3.1.2 バックプロパゲーション

バックプロパゲーションにおける目的は、フォワードプロパゲーションにて得られた出力ベクトル  $\mathbf{y}$  と教師信号  $\mathbf{t}$  との差分ベクトル  $\mathbf{e}$  を求め、隠れ層の重み行列を更新することである。このとき教師信号  $\mathbf{t}$  は、学習の枠組みがオートエンコーダであるため、入力の評価値ベクトル  $\mathbf{r}$  となる。

バックプロパゲーションはまず差分ベクトル  $\mathbf{e}$  から、確率的勾配降下法などのアルゴリズムを用いて勾配ベクトル  $\mathbf{g}$  を求める。

次に勾配ベクトル  $\mathbf{g}$  と隠れ層の出力ベクトル  $\mathbf{v}$  との積  $\Delta \mathbf{W}'$  がデコードの重み行列  $\mathbf{W}'$  の更新量となる。

$$\Delta \mathbf{W}' = \mathbf{g}\mathbf{v} \quad (4)$$

$$\mathbf{W}'_+ = \Delta \mathbf{W}' \quad (5)$$

エンコードの重み行列は、入力の評価値ベクトル  $\mathbf{r}$  と、隠れ層の出力ベクトル  $\mathbf{v}$  との積  $\Delta \mathbf{W}$  で更新する。

$$\Delta \mathbf{W} = \mathbf{v}\mathbf{x} \quad (6)$$

$$\mathbf{W}_+ = \Delta \mathbf{W} \quad (7)$$

フォワードプロパゲーションとバックプロパゲーションは設定した学習回数分繰り返す。

### 3.1.3 積層オートエンコーダ

積層オートエンコーダは、隠れ層を多層にしたオートエンコーダである。通常のオートエンコーダでは、入力となるデータの特徴量が単層の隠れ層に蓄積され、入力データを低次元で表現することが可能であったが、隠れ層を多層にすることで、隠れ層に蓄積された特徴量をより低次元で表現することができる。これにより学習効果の向上が期待できる。

### 3.1.4 オートエンコーダの問題点

オートエンコーダは入力に欠損値の存在を許容することができない。嗜好データから欠損値を取り除く方法は2通りある。

1つが欠損値を含む行を丸ごと削除するリストワイズ法を適用する方法である。これは、あるユーザがデータベース中に存在するアイテムすべてに対して評価値を付与していなければ、そのユーザの嗜好データは学習に寄与しなくなることを意味する。嗜好データの欠損率は99%以上であることも珍しくはないため、欠損値を含む行を削除するリストワイズ法は適用することができない。

もう1つの方法が欠損値を適当な値で補完する方法である。欠損値を補完する手法として、ユーザがアイテムに付与した評価値の平均値や評価値定義域の中央値で補完するなどの簡易的な手法から、欠損値の一部はユーザが興味をもっていないアイテムであるために欠損していると考え、これらのアイテムの欠損値は0で補完を行う [10] などの方法がある。我々は嗜好データに存在する欠損値を上述した簡易的な手法で補完し、オートエンコーダへの入力する方法について実験を行い、その手法を適用した場合の推定精度については、以前報告をした [12]。そ

の結果、欠損値を補完し嗜好データの学習をオートエンコーダで行った場合には、欠損部分で補完値に近い値が推定され、これがユーザの嗜好を反映しているとは言い難い。そこで、次節で欠損値の補完を行わず、欠損値を欠損値として入力することが可能な部分次元法について述べる。

### 3.1.5 部分次元法

本節ではユーザが付与した評価値のみで学習が可能であり、オートエンコーダの枠組みを利用した手法である部分次元法について述べる。部分次元法は、以前我々が提案した手法 [11] であるが、[9] も同様の手法で嗜好データの学習を行っている。

学習時には、まず隠れ層の出力ベクトルを得るために、式 (1) のようにエンコードの重み行列  $\mathbf{W}$  と評価値ベクトル  $\mathbf{r}$  との線形和を求める。このとき、評価値ベクトル  $\mathbf{r}$  に0が含まれている次元が存在した場合、その次元の項は消失する。これを利用し、あるニューロンに欠損値であるという情報が入力されたならば入力値を0とすることで、欠損値であるという情報が入力されたニューロンと隠れ層間のエッジが切断される。これにより、出力値はユーザが付与した評価値のみ出力に影響を及ぼすようになる。

入力層の次元数は欠損値であるという情報が入力されたニューロンの数だけ減少するが、隠れ層の次元数については変化しない。このとき、隠れ層の次元数が入力層の次元数を超過してしまう場合には、ネットワークが恒等写像を学習してしまう。これにより、ユーザによってはネットワークの学習に寄与しなくなるため、隠れ層の次元数は入力層の次元数未満としなければならない。

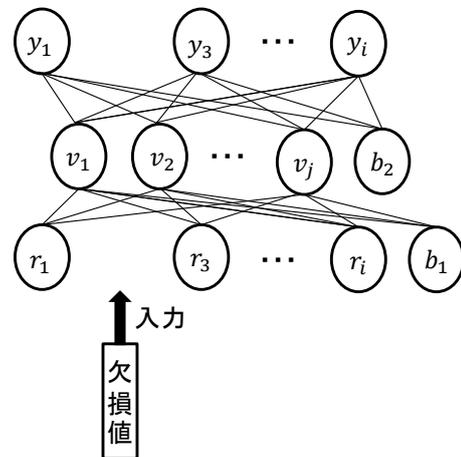


図3 部分次元法

さらにバックプロパゲーションでは欠損値を学習に寄与させないために、入力値が欠損値であった次元と対応する出力層の出力値を0とする。これにより欠損値であった次元の入出力間の差分が0になるため、対応した重み行列の要素が更新されず、欠損値が学習に寄与されることはない。

## 3.2 Matrix Factorization

Matrix Factorization は協調フィルタリングの一手法であり、評価値行列  $\mathbf{R}$  の次元を圧縮し、2つの行列  $\mathbf{P}$ ,  $\mathbf{Q}$  に分解し評価値を推定する技術である。例えばユーザが  $i$  人、アイテムが  $j$

個であった場合には、評価値行列  $R$  は  $i \times j$  の行列となる。このとき、 $P$  は  $i \times k$  の行列、 $Q$  は  $k \times j$  の行列となり、これらが評価値行列  $R$  を分解した行列となる。 $k$  は評価値行列  $R$  を  $k$  次元まで削減するという意味の値である。Matrix Factorization は、行列  $P$  と行列  $Q$  の積が、評価値行列  $R$  の近似となるように、行列  $P$  と行列  $Q$  の要素を最適化する。

$$R \simeq PQ \quad (8)$$

$$R \simeq \begin{pmatrix} p_{11} & \cdots & p_{1k} \\ p_{21} & \cdots & p_{2k} \\ \vdots & \ddots & \vdots \\ p_{i1} & \cdots & p_{ik} \end{pmatrix} \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{kj} \end{pmatrix} \quad (9)$$

行列  $P$  及び行列  $Q$  は、式 (10) の二乗誤差を最小化するように最適化を行う。

$$E = \sum (r_{ij} - \sum_{k=1}^k p_{ik}q_{kj})^2 \quad (10)$$

このとき、 $r_{ij}$  は嗜好データにもともと存在する評価値である。すなわち、Matrix Factorization は、評価値が存在する要素のみで学習し、評価値行列が欠損していたとしても無視して学習を行うことができる。

しかし、式 (10) の二乗誤差を最小化するのみでは、十分な汎化性能が得られない可能性があり、過学習を引き起こしてしまう場合がある。これに対し、式 (10) の二乗誤差式に正則化項を導入し、最適化を行う方法がある。

$$e_{ij}^2 = \sum (r_{ij} - \sum_{k=1}^k p_{ik}q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^k (\|P\|^2 + \|Q\|^2) \quad (11)$$

このとき、行列  $P$  及び行列  $Q$  の更新量は次の式となる。

$$p'_{ik} = p_{ik} + \alpha(2e_{ij}q_{kj} - \beta p_{ik}) \quad (12)$$

$$q'_{kj} = q_{kj} + \alpha(2e_{ij}p_{ik} - \beta q_{kj}) \quad (13)$$

本稿では、式 (12)、式 (13) の更新式を用いて嗜好データの学習を行う。

## 4. 実験

### 4.1 目的

推薦システムにより推薦されるアイテムは、推薦システムが高い評価値であると推定したアイテムとなる可能性が高い。しかし、現在の統計的な指標を用いた評価では、推薦されるアイテムが、うまく推定できているかどうかは不明である。さらに、推薦システムを利用するユーザにとって、推薦される可能性が低いアイテムの評価は意味のないものであると考えられ、現在の評価方法は推薦システムの利用シーンと乖離した評価を行っているといえる。

そこで、3章で紹介した部分次元法と Matrix Factorization について、評価値別の推定精度を比較検討し、それぞれの手法の特徴について考察する。

### 4.2 条件

実験で使用したデータセットは MovieLens-100k である。MovieLens-100k は、ユーザが映画に対して 5 段階評価を行った記録を収集したデータセットである。

表 2 使用データセット

	MovieLens-100K
ユーザ数	943
アイテム数	1682
評価値数	100000
欠損率	93.7%

推薦システムの精度を計るためにデータセットを 5 分割し交差検定を行う。5 分割した内、4 つを学習データ、1 つをテストデータとする。テストデータは学習データの評価値を隠したものであり、隠した評価値が推定評価値と近い値をとれば推定精度が高いといえる。

### 4.3 パラメータ設定

#### 4.3.1 部分次元法

部分次元法では、入力層、出力層、隠れ層 3 層の全 5 層で構成された積層オートエンコーダを用いる。隠れ層の 2、4 層目の次元数を 100 固定、3 層目の次元数は 20,60,100 の 3 パターンで実験を行った。学習アルゴリズムは確率的勾配降下法とし、バッチサイズは 1 のオンライン学習とした。学習回数は 1000 回とした。

#### 4.3.2 Matrix Factorization

Matrix Factorization では、評価値行列を何次元まで削減するかを決める  $k$  の値を 10、学習を進める幅を決める  $\alpha$  を 0.0005、 $\beta$  を 0.02 と設定した。学習回数は 1000 回とした。

### 4.4 評価

推定精度は RMSE で評価する。

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n (r_k - \hat{r}_k)^2} \quad (14)$$

$r_k$  はデータセットに存在する全評価値であり、 $\hat{r}_k$  は推定した評価値である。RMSE は推定精度の低さを示しており、値が小さいほうがより良い結果である。本実験では評価値別に評価を行うため、 $n$  は各評価値の個数となる。

### 4.5 実験結果

図 4 は、評価値のすべての段階で評価を行った場合についての結果であり、図 5 から図 9 は、評価値ごとに評価を行った結果である。グラフは、横軸が学習回数であり、縦軸が推定精度を示す RMSE である。RMSE は推定精度の低さを示すため、値が小さいほうが良い結果となる。

### 4.6 考察

#### 4.6.1 実験結果に関する考察

図 4 のすべての評価値で推定精度の評価を行う従来の方法に

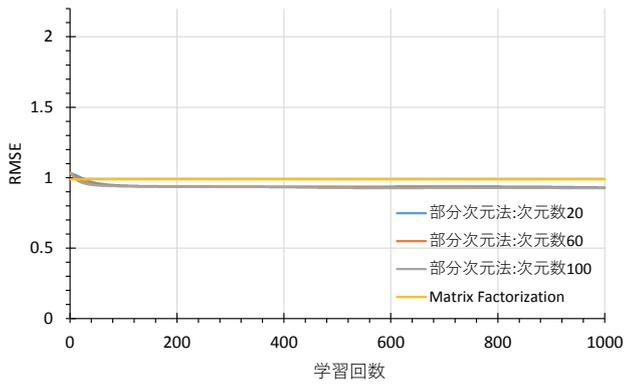


図 4 すべての評価値での評価

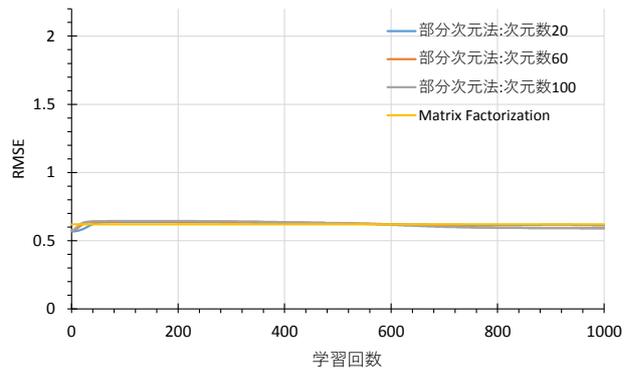


図 7 評価値 3 のみの評価

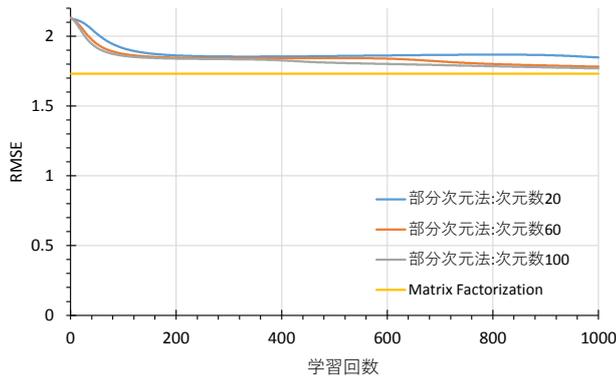


図 5 評価値 1 のみの評価

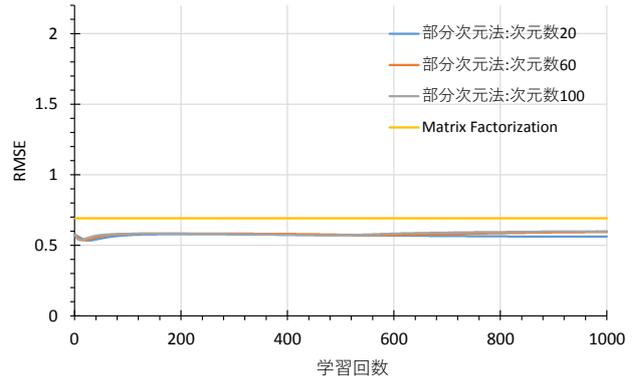


図 8 評価値 4 のみの評価

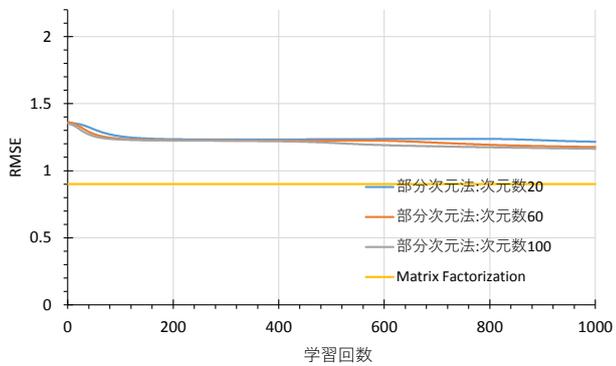


図 6 評価値 2 のみの評価

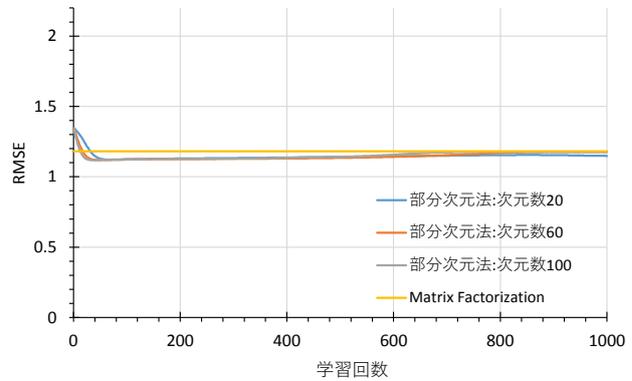


図 9 評価値 5 のみの評価

おいての結果は、部分次元法の RMSE が 0.92、MF の RMSE が 0.98 であった。これに対し、図 5 から図 9 の評価値別の結果は、評価値 3 と評価値 4 については RMSE が 0.5 前後と良い結果を示しているのに対し、他の評価値は RMSE が 1 以上の値をとっており悪い結果となっている。特に評価値 5 のアイテムは推薦システムにより推薦される可能性が高いアイテムであるにもかかわらず、RMSE が 1.2 程度となっており正確に推定できているとは言い難い。評価値 3 と評価値 4 の推定精度が良い理由としては、MovieLens-100k の評価値が評価値 3 と評価値 4 に偏っているためであり、モデルが評価値数の多い評価値に合わせて最適化を行ったためであると考えられる。

また、より正確に評価値を推定するためには、評価値行列を分解した 2 つの行列の初期化の方法を見直す必要があると考えられる。今回の実験では、評価値行列を分解した 2 つの行列の各要素は  $[0,1]$  の値域でランダムに初期化していたが、用いる嗜好データによって収束がはやく、かつ良い局所解が得られるような初期値を設定すべきであると考えられる。

#### 4.6.2 推定評価値の分布

実験では評価値行列の疎な部分について、Matrix Factorization や部分次元法で推定を行った。推定後の評価値行列の評価値別相対度数を表 3、表 4 で示した。表 3 は Matrix Factorization における評価値推定後の評価値別相対度数である。表 4 は隠れ

層の次元数を 60 と設定した部分次元法における評価値推定後の評価値別相対度数である。表の見方として、例えば、表 3 の右上の 0.0040 と入力されているセルは、評価値 5 と評価されるべき全アイテムの割合のうち 0.0040 が、評価値 5.5 から 6.0 の間の値と推定されていることを示している。

また、図 10 のグラフは部分次元法の相対度数から Matrix Factorization の相対度数を引いたものである。このグラフは、相対度数の差が正の値をとったならば、その値の分だけ部分次元法が Matrix Factorization よりも評価値数が多いことを示している。

ここでは、評価値 4.5 以上と評価されたアイテムが推薦されると仮定して考察を行う。まず、評価値 5 と推定されるべきアイテムに着目すると、評価値 5 と推定されるべきアイテムのうち、表 3 の Matrix Factorization では 0.1989、表 4 の部分次元法では 0.1049 のアイテムが推薦がされると考えられる。

図 10 から、評価値 3.0 から 4.0 の間の値と推定されたアイテムは部分次元法の評価値数が多く、評価値 4.0 から 4.5 の間の値と推定されたアイテムにおいても評価値 5 のみ、部分次元法の評価値数が多いことが見てとれる。これと図 1 から、部分次元法を用いた場合の推定評価値は、学習に用いた嗜好データの評価値分布に大きく依存するといえる。また、評価値 4.5 から 5.5 の間の値と推定されたアイテムに着目すると、Matrix Factorization は評価値 4 や評価値 5 は部分次元法よりも評価値数が多い。しかし、評価値 1 から評価値 3 が、評価値 4.5 以上と推定されている場合が部分次元法よりも多いことが見て取れる。これにより、Matrix Factorization は低評価値のアイテムを誤って推薦してしまう可能性が高いが、評価値 5 の本来推薦されるべきアイテムを部分次元法よりも高い割合で推薦することが可能な手法であるといえる。

また、実験結果では、Matrix Factorization よりも部分次元法のほうが RMSE の値は小さく、推定の誤差は少なかった。しかし、評価値 5 と推定されるべきアイテムが推薦される割合は Matrix Factorization のほうが高いという結果が得られた。このため、RMSE の値が小さければ良い推薦システムであるとは一概には言えない、という結果になった。

表 3 Matrix Factorization-相対度数

5.5~6.0	.0000	.0000	.0003	.0005	.0040
5.0~5.5	.0011	.0014	.0028	.0077	.0345
4.5~5.0	.0083	.0128	.0230	.0577	.1604
4.0~4.5	.0347	.0531	.0977	.2040	.3046
3.5~4.0	.0995	.1566	.2517	.3360	.2845
3.0~3.5	.1954	.2832	.3251	.2547	.1407
2.5~3.0	.2457	.2931	.2091	.1041	.0507
2.0~2.5	.2239	.1522	.0735	.0299	.0171
1.5~2.0	.1367	.0402	.0149	.0047	.0031
1.0~1.5	.0494	.0070	.0018	.0007	.0005
0.5~1.0	.0052	.0004	.0001	.0000	.0000
0~0.5	.0000	.0000	.0000	.0000	.0000
	1	2	3	4	5

表 4 部分次元法：次元数 60-相対度数

5.5~6.0	.0000	.0000	.0000	.0000	.0000
5.0~5.5	.0000	.0001	.0000	.0004	.0048
4.5~5.0	.0011	.0022	.0049	.0218	.1001
4.0~4.5	.0160	.0321	.0711	.1949	.3522
3.5~4.0	.0974	.1639	.2863	.4172	.3611
3.0~3.5	.2194	.3297	.3863	.2754	.1451
2.5~3.0	.2441	.2978	.1985	.0745	.0291
2.0~2.5	.1961	.1271	.0437	.0137	.0058
1.5~2.0	.1267	.0382	.0077	.0019	.0014
1.0~1.5	.0711	.0066	.0012	.0003	.0002
0.5~1.0	.0208	.0019	.0003	.0000	.0000
0~0.5	.0072	.0004	.0000	.0000	.0000
	1	2	3	4	5

## 5. おわりに

本稿では、部分次元法や Matrix Factorization の評価値別の推定精度を比較し、考察を行った。その結果、評価値数が多い評価値の段階にモデルが最適化していると考えられ、推薦システムにより推薦される可能性が高い評価値 5 のアイテムにおける推定精度は良くないということを示した。また、部分次元法は低評価値のアイテムを誤って推薦する可能性が低く、Matrix Factorization は部分次元法よりも多くの評価値 5 のアイテム、すなわち推薦されるべきアイテムを推薦することが可能な手法であると考えられる。深層学習を用いたオートエンコーダにおいて、隠れ層の次元数を決定する方法については今後の課題である。

## 謝 辞

本研究の一部は科研費（26242013）の助成を受けたものである。

## 文 献

- [1] Quanquan Gu, Jie Zhou, Chris Ding, “Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs” the SIAM International Conference on Data Mining, pp.199–210, 2010.
- [2] Sheng Li, Jaya Kawale, Yun Fu, “Deep Collaborative Filtering via Marginalized Denoising Auto-encoder” the 24th ACM International on Conference on Information and Knowledge Management, pp.811–820, 2015.
- [3] 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 23(1), 89-103, 2008-01-01.
- [4] 岩田具治, 山田武士, 土田修功, “購買順序を効率的に用いた協調フィルタリング” 情報処理学会論文誌: 数理モデルと応用 Vol.49 SIG4(TOM20), pp.125–134, 2008.
- [5] 小川祐樹, 諏訪博彦, 山本仁志, 岡田勇, 太田敏澄, “動的なトピック分類に基づく Novelty を考慮した推薦アルゴリズムの提案” 情報処理学会論文誌 Vol.50 No.6, pp.1636–1648, 2009.
- [6] 山田俊哉, 中道上, 松井知子, “行列因子分解による Web ユーザビリティ評価値の予測” 情報処理学会論文誌 Vol.56 No.1, pp.97-105, 2015.
- [7] 吉川友也, 岩田具治, 澤田宏, “グループ推薦のための非線形行列分解” 人工知能学会誌 Vol.30 No.2, pp.485–490, 2015.
- [8] Hanjun Dai, Yichen Wang, Rakshit Trivedi, Le Song, “Recurrent Coevolutionary Feature Embedding Processes for

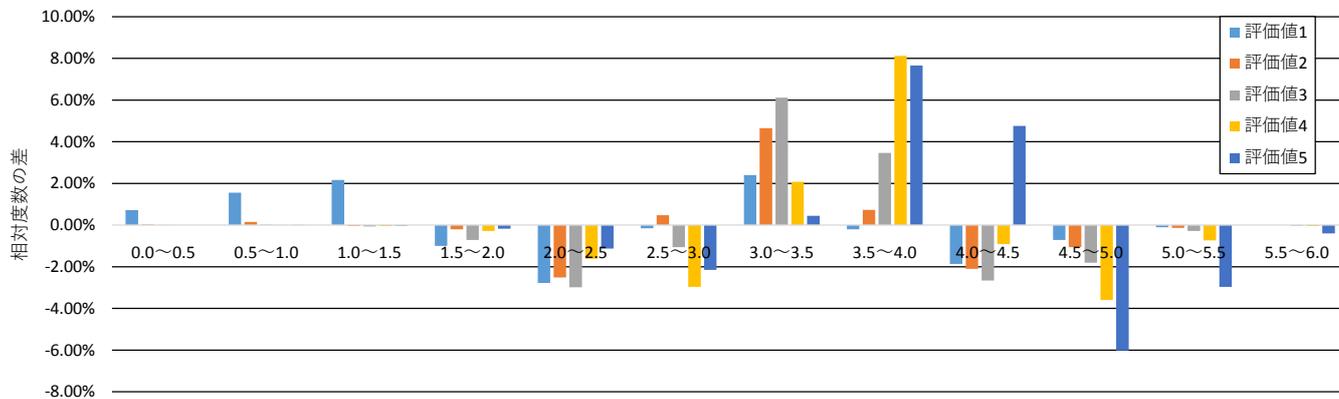


図 10 比較手法の相対度数の差分

Recommendation” In ACM Workshop on Deep Learning for Recommender Systems, pp.29-34, 2016.

- [9] Florian Strub, Jeremie Mary, Romaric Gaudel, “Hybrid Recommender System based on Autoencoders” In ACM workshop on Deep Learning for Recommender Systems, pp.11-16, 2016.
- [10] Won-Seok Hwang, Juan Parc, Sang-Wook Kim, Jongwuk Lee, Dongwon Lee, “”Told you I Didn’t Like It”: Exploiting Uninteresting Items for Effective Collaborative Filtering” Data Engineering(ICDE), 2016 IEEE 32nd International Conference, pp.349-360, 2016.
- [11] 田中恒平, 小林亜樹, “深層学習を用いた協調フィルタリングにおける欠損値の取り扱いに関する一検討”, 信学技報 Vol.116 No.214, pp.49-52, 2016.
- [12] 田中恒平, 小林亜樹, “深層学習を用いた情報推薦のための欠損値補完手法”, DEIM2016, C7-4, 2016.