インタラクティブプラットフォームにおける アイテム利用履歴に基づいた協調フィルタリング手法

周 彦† 牛尼 剛聡‡

†九州大学芸術工学府 〒815-8540 福岡県福岡市南区塩原 4-9-1
‡九州大学芸術工学研究院 〒815-8540 福岡県福岡市南区塩原 4-9-1
E-mail: †zhouyan1993ys@gmail.com, ‡ushiama@design.kyushu-u.ac.jp

あらまし 近年，オンラインゲームや音楽配信サービスにおいて，ユーザが継続的にコンテンツを利用するインタラクティブプラットフォームが増大している．インタラクティブプラットフォームでは，ユーザとアイテムのインタラクションデータ（利用履歴）が収集される．このデータは時間的な因子を含んでいるため，ユーザがシステムで行なったインタラクションの時系列である．しかし，従来の協調フィルタリング手法では，このような時系列データの特徴を利用して推薦を行うことが困難である．本研究では，ユーザの利用に関する時系列データを利用して，ユーザの状態を考慮した，新しい協調フィルタリング手法を提案する．この手法では，ユーザの利用に関する時系列パターンを分析することにより，ユーザの各アイテムに対する状態を理解する．そして，この状態と合わせて類似したユーザを探し出して，Top-N に基づいて推薦するアイテムを決定する．そして実験により，提案手法が従来の協調フィルタリング手法より，高い性能を有することを示す．

キーワード Top-N 推薦, 協調フィルタリング, インタラクティブプラットフォーム, 時系列

1. はじめに

近年，推薦システムは e コマースや SNS など幅広い分野で利用されている．推薦システムは，ユーザが莫大な候補の中からユーザに適したアイテムを発見することが困難であるという問題を効果的に解決した．協調フィルタリングは，現在最も利用されている推薦アルゴリズムの一つである．協調フィルタリングは多くのユーザのデータを分析し，嗜好の類似した他のユーザの情報を用いて，対象とするユーザの未知の嗜好を予測する手法である[1]．協調フィルタリングの手法は，ユーザベース協調フィルタリング[1]とアイテムベース協調フィルタリング[2]に大別できる．

一方，近年，オンラインゲームや音楽配信サービスにおいて，ユーザが継続的にコンテンツを利用するインタラクティブプラットフォームが増大している．インタラクティブプラットフォームでは，ユーザとアイテムのインタラクションデータ（利用履歴）が収集される．代表的なゲームプラットフォームの一つである Steam(注 1)では 2 週間ごとにユーザのプレイ状況を収集できる[3]．一方，音楽配信サイトの last.fm(注 2)では過去任意時期の再生回数を収集できる[4]．

時系列とは，連続的に（または一定間隔をおいて不連続に）観測して得られた一連の値のことである[5]．時系列データは時間因子を含んでいるため，インタラクションの時系列となる．時系列の変化パターンにはユーザの状態と嗜好に関する情報を含んでいると考えられる．しかし，従来の協調フィルタリング手法では，このようなユーザ操作の時系列に基づいて，その時のユーザの状態に基づいて適切な推薦を行うことが困難である．従来の代表的な推薦手法である協調フィルタリング手法[1,2]では時間因子を考慮していない．これまでに，時間因子を考慮した推薦手法[6-12]もいくつか提案されているが，このような時系列の変化パターンを考慮していない．

この問題を解決するため，本論文は，インタラクティブシステムに対するユーザのインタラクションの時系列特徴を考慮した協調フィルタリング手法を提案する.この手法では，一定期間の利用履歴を時系列とし，多項式を用いて時系列にカーブフィッティングを行い，最小二乗法により，多項式の最適関数を推定する．上記の多項式の微分を各時間にアイテムへの興味の状態とし,この状態により,ユーザ同士の類似度を計算し，各ユーザの近隣を探す．近隣のアイテムを優先順位でランキングし，最も上の*N*個のアイテムをユーザに推薦する．実験では，12 週間の last.fm のデータセットに基づいて，本研究の手法を従来の手法と比べ，提案手法が高い性能を持つことを示す．

本論文は，以下のように構成される．第 2 節で関連研究について述べる．第 3 節で提案手法について述べる．第 4 節で検証実験，第 5 節で実験の考察，第 6 節でまとめを述べる．

2. 関連研究

Baltrunas ら[6]は，時間に基づいたフィルタリング手法を提案した．この手法では，同じ期間に現れた評価データのみでモデルをトレーニングする．時系列から要素データを抽出し，同じ時刻のデータに基づいて，

(注 1): http://store.steampowered.com/
(注 2): http://www.last.fm/

ユーザ同士の類似度を計算する．この手法は時間因子を考慮するために，ベースラインと比べ精度が向上するが，アルゴリズムが複雑になる．

Gordea ら[7]と Campos ら[8]は，時間因子により評価データにペナルティ値を設定する手法を提案した．時間の距離はこのペナルティ値より大きい場合，評価データが無効になる．ペナルティ値がユーザとアイテムの属性と時間窓の選択より決められる．

Lee ら[9，10]は時間による重み関数$f_t(r)$を提案した．この関数では重みは評価の時間から決定されるものであり，アイテムの使用時間とは関係ない．Lee らはこの関数を用い，最近現れた暗黙の評価データを重み付け，明示的なデータに変換する．Lee らの手法は効果的に評価行列を補充する．

Zimdars ら[11]は時系列の処理方法を提案した．彼らは時系列を正規化したデータ集合を利用する．正規化したデータ集合とは，ユーザとアイテムのインタラクションの順番から作成されたデータ集合である．Zimdars らはこの正規化したデータ集合を利用して推薦を行う．

Ding ら[12]の研究では，ユーザの行動は，主に最近の興味に影響されることを論証し，推薦アルゴリズムには最近のデータの重みを増加するべきであると主張している．そのため，Ding らは協調フィルタリングで類似度を計算するとき，時間減衰関数を導入する．時間減衰関数は時間距離と関する逆関数である．

3. 提案手法

本研究はインタラクティブプラットフォームの提供した API を利用し，一定期間に連続する複数のユーザの利用履歴を収集する．この中に，アイテム集合 I，ユーザ集合 U，時間集合 T，アイテム$i \in$ I，ユーザ$u \in$ U，時間$t \in$ T，時間tにおけるユーザuのアイテムiに対する評価値を$r_{u,i}^t$とする．特に，時間集合 T は全順序集合，順序は時間の前後である．

3.1 時系列の最適化

本研究で注目するのはユーザの各アイテムに対する興味の変化である．しかし，ユーザがインタラクティブプラットフォームを使用する場合，ユーザの休暇や忙しさなど様々な現実条件の影響を受け，ユーザの利用状況は大きく変動する．この全体的な変動のため，各アイテムへの評価の変化の抽出が難しくなる．この全体的な変動を排除するため，本論文は k-順序平均法との時系列最適化手法を提案する．この最適化手法では，全てのユーザ$u \in$ U に対し，以下の処理をする．

まず，ユーザuの全ての評価値$r_{u,i}^t$を行列R_uとする．

$$R_u = \left[r_{u,i}^t\right]_{|\mathrm{I}|\times|\mathrm{T}|} \tag{1}$$

数式(1)により，同じ時間tにおける評価値$r_{u,i}^t$は行列R_uの列となり，同じアイテムiに対する評価値$r_{u,i}^t$は行列R_uの行となる．その中に，行列R_uの行ベクトルは時系列である．積み上げ縦棒グラフとして表示する場合，行列R_uは図 1 のようになる．

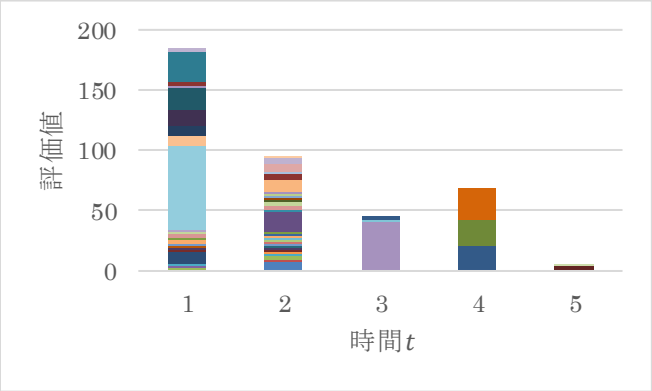

図 1: 行列R_uの例

R_uの列ベクトルを$\mathbf{c}_t(t \in T)$とする．処理後の行列はR'_u，R'_uの列ベクトルを$\mathbf{c'}_t(t \in [1,k])$とする．全ての$t \in [1,k]$に対し，$\mathbf{c'}_t$各要素の総和$\|\mathbf{c'}_t\|_1$は同値であり，ここから$V$とする．$V$は数式(2)により決定される．

$$V = \frac{\|R_u\|_1}{k} = \frac{\sum_{i\in \mathrm{I}}\sum_{t\in \mathrm{T}} r_{u,i}^t}{k} \tag{2}$$

k-順序平均法のプロセスは以下の様に示される．

まずフィルステップを説明する．前提として，ベクトル$\mathbf{c}_x$と$\mathbf{c'}_z$があるとするとき，$\mathbf{c'}_z$の足りない部分は$V - \|\mathbf{c'}_z\|_1$となる．$\|\mathbf{c}_x\|_1 > V - \|\mathbf{c'}_z\|_1$の場合，$\mathbf{c}_x$から引いて，$\mathbf{c'}_z$に加える比率を$p = (V - \|\mathbf{c'}_z\|_1)/\|\mathbf{c}_x\|_1$とすると，$\mathbf{c}_x$は$(1-p)\mathbf{c}_x$になり，$\mathbf{c'}_z$は$\mathbf{c'}_z + p\mathbf{c}_x$になる．これで，$\|\mathbf{c'}_z\|_1 = V$となり，$\mathbf{c}_x$と$\mathbf{c'}_{z+1}$は次のフィルステップの入力となる．$\|\mathbf{c}_x\|_1 < V - \|\mathbf{c'}_z\|_1$の場合，$\mathbf{c}_x$を全部$\mathbf{c'}_z$に加える．これで，$\|\mathbf{c'}_x\|_1 = 0$となり，$\mathbf{c}_{x+1}$と$\mathbf{c'}_z$は次のフィルステップの入力となる．

初期状態では，全ての$\mathbf{c'}_t$は零ベクトルである．まず$\mathbf{c}_1$と$\mathbf{c'}_1$を入力とし，フィルステップを行う．そして，フィルステップを繰り返し，最終に全ての$\|\mathbf{c'}_t\|_1 = V$となる．

以上のアルゴリズムの疑似コードは以下のように示される．

Algorithm: k-Orderly Average Regrouping

Input: $\mathbf{c}_t \in R_u$
Output: $\mathbf{c'}_t \in R'_u$

$norm \leftarrow \|R_u\|_1/k$
$\boldsymbol{\theta} \leftarrow \mathbf{0}$
$index \leftarrow 1$
foreach $\mathbf{c}_t \in R_u$ ***do***
  while $\|\boldsymbol{\theta}\|_1 + \|\mathbf{c}_t\|_1 > norm$ ***do***
    $proportion \leftarrow (norm - \|\boldsymbol{\theta}\|_1)/\|\mathbf{c}_t\|_1$
    $\mathbf{c'}_{index} \leftarrow \boldsymbol{\theta} + \mathbf{c}_t \cdot proportion$
    $\boldsymbol{\theta} \leftarrow \mathbf{0}$
    $index \leftarrow index + 1$
    $\mathbf{c}_t \leftarrow \mathbf{c}_t \cdot (1 - proportion)$
  end while
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{c}_t$
end for

k-順序平均法により，図 1 の処理結果($k = 5$)は図 2 のように示される．

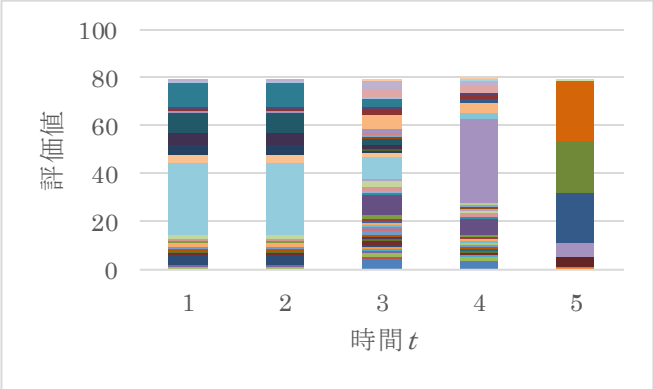

図 2: 行列$R_u{}'$の例

結果として，各時間$t \in$ Tにおける評価値の総和は同値となり，全体的な変動を排除するが．各アイテムに対する評価値の総和と評価の順序が不変である．そのため，各アイテムにの評価の変化が明確化される．数式(1)と同様に，処理後の行列$R_u{}'$と処理後の評価値$r'^{t}_{u,i}$は以下の数式を満たす．

$$R'_u = \left[r'^{t}_{u,i}\right]_{|\mathrm{I}|\times|\mathrm{T}|} \tag{3}$$

3.2 多項式カーブフィッティング

行列$R_u{}'$の行ベクトル$\mathbf{s}_{u,i}$は処理後の時系列である．ユーザのアイテムへの興味の状態を判断するため，ある方法でこの時系列を説明するのは必要である．本研究は多項式カーブフィッティングを使う[13]．即ち，数式(4)の多項式で時系列を説明する．

$$g(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_m x^m \tag{4}$$

mは多項式の次元，wは多項式のパラメータである．wを求めるため，ここに最小二乗法を使う．繰り返しwの値を調整により，数式(5)の誤差関数を最小化し，最適のwを決定する．

$$E(w) = \sum \left(g(x_j, w) - y_j\right)^2 \tag{5}$$

(x_j, y_j)は時系列$\mathbf{s}_{u,i}$の点のため，数式(3)により，以下の数式を満たす．

$$\begin{cases} x_j = t \\ y_j = r'^{t}_{u,i} \end{cases} \quad (t \in [1, k]) \tag{6}$$

これを数式(4)と(5)に代入すると，以下のようになる．

$$g_{u,i}(t, w) = w_0 + w_1 t + w_2 t^2 + \cdots + w_m t^m \tag{7}$$

$$E(w) = \sum_{t \in [1,k]} (g_{u,i}(t, w) - r'^{t}_{u,i})^2 \tag{8}$$

$g_{u,i}(t, w)$の導函数$g_{u,i}{}'(t, w)$を各時間の興味状態とすると，$g_{u,i}{}'(t, w) > 0$の場合，ユーザの興味が増えつつあり，$g_{u,i}{}'(t, w) < 0$の場合，ユーザの興味が失われつつあり，増減のスピードは$|g_{u,i}{}'(t, w)|$と見られる．$t = t_0$時間で推薦を行うと考えると，ユーザuのアイテムiに対する興味状態は$g_{u,i}{}'(t_0, w)$となる．多項式の次元$m = 2$とし，例を挙げると，時系列$\mathbf{s}_{u,i}$，フィット曲線$g_{u,i}(t, w)$と導函数$g_{u,i}{}'(t, w)$は図 3 のようになる．

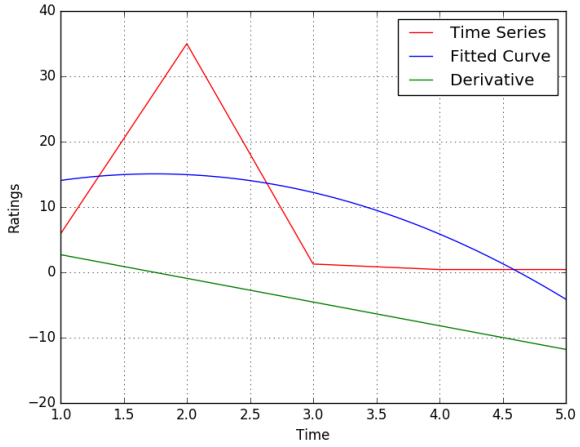

図 3: カーブフィッティングの例

3.3 興味状態に基づいた Top-N 推薦

本論文では，ユーザベース協調フィルタリングの代わりに，興味状態ベース協調フィルタリング手法を提案する．目標ユーザaの推薦時間t_0における興味状態と残りの全てのユーザ$b \in$ Uの全ての時間$t_b \in$ Tにおける興味状態の類似度を計算する．この類似度により，時間t_b前後にユーザbの評価したアイテムをユーザaに推薦する．以下に，具体的な手法を説明する．

まず，文献[12]の時間減衰関数を導入する．数式(9)は最も利用された時間減衰関数の一つである．

$$f(t, t_0) = \frac{1}{1 + \alpha|t - t_0|} \tag{9}$$

t_0は推薦を行う時間，t_uは評価データの時間，αは時間感度（定数）である．

本論文はコサイン類似度により，ユーザの興味状態の類似度を計算する．簡潔のため，以後t時間におけるユーザuのアイテムiに対する興味状態$g_{u,i}{}'(t,w)$を$g_{u,i}^{t}$とする．

ユーザ$a,b \in \mathrm{U}$，t_0時間におけるユーザaと$t_b \in \mathrm{T}$時間におけるユーザbの類似度を求める数式を以下に示す．

$$sim(a,b,t_b) = \frac{\sum_{i\in I}\left(g_{a,i}^{t_0} \times g_{b,i}^{t_b}\right)}{\sqrt{\sum_{i\in I}(g_{a,i}^{t_0})^2} \times \sqrt{\sum_{i\in I}(g_{b,i}^{t_b})^2}} \times f(t_b, t_0) \quad (10)$$

全てのアイテムはライフサイクルがあり，長い時間が経過すると，そのアイテムの推薦度が低くなる[7]．そのため，数式(10)はコサイン類似度に基づいて，時間の影響を考慮する．即ち，t_bとt_0の距離が長くなると，コサイン類似度の重みは低くなる．

時間t_bにおけるユーザbのアイテムiに対する評価は以下の数式で求める．

$$rat(b,t_b,i) = \frac{\sum_{t_u\in \mathrm{T}}\left(f(t_u,t_b) \times r'^{t_u}_{u,i}\right)}{\sum_{t_u\in \mathrm{T}} f(t_u,t_b)} \quad (11)$$

興味状態類似度に基づいて，各アイテムの推薦度は以下の式で計算する．

$$pri(a,i) = \frac{\sum_{b\in \mathrm{U}}\sum_{t_b\in \mathrm{T}}\left(sim(a,b,t_b) \times rat(b,t_b,i)\right)}{\sum_{b\in \mathrm{U}}\sum_{t_b\in \mathrm{T}} sim(a,b,t_b)} \quad (12)$$

数式(12)により，各アイテムの優先順位を計算し，目標ユーザの既に評価したアイテムを排除し，最も上のN個のアイテムを推薦結果とする．ここで，注意すべき点は，数式(12)においては，目標ユーザ$a \in \mathrm{U}$の類似状態は全てのユーザUの全ての時間Tにおける状態である．これは基本的な協調フィルタリングの思想である[1]．しかし，データが巨大になると，この思想による手法は容認できる以上の時間が掛かる．そのため，本論文では，類似度上位のS個の状態のみを類似状態とする．即ち実験においては，数式(12)では以下のようにする

$$\#(a,b,t_b) = S \quad (13)$$

ここで，$\#(a,b,t_b)$は(a,b,t_b)の数である．

4. 評価実験

4.1 データセット

本実験では，last.fm API[4]で収集した12週間のデータセット(2016年10月10日～2017年1月1日)を使用する．データセットを2週間ごとに区切り，6つの時間セグメントからなるデータセットとなる．データセットにはユーザのID，アーティストの名前，収集の期間とその期間に各アーティストを聞いた回数が含まれ，5,000ユーザの1,870,816件のデータがある．データセットの希少度は99.94%である．

4.2 評価方法

本実験は文献[14]の評価方法を利用する．Lathiaらは時間Tを決定し，時間T前のデータセットをトレーニングセット，残りのデータセットをテストセットとする．

本実験は，ユーザの利用履歴に基づいてアーティストを推薦すると想定し，推薦結果のヒット率で実験を評価する．具体的には，前5段(10週間)のデータセットをトレーニングセット，最新の1段(2週間)をテストセットとする．ユーザの各アーティストを聞いた回数をそのアーティストへの評価値とする．前処理は5-順序平均法を利用する．多項式カーブフィッティングにの次元$m=2$，$m=3$，$m=4$とし，それぞれに実験を行う．特に，多項式の次元$m=4$の場合，カーブフィッティングの理論的誤差は0である．数式(10)の時間感度$\alpha = 0.1$，数式(11)の時間感度α=1，近隣の状態数$S=100$とし，推薦のアイテム数Nを調整しながら，実験を繰り返す．

4.3 比較実験

比較実験は文献[12]により，時間減衰関数を導入した協調フィルタリングを利用する．数式(9)の時間減衰関数を考慮すると，ユーザuのアイテムiに対する評価は数式(14)となる．

$$\gamma_{u,i} = \sum_{t\in T} f(t,t_0) \times r_{u,i}^{t} \quad (14)$$

ユーザ同士の類似度とアイテムの推薦度は数式(15)と(16)となる．

$$sim(a,b) = \frac{\sum_{i\in I}\left(\gamma_{u,i} \times \gamma_{u,i}\right)}{\sqrt{\sum_{i\in I}{\gamma_{u,i}}^2} \times \sqrt{\sum_{i\in I}{\gamma_{u,i}}^2}} \quad (15)$$

$$pri_b(a,i) = \frac{\sum_{b\in \mathrm{U}} sim(a,b) \times \gamma_{b,i}}{\sum_{b\in \mathrm{U}} sim(a,b)} \quad (16)$$

提案手法の実験と同じレベルとなるため，比較実験の近隣ユーザを$S/k=20$とする．そして，明確なコントラストが表れるようにするため，数式(13)の時間感度$\alpha=0$と$\alpha=1$とし，二つの比較実験を行う．

4.4 実験結果

提案手法では，多項式カーブフィッティングの平均誤差$\overline{E(w)}$は24.69($m=2$)，9.22($m=3$)，0.00($m=4$)である．実験結果を表1と図4に示す．ヒット率は以下の様に計算する．

表 1:実験結果

		$N=10$		$N=20$		$N=40$	
		ヒット数	ヒット率	ヒット数	ヒット率	ヒット数	ヒット率
提案手法	$m=2$	1917	3.83%	3237	3.24%	5412	2.71%
	$m=3$	1827	3.65%	3163	3.16%	5389	2.70%
	$m=4$	1884	3.77%	3262	3.26%	5452	2.73%
比較手法	$\alpha=0$	312	0.62%	634	0.63%	1184	0.59%
	$\alpha=1$	330	0.66%	646	0.65%	1171	0.59%

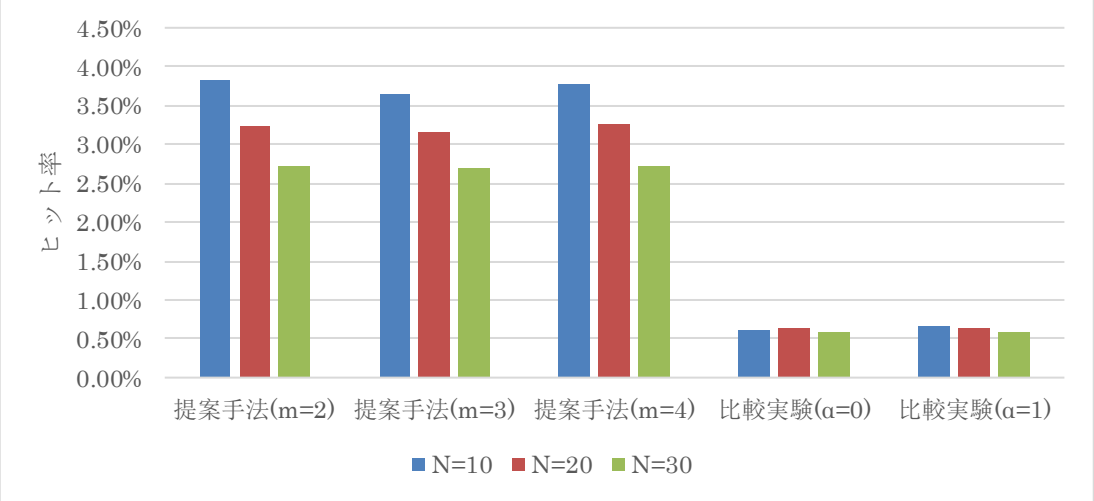

図 4: 実験結果

$$Hits\ \ Rate = \frac{\#Hits}{\#User \times N} \tag{17}$$

ここで，$\#Hits$はヒット数であり，$\#User$は実験対象の数である．

5. 考察

実験結果により，提案手法は，比較手法と比べて高いヒット率を持ち，高い性能を持つことを示された．NやSが低いほど，提案手法のヒット率は高くなるに対して，比較手法はあまり変化がない．そのため，提案手法の方がより適切なアイテムを選択できる．提案手法では，多項式の次元の増加に伴い，平均誤差が低下するが，ヒット率がほとんど変わらない．即ち，今回の実験では，二次関数で十分であると考えることができる．本手法は，推薦時刻に近いデータを重視するのに対して，比較手法では，時間感度α=0と比べ，α=1としても結果が良くならない．即ち，最近のデータより，データの変化と傾向の方がユーザの嗜好を反映すると考えられる．

6. おわりに

本論文では，従来の推薦手法はインタラクティブなプラットフォームにおける時系列データを適切に処理できないという問題を解決するために，新しい推薦手法を提案した．提案手法はカーブフィッティングを使い，時系列の特徴を説明する．その特徴に基づいたTop-N 推薦を行う．実験により，提案手法は従来の協調フィルタリング手法より高い性能を持つことを示した．

今後，本論文提案手法に利用した技術に対して，更に検討を行う必要がある．また，本論文で示した実験では，データセットが少なすぎ，更に実験を行い最適なパラメータを取得する必要がある．今後，他のデータセットに対しても，提案手法の有効性を評価するための実験を行う予定である．そして，今回比較した以外の時系列処理方法とTop-N 推薦モデルを検討する予定である．

参 考 文 献

[1] Resnick, Paul, et al. GroupLens: an open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994.

[2] Linden, Greg, Brent Smith, and Jeremy York. Amazon.com recommendations: item-to-item collaborative filtering. Internet Computing, IEEE 7.1(2003): 76-80.

[3] https://developer.valvesoftware.com/wiki/Steam_Web_API

[4] http://www.last.fm/api

[5] Michael Folk, et al. A First Course on Time Series

Analysis. GNU Free Documentation Licence.

[6] L. Baltrunas, X. Amatriain. Towards time-dependant recommendation based on implicit feedback. Workshop on context-aware recommender systems (CARS' 09), 2009: 1-5.

[7] S. Gordea, M. Zanker. Time filtering for better recommendations with small and sparse rating matrices. Proceedings of the 8th international conference on Web information systems engineering (WISE'07), 2007: 171-183.

[8] P. G. Campos, F. Diez, I. Cantador. Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. User modeling and user-adapted interaction, 2013: 1-53.

[9] T. Q. Lee, Y. Park, Y.-T. Park. A time-based approach to effective recommender systems using implicit feedback. Expert systems with applications, 2008, 34(4): 3055-3062.

[10] T. Q. Lee, Y. Park, Y.-T. Park. An empirical study on effectiveness of temporal information as implicit ratings. Expert systems with applications, 2009, 36(2): 1315-1321.

[11] A. Zimdars, D. M. Chickering, C. Meek. Using temporal data for making recommendations. Proceedings of the 17th conference on uncertainty in artificial intelligence, 2001: 580-588.

[12] Y. Ding, X. Li. Time weight collaborative filtering. Proceedings of the 14th ACM international conference on information and knowledge management, 2005: 485-492.

[13] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2007.

[14] N. Lathia, S. Hailes, L.Capra. Temporal collaborative filtering with adaptive neighborhoods. Proceeding of the 32nd international ACM SIGIR conference on research and development in information retrieval, 2009: 796-797.