

知識グラフを利用した協調フィルタリングにおけるユーザ間の類似度について

岸 嘉丸[†] 宮崎 純[†]

[†] 東京工業大学情報理工学院情報工学系 〒152-8552 東京都目黒区大岡山 2-12-1

E-mail: [†]kishi@lsc.cs.titech.ac.jp, ^{††}miyazaki@cs.titech.ac.jp

あらまし 本研究では、ユーザ評価と知識グラフを組み合わせたグラフを利用してユーザ間の類似度を計算する手法を提案し、評価を行う。推薦アルゴリズムの一つである協調フィルタリングは、ユーザの評価によって類似度を計算するため、アイテムの性質について考慮しない。そこで、協調フィルタリングに Wikidata から抽出した知識グラフのデータを組み合わせることで、アイテムの性質を考慮したユーザの嗜好の類似度を計算する手法を提案し、推薦精度の向上を目指す。

キーワード 情報推薦, 協調フィルタリング, 知識グラフ, オープンデータ

1. はじめに

近年、インターネットの普及により、ユーザがアクセスできる情報の量は膨大になっている。ユーザにとっては選択肢が増える一方で、どのニュースを読むか、どの動画を見るか、どの商品を購入するかなど、自分の興味に合わせた選択をするのは困難になっている。このような問題に対し、膨大なアイテムの中からユーザの嗜好に合わせたアイテムを提案するための情報推薦技術が開発されている [1]。

代表的な情報推薦アルゴリズムには、協調フィルタリングと内容ベース型推薦がある [4]。協調フィルタリングは、アイテムに対するユーザの評価をもとに推薦を行う。推薦対象のユーザと評価傾向が似ているユーザを特定し、推薦を行う。一方、内容ベース推薦は、アイテム自体の特徴を利用して推薦を行う。アイテムの特徴とは、映画であれば出演者、ジャンルといった知識である。推薦対象のユーザの好みを基に、似た特徴を持つアイテムを推薦する。協調フィルタリングは、評価値の類似によってのみ類似を計算するため、アイテムの質的な情報は考慮できない。そのため、似ているアイテムを評価していたとしても、類似度に反映できない。そこで、協調フィルタリングにアイテム知識を使用して精度向上を図る。

商品のアイテム知識はその商品の制作者によって提供されることが多い他、近年では、Wikidata^(注1)のようなオープンデータとして編集されており、利用することは容易である。

本研究では、ユーザの評価値データと Wikidata から抽出した知識グラフのデータを組み合わせることで、アイテムの性質を考慮したユーザ間類似度を計算する手法を提案する。MovieLens のデータセットを使用して、映画の協調フィルタリング推薦を行うことで精度を評価する。アイテム知識を使用することにより、評価しているアイテムの重なりだけでなく、出演者やジャンルの重なりからも類似性を測ることが期待できる。

2. 基本的事項

2.1 協調フィルタリング

協調フィルタリングは、アイテムに対するユーザの評価を基に推薦を行う手法である。推薦対象のユーザと、その他のユーザとの嗜好の類似度を計算する。この類似度をもとに、対象ユーザと似ているユーザ（ピアユーザと呼ばれる）を特定する。ピアユーザが好むアイテムは対象ユーザも好むという仮定のもと、推薦アイテムを提示する。

2.2 グラフ

グラフはノード（頂点）とエッジ（辺）によって構成される構造である。ノードとノードをエッジでつなぐことによって要素のつながりを表す。

エッジの方向を考慮するグラフを有向グラフ、しないものを無向グラフと呼び分ける。エッジごとに重みが設定されているグラフを重み付きグラフと呼ぶ。

2.3 グラフの行列表現

隣接行列は、グラフ上のノードが隣接しているかどうかを表す正方形行列である。無向グラフにおいては対称行列となる。隣接行列 \mathbf{A} をノード i とノード j をつなぐエッジの重み w_{ij} を使って次のように定義する。

$$[\mathbf{A}]_{ij} = a_{ij} = \begin{cases} w_{ij} & i \text{ と } j \text{ が接続している場合} \\ 0 & \text{それ以外の場合} \end{cases} \quad (1)$$

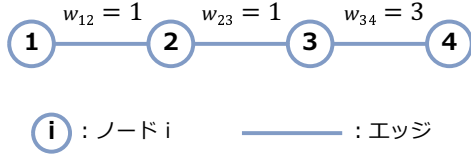
次数行列は、グラフの各ノードの次数を成分に持つ対角行列である。重み付きグラフにおいては、各エッジの重みを合計する。次数行列 \mathbf{D} を次のように定義する。

$$[\mathbf{D}]_{ij} = d_{ij} = a_i = \sum_{j=1}^n a_{ij} \quad (2)$$

ラプラシアン行列 \mathbf{L} は隣接行列 \mathbf{A} と次数行列 \mathbf{D} によって、 $\mathbf{L} = \mathbf{D} - \mathbf{A}$ で表される。グラフの接続を表した行列である。グラフとそのラプラシアン行列の例を図 1 に示す。

(注1) : https://www.wikidata.org/wiki/Wikidata:Main_Page

重み付き無向グラフの例



$$\text{隣接行列 } A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 3 & 0 \end{pmatrix}$$

$$\text{次数行列 } D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

$$\text{ラプラシアン行列 } L = D - A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 4 & -3 \\ 0 & 0 & -3 & 3 \end{pmatrix}$$

図 1 グラフとその行列表現の例

2.4 知識グラフ

知識グラフとは、様々な概念の関係を表す有向グラフである。知識グラフのノードをエンティティ、エッジを関係（リレーション）と呼ぶ。〈主語，述語，目的語〉の3つ組で構成され、これをトリプルと呼ぶ[7][11]。

2.5 ランダムウォーク

グラフ上のランダムウォークとは、あるノードを開始ノードとして、ノードにつながるエッジをランダムに辿っていくマルコフ連鎖モデルである。

マルコフ連鎖とは、遷移確率行列に基づいて状態遷移を行うモデルである[8]。現在の状態によってのみ次の状態が決定される。 t ステップ時点での状態（ノード）を $s(t)$ で表すとき、ノード $i = s(t)$ から、隣接するノード $j = s(t+1)$ に確率 $P(s(t+1) = j | s(t) = i) = \frac{a_{ij}}{a_i} = p_{ij}$ で移動する。つまり、重みの大きいエッジほど辿る確率が高くなる。

ランダムウォーカーが t ステップ時点で状態 i にいる確率を $\pi_i(t) = P(s(t) = i)$ で表す。さらに、 t ステップにおける確率分布ベクトルを $\pi(t) = [\pi_1(t), \pi_2(t), \dots, \pi_n(t)]^T$ で表す。また、遷移確率行列 P を $[P]_{ij} = p_{ij} = P(s(t+1) = j | s(t) = i)$ と定義する。 t での確率分布 $\pi(t) = \pi^0$ が決まれば、 $t+1$ での確率分布は $\pi(t+1) = P^T \pi(t)$ で計算できる。

3. 関連研究

グラフ上のランダムウォークに基づきノード間類似度を測る手法がこれまでに開発されている。

3.1 ランダムウォーク

対象のユーザノードを開始ノードとして、ランダムウォークを行う。そして、数ステップ先での各ノードへの確率分布を類

似度として使用する方法である。Clements らは、アイテム知識にあたる情報として、ユーザによってアイテムにつけられたタグを使用した。ユーザ・アイテム間、アイテム・タグ間、タグ・ユーザ間にそれぞれ遷移確率を定義し、一定の確率でノードを移動せずに自己遷移を行っている[3]。

3.2 Random walk with restart (RWR)

ランダムウォークを行う際、一定の確率で開始ノードに戻る。各ノードへの存在確率が収束するまで繰り返し、その定常状態確率を類似度とする手法である[10]。Shang らは、アイテムにつけられたタグに加え、ユーザプロフィール、ユーザ間のソーシャル関係を含めたグラフ上で RWR を適用した[9]。

3.3 average commute time $n(i, j)$

Fouss らはランダムウォークに基づき、グラフ上の距離指標を計算した[5]。

average first-passage time $m(k|i)$ は、ランダムウォークによってノード i をスタートしてノード k に初めてたどり着くまでのステップ数の平均値である。次の式を使うことで求めることができる。

$$m(k|i) = \begin{cases} 0 & (i = k) \\ 1 + \sum_{j=1}^n p_{ij} m(k|j) & (i \neq k) \end{cases} \quad (3)$$

average commute time $n(i, j)$ は $n(i, j) = m(j|i) + m(i|j)$ で計算できる。状態 i をスタートして j に到達して初めて i に帰ってくるまでのステップ数の平均値である。これは、距離の定義、(1) $n(i, j) \geq 0$, (2) $n(i, j) = 0 \Leftrightarrow i = j$, (3) $n(i, j) = n(j, i)$, (4) $n(i, j) \leq n(i, k) + n(k, j)$ を満たす。

グラフ上のノード間の距離を測る最も素朴な方法である最短経路 (shortest path) は、ノード間の経路の多さを考慮できない。それに対し、average commute time $n(i, j)$ はノード間の経路が多いほどランダムウォークによってたどり着く確率が上がるため、距離も小さくなる。

3.4 ラプラシアン行列の擬似逆行列 (L^+)

Fouss らはグラフのラプラシアン行列の擬似逆行列を利用した類似度指標を提案した[6]。評価値データから、重みなし2部グラフを作成し、このグラフのラプラシアン行列の擬似逆行列を類似度行列として使用する。

行列の擬似逆行列とは、逆行列の概念を正則でない行列や正方でない行列に拡張した概念である。ラプラシアン行列の擬似逆行列を L^+ で表し、次の式で求めることができる。

$$L^+ = \left(\frac{L - ee^T}{n} \right)^{-1} + \frac{ee^T}{n} \quad (4)$$

ここで、 n はグラフ上のノード数、 $e = [1, 1, \dots, 1]^T$ であり、 T は行列の転置を表す。

3.3 節で紹介した $n(i, j)$ は、 L^+ を使って以下のように表すことができる[6]。

$$\begin{aligned} n(i, j) &= V_G(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+) \\ &= V_G(e_i - e_j)^T L^+ (e_i - e_j) \end{aligned} \quad (5)$$

ここで、 $V_G = \sum_{k=1}^n d_{kk}$ 、 $e_i = [0, \dots, 0, 1, 0, \dots, 0]^T$ で

ある。

$\mathbf{x}_i = \mathbf{U}^T \mathbf{e}_i$, $\mathbf{x}'_i = \mathbf{\Lambda}^{1/2} \mathbf{x}_i$ を用いて変形する。 \mathbf{x}'_i を変換ノードベクトルと呼ぶ。 \mathbf{U} は \mathbf{L}^+ の固有ベクトルを、対応する固有値の降順に並べた行列、 $\mathbf{\Lambda}$ は固有値を対角に並べた行列である。

$$n(i, j) = V_G \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \quad (6)$$

つまり、変換ノードベクトルの距離は $\sqrt{n(i, j)}$ になっている。同様に、 l_{ij}^+ を変形すると、変換ノードベクトルの内積になっていることがわかる。

$$l_{ij}^+ = \mathbf{x}_i'^T \mathbf{x}_j' \quad (7)$$

よって、ラプラシアン行列の擬似逆行列の各要素は、距離が $\sqrt{n(i, j)}$ であるようなベクトル（変換ノードベクトル）の内積になっている。

4. 提案手法

本研究では、MovieLens のデータセットを対象として類似度計算を行う。提案手法の処理の流れを図2に示す。

入力として使用するデータは大きく二つである。まず、評価値データとして MovieLens のデータセットを取得する。加えて、Wikidata から映画に関連する知識グラフを取得する。

続いて、入力の二つのデータを組み合わせて一つの重み付き無向グラフを作成する。ユーザ・映画間には、評価値が存在する場合にエッジを張る。ここで、評価値の大小は考慮しない。知識グラフ上のエッジは、“different from”（混同されやすい別の物と違うことを表す）のリレーションを削除し、それ以外のエッジをそのまま使用する。知識グラフは本来有向グラフであるが、ここでは方向を考慮せず、無向グラフとして扱う。また、双方向にエッジがあったとしても、1本の無向エッジとする。知識グラフ上のエッジの重みを w_{KG} 、評価値のエッジの重みを w_{rating} とする。

このグラフのラプラシアン行列の擬似逆行列を計算し、これを類似度行列とする。

ラプラシアン行列の擬似逆行列は、内積に基づく指標であるため、距離に基づく指標に比べ、ノードの類似をより精度良く測ることができる [6]。

4.1 実装

知識グラフを含めると、グラフのノード数は数万ノードになる。ラプラシアン行列自体は疎行列（要素のほとんどが0であるような行列）であるため、メモリ上で扱うことができる。しかし、その擬似逆行列は密になるため、行列全体を一度に計算してしまうとメモリ上には展開できない。ただし、推薦に使用するのは対象ユーザとその他のユーザの類似度という1行である。そのため、次の式を使用し、1行ずつ計算していく。

$$\begin{cases} \mathbf{y}_i = \left(\mathbf{I} - \frac{\mathbf{e}\mathbf{e}^T}{n} \right) \mathbf{e}_i & (8) \\ \mathbf{L}\mathbf{l} = \mathbf{y}_i & (9) \\ \mathbf{l}_i^+ = \left(\mathbf{I} - \frac{\mathbf{e}\mathbf{e}^T}{n} \right) \mathbf{l} & (10) \end{cases}$$

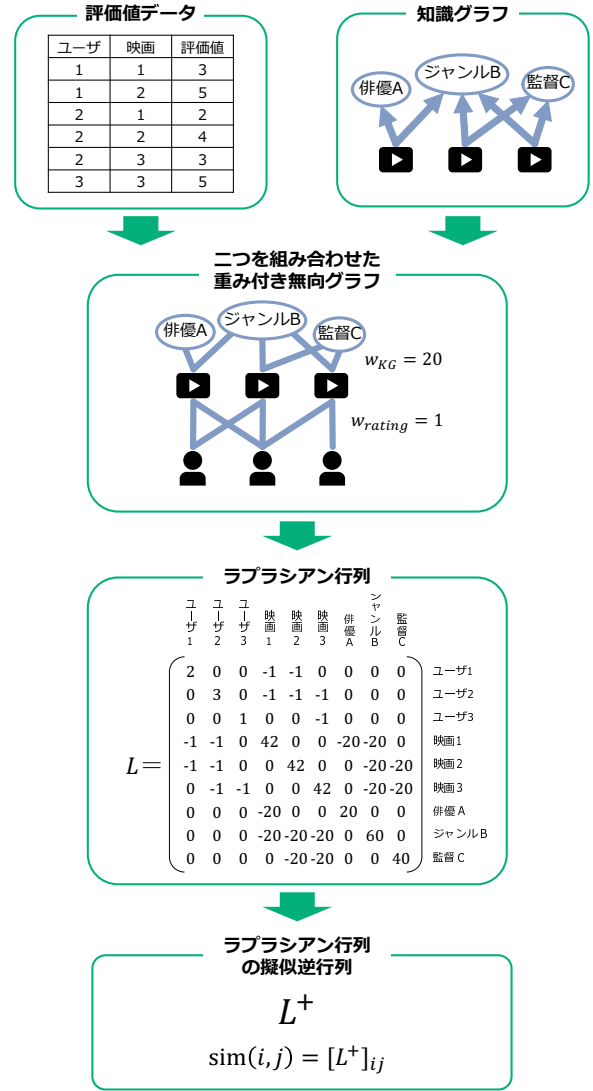


図2 提案手法の処理の流れ

ラプラシアン行列が大きいため、式 (9) を解く際に直接法で解析的に解くと時間がかかりすぎてしまう。そのため、共役勾配法を用いて近似的に求める。

4.2 予測評価値の計算

ユーザ p_0 が未評価のアイテム m_0 に対する予測評価値 $\text{pred}(p_0, m_0)$ を式 (11), (12) によって計算し、ユーザベース協調フィルタリングによる推薦を行う。ここで、ユーザ間の類似度 $\text{sim}(i, j) = [L^+]_{ij}$ を使用する。

$$\text{pred}(p_0, m_0) = \frac{\sum_{p=1}^k \text{sim}(p_0, p) a_{pm_0}}{\sum_{p=1}^k \text{sim}(p_0, p)} \quad (11)$$

$$a_{pm_0} = \begin{cases} 1 & \text{ユーザ } p \text{ が映画 } m_0 \text{ を見ている場合} \\ 0 & \text{見ていない場合} \end{cases} \quad (12)$$

この予測値の大きさと推薦順位を決定する。

5. 評価実験

5.1 実験環境

実験は、MATLAB R2019b でプログラムを作成して行った。

実験に使用したマシンの構成を以下に示す。

CPU Intel Core i5-4300M (2.60GHz, 2 コア)
RAM 16GB
HDD SAMSUNG MZMT128HMGR-00000 (128GB)
OS Windows 10 Pro 64bit

5.2 データセット

データセットは MovieLens^(注2) データセットを用いる。ここでは、1998 年に公開された MovieLens 100K Dataset を使用する。このデータセットに含まれるユーザ数は 943、アイテム数は 1,682、評価件数は 100,000 である。なお、ユーザ 1 人につき、最低 20 件の評価を行っている。

知識グラフは Wikidata から映画に関するエンティティのみを抽出して使用する。取得した知識グラフのエンティティ数は 516,287、エッジ数は 2,974,852 である。

なお、MovieLens 100K Dataset で評価対象となっている 1682 件の映画のうち、Wikidata のアイテムに対応付けられたのは 1660 件である。対応付けができた 1660 件の映画の評価のみを使用して実験を行う。このとき、全体の評価数は 99,476 件になる。

5.3 交差検証

MovieLens データセットに対し、10 分割交差検証を行った。まず評価値データ全体を 10 分割し、そのうち 1 つをテストデータ、残りの 9 つをトレーニングデータとして検証を行う。トレーニングデータを基に全ユーザへの推薦を行い、テストデータに含まれる映画を推薦結果として返すことができるかどうかを検証する。これを、分割したすべてのブロックに対してテストデータを変えて検証する。10 回の平均値を結果とする。

また、近傍ユーザ数は 20,40,60,80,100 と変更して実験を行った。

5.4 評価指標

推薦精度の評価として、次の 2 つの指標を使用する。

Recall (再現率) テストセットに含まれる映画のうち、上位 k 件の推薦に含まれていたアイテムの数の割合である。精度が良いほど、100 % に近い、高い値になる。ここでは上位 10 件の推薦を行った際の Recall スコアを使用する。

Percentile テストセットに含まれる映画の中央値が順位の中の位置を % で表した値。精度が良いほど、0 % に近い、小さい値になる。例えば、100 件の映画の順位を計算し、1 位、10 位、30 位にテストセットの映画が存在する場合、Percentile スコアは 10 % となる。

nDCG DCG (Discounted Cumulative Gain) を正規化することで求める評価指標である。DCG とは、適合度の高いアイテムがより高い順位にあるほど大きくなるような評価指標である。DCG, nDCG は次式で計算する。[2]

$$DCG = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (13)$$

$$nDCG = \frac{DCG}{iDCG} \quad (14)$$

表 1 ステップ数を変化させたときの上位 10 件での Recall スコアの比較 (%)

ステップ数	20	40	60	80	100	ノード数	エッジ数
すべて	23.59	24.17	24.08	23.80	23.55	516,287	2,974,852
3 ステップ	23.59	24.17	24.08	23.80	23.57	513,882	2,971,347
2 ステップ	23.61	24.19	24.08	23.81	23.58	301,578	1,679,471
1 ステップ	23.93	24.52	24.36	24.01	23.74	20,865	62,189

表 2 次数 1 のノードを削除したときの上位 10 件での Recall スコアの比較 (%)

	20	40	60	80	100	ノード数	エッジ数
削除なし	23.93	24.52	24.36	24.01	23.74	20,865	62,189
削除	24.30	24.87	24.76	24.52	24.26	10,176	51,500

$iDCG$ は、最も精度の高いランキングが生成できたときの DCG の値である。ここで、 rel_i は各アイテムの適合度を表すが、この実験ではその映画を見たことがあるかないかによって、0 または 1 とする。nDCG は 0 から 1 の値を取り、精度が良いほど 1 に近づく。

5.5 予備実験

提案手法では、使用する知識グラフの条件を変えて実験した。以降の節で他の手法との比較に使用する条件設定を決定するため、予備実験を行う。条件を変更した際の精度比較結果を示す。ここでは、上位 10 件を推薦したときの Recall スコアで比較する。

5.5.1 知識グラフのうち、使用するノードの変更

知識グラフのノードのうち、使用するノード数を変化させた。ここでは、MovieLens で評価対象となっている 1660 件のノードを起点として、何ステップで到達するかを基準に選択した。すべて使用、3 ステップ、2 ステップ、1 ステップと変化させた。なお、知識グラフを使用しない従来手法は 0 ステップにあたる。それぞれの条件で、近傍ユーザ数を 20,40,60,80,100 と変化させて推薦を行った。上位 10 件の Recall の結果を表 1 に示す。参考として、それぞれの場合のノード数とエッジ数を付記する。

1 ステップまでを使用すると最も精度が高い。

5.5.2 次数が 1 であるノードの削除

前の実験の結果から、評価対象の映画から 1 ステップのみを使用する。そこから、次数が 1 のノード（つまり、映画同士のつながりには関係していないノード）を削除する。結果を表 2 に示す。

次数が 1 のノードを削除すると精度が向上した。

5.5.3 重みの比率の変更

前の実験の結果から、知識グラフは 1 ステップまでのノードを使用し、次数が 1 のノードは削除する。知識グラフ上のエッジの重みと、評価データのエッジの重みの比率 $w_{KG} : w_{rating}$ を変化させる。結果を表 3 に示す。 $w_{KG} : w_{rating} = 20 : 1$ のとき最も良い結果を示した。

5.5.4 使用する実験設定

条件を変えた実験の結果から、比較では最も精度の高い結果を使用している。最終的な実験設定は以下の通りである。

(注2) : <https://grouplens.org/datasets/movielens/>

表 3 重みの比率を変化させたときの上位 10 件での Recall スコアの比較 (%)

$w_{KG} : w_{rating}$	20	40	60	80	100
30:1	25.33	25.59	25.55	25.29	25.02
20:1	25.23	25.69	25.61	25.46	25.29
10:1	24.95	25.55	25.37	25.17	25.05
1:1	24.30	24.87	24.76	24.52	24.26
1:10	23.87	24.46	24.38	24.09	23.84
1:20	23.81	24.44	24.32	24.00	23.76
1:30	23.84	24.40	24.30	23.98	23.73

- 知識グラフのうち、評価対象の映画から 1 ステップで到達できるノードのみ（つまり、評価対象の映画に関する知識のみ）使用する。
- 知識グラフのうち、次数が 1 であるノードを削除する。
- グラフ上のエッジの重みの比率を $w_{KG} : w_{rating} = 20 : 1$ に設定する。

5.6 実験 1 手法の比較

他の手法と比較し、ラプラシアン行列の擬似逆行列の類似度指標が妥当であるか検証する。

5.6.1 比較手法

ピアソン相関係数ユーザ a とユーザ b のピアソンの相関係数を類似度として、式 (15) によって求める。

$$sim(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}} \quad (15)$$

ここで、 P はアイテム集合、 $r_{a,p}$ はユーザ a のアイテム p に対する評価、 \bar{r}_a はユーザ a の評価値の平均を表す。

コサイン類似度ユーザ a とユーザ b のコサイン類似度を類似度として、式 (16) によって求める。

$$sim(a, b) = \frac{r_a \cdot r_b}{|r_a| \times |r_b|} \quad (16)$$

ここで、 r_a はユーザ a の評価値ベクトルである。

L^+ （知識グラフなし）Fouss らの手法 [6] に基づき、知識グラフを使用せずラプラシアン行列の擬似逆行列を求め、ユーザ a とユーザ b に対応する行列要素を類似度として使用する。

ランダムウォーク (RW) 提案手法と同じグラフ上で、対象ユーザのノードから開始してランダムウォークを行う。ただし、グラフのエッジの重み（＝ノード間の遷移確率）はランダムウォークにおいて最も高い精度の出る条件に変更する。また、ステップ数や自己遷移確率を変更し、最も高い精度の出る条件を使用する。条件を変えた実験の結果から、以下の実験設定を使用する。

- エッジの重みの比率を $w_{KG} : w_{rating} = 1 : 20$ に設定する。
- 自己遷移確率を 0 に設定する。
- ステップ数 2 での確率分布を使用する。

表 4 上位 10 件での Recall スコアの比較 (%)

	20	40	60	80	100
提案手法	25.23	25.69	25.61	25.46	25.29
ピアソン相関係数	20.16	20.97	21.24	21.27	21.01
コサイン類似度	24.51	24.92	24.64	24.41	24.09
L^+ （知識グラフなし）	24.66	25.29	25.16	24.92	24.70
RW	19.54	21.64	22.22	22.18	22.11
RWR	19.47	21.62	22.14	22.15	22.09

表 5 上位 20 件での Recall スコアの比較 (%)

	20	40	60	80	100
提案手法	36.49	37.16	37.07	36.84	36.54
ピアソン相関係数	30.27	31.28	31.34	31.17	30.96
コサイン類似度	35.87	36.10	35.80	35.38	35.11
L^+ （知識グラフなし）	35.46	36.43	36.45	36.05	35.61
RW	30.01	31.70	32.02	31.84	31.52
RWR	29.91	31.66	31.92	31.78	31.49

表 6 Percentile スコアの比較 (%)

	20	40	60	80	100
提案手法	4.723	4.168	4.050	4.042	4.071
ピアソン相関係数	6.191	5.601	5.542	5.581	5.654
コサイン類似度	4.433	4.205	4.211	4.263	4.322
L^+ （知識グラフなし）	4.697	4.181	4.091	4.090	4.118
RW	5.593	5.447	5.502	5.619	5.739
RWR	5.593	5.447	5.502	5.619	5.739

Random walk with restart (RWR) 提案手法と同じグラフ上で、対象ユーザのノードから開始して、RWR によって類似度を求める。ただし、エッジの重み（＝ノード間の遷移確率）は RWR において最も高い精度の出る条件に変更する。また、リスタート確率を変更し、最も高い精度の出る条件を使用する。条件を変えた実験の結果から、以下の実験設定を使用する。

- エッジの重みの比率を $w_{KG} : w_{rating} = 1 : 20$ に設定する。
- リスタート確率 0.9 に設定する。

5.6.2 実験結果

上位 10 件を推薦したときの Recall、上位 20 件の Recall、Percentile、nDCG の結果をそれぞれ表 4、5、6、7 に示す。表中で最も良い結果を太字で示している。提案手法は、知識グラフを使用しない従来の L^+ による手法よりも高い精度を示している。それに対し、ランダムウォークによる手法は知識グラフを使用しない L^+ の手法よりも低い結果を示した。提案手法と知識グラフなしの L^+ について t 検定を実施したところ、Recall(上位 10 件、20 件ともに)と nDCG では統計的に有意に ($p < 0.01$) 精度が向上していた。一方、Percentile は有意な差はなかった。

5.7 実験 2 評価値の大小の考慮

実験 1 では、ユーザがその映画を見たか見ていないかの 0,1 の値のみ使用して推薦を行った。しかし、実際の評価値データは 1~5 の 5 段階でつけられている。この評価値の大小をグラ

表 7 nDCG スコアの比較

	20	40	60	80	100
提案手法	0.5158	0.5237	0.5244	0.5226	0.5207
ピアソン相関係数	0.4676	0.4769	0.4788	0.4786	0.4772
コサイン類似度	0.5064	0.5141	0.5136	0.5119	0.5098
L^+ (知識グラフなし)	0.5087	0.5197	0.5196	0.5176	0.5146
RW	0.4558	0.4750	0.4815	0.4832	0.4827
RWR	0.4558	0.4750	0.4815	0.4832	0.4827

表 8 評価値を使用したときの精度比較

間隔 d	20	40	60	80	100
1	24.82	25.61	25.59	25.41	25.28
0.5	24.94	25.62	25.72	25.48	25.36
0.1	25.20	25.77	25.82	25.57	25.38
0.05	25.22	25.71	25.68	25.55	25.31

表 9 評価値とエッジの重みの対応

間隔 d	w_{KG}	w_{rating}				
1	60	1	2	3	4	5
0.5	40	1	1.5	2	2.5	3
0.1	24	1	1.1	1.2	1.3	1.4
0.05	22	1	1.05	1.1	1.15	1.2

表 10 上位 10 件での Recall スコアの比較 (%)

	20	40	60	80	100
評価値使用	25.20	25.77	25.82	25.57	25.38
0,1 の 2 値	25.23	25.69	25.61	25.46	25.29

フのエッジの重みに反映させることで、精度が向上できるか検証する。

5.7.1 実験設定の変更

評価値データをエッジの重みに割り当てる際、計算を行う。ユーザ a のアイテム p に対する評価値を $r_{a,p}$ として、 $w_{rating} = 1 + (r_{a,p} - 1) \times d$ で計算し、間隔 d を変更する。知識グラフ上のエッジの重みは、評価値の重みの中央値 $(1 + 2 \times d)$ を 1 として、20:1 になるように設定する。結果を表 8 に示す。参考として、実際に割り当たる重みを表 9 に示す。使用する評価値の間隔を 0.1 にしたとき最も精度が良かった。

以上の実験結果から、知識グラフ上のエッジの重みを $w_{KG} = 24$ に、評価値のエッジの重みを $w_{rating} = 1 + (r_{a,p} - 1) \times 0.1$ に設定する。

5.7.2 実験結果

上位 10 件を推薦したときの Recall、上位 20 件の Recall、Percentile、nDCG の結果をそれぞれ表 10、11、12、13 に示す。表中で最も良い結果を太字で示している。評価値の大小をエッジの重みに反映させた手法は、見たか見ていないかの 0,1 のみを使用する手法よりも高い精度を示している。t 検定を実施したところ、上位 20 件での Recall スコアは統計的に有意に ($p < 0.05$) 精度が向上していた。

6. ま と め

本研究では、協調フィルタリングの精度を向上させるため、

表 11 上位 20 件での Recall スコアの比較 (%)

	20	40	60	80	100
評価値使用	36.56	37.24	37.13	36.82	36.63
0,1 の 2 値	36.49	37.16	37.07	36.84	36.54

表 12 Percentile スコアの比較 (%)

	20	40	60	80	100
評価値使用	4.759	4.137	3.993	3.969	4.033
0,1 の 2 値	4.723	4.168	4.050	4.042	4.071

表 13 nDCG スコアの比較

	20	40	60	80	100
評価値使用	0.5156	0.5243	0.5247	0.5231	0.5212
0,1 の 2 値	0.5158	0.5237	0.5244	0.5226	0.5207

知識グラフを用いた類似度指標を提案し、その評価を行った。評価値データと知識グラフを組み合わせたグラフにおいて、ラブラシアン行列の擬似逆行列 L^+ を類似度とする手法によって、協調フィルタリングの精度向上を実現した。推薦タスクによる実験において、知識グラフを使用しない手法よりも精度が向上した。アイテム知識を使用することで、嗜好が類似しているユーザを精度良く検出することができることが明らかになった。また、評価値をグラフのエッジの重みに反映させることで、さらに精度を高めることができた。

知識グラフを使用することで、Recall スコアや nDCG スコアが統計的に有意に向上した一方で、Percentile スコアにおいては有意な向上は見られなかった。Recall や nDCG が向上していることから、これまで比較的上位にあったアイテムの順位をより上位にすることには成功しているが、適合アイテム全体を上位に推薦できてはいないものと推測される。

従来ではユーザが同一のアイテムを評価していないとユーザ間の類似を計算できなかったのに対し、知識グラフを使用することで、それぞれ別の類似するアイテムを評価している場合もユーザ間の類似度に反映できた。これにより、アイテム数が多かったり、一人あたりの評価件数が少なかったりして、同じアイテムを評価しているユーザが少ない状況（データセット）においても高い推薦精度を示すと予想される。

比較手法のうち知識グラフを使用している RW および RWR では、最良の結果を示したグラフの重みの設定が提案手法と異なっていた。提案手法では $w_{KG} : w_{rating} = 20 : 1$ (知識グラフのエッジの重みの方が大きい) であったのに対し、RW および RWR では $w_{KG} : w_{rating} = 1 : 20$ (評価値データのエッジの重みの方が大きい) であった。ランダムウォーク手法では、知識グラフのエッジをほとんど使用していないと言える。

評価値をエッジの重みに反映させる際の計算については課題がある。まず、この実験ではユーザごとに評価の甘さの違い (どんなアイテムも高く評価するユーザや、その逆のユーザの違い) を考慮していない。予めユーザの評価値の平均を引くことで、ユーザの評価の甘さを考慮して計算できる。また、今回は各評価値を等間隔に使用したが、必ずしもこれが最良かはわからない。エッジの重みの設定については、今後の課題となる。

謝 辞

本研究の一部は、JSPS 科研費 (18H03242, 18H03342, 19H01138A) の助成を受けたものである。

文 献

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734–749, June 2005.
- [2] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pp. 89–96, 2005.
- [3] Maarten Clements, Arjen P de Vries, and Marcel JT Reinders. Optimizing single term queries using a personalized markov random walk over the social graph. In *Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, pp. 18–24, 2008.
- [4] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, Markus Zanker. 情報推薦システム入門. 共立出版, 2012. , 田中 克己, 角谷 和俊 訳.
- [5] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering and subspace projection of the graph nodes. *Technical Report, IAG, Universite catholique de Louvain*, 2006.
- [6] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, Vol. 19, No. 3, pp. 355–369, 2007.
- [7] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [8] James R Norris and James Robert Norris. *Markov chains*. No. 2. Cambridge university press, 1998.
- [9] Shang Shang, Sanjeev R Kulkarni, Paul W Cuff, and Pan Hui. A randomwalk based model incorporating social information for recommendations. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6. IEEE, 2012.
- [10] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth International Conference on Data Mining (ICDM'06)*, pp. 613–622. IEEE, 2006.
- [11] 市瀬龍太郎, 趙麗花. 知識グラフ作成のための統合知識基盤の構築に向けて. 人工知能学会全国大会論文集 第 32 回全国大会 (2018), pp. 2F403–2F403. 一般社団法人 人工知能学会, 2018.