

プレイリスト生成における遷移確率を用いたスコアリング手法の提案

植田 聖司[†] 櫻 惇志^{††} 宮崎 純^{††}

[†] 東京工業大学総合理工学研究科知能システム科学専攻 〒152-8552 東京都目黒区大岡山2丁目12-1

^{††} 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丁目12-1

E-mail: †{ueda,keyaki}@lsc.cs.titech.ac.jp, ††miyazaki@cs.titech.ac.jp

あらまし 本稿では、プレイリストデータに付随する最も基本的な情報である楽曲、アーティスト、プレイリスト、ユーザの情報をグラフで表現し、遷移確率に基づき楽曲を推薦するプレイリスト生成手法を提案する。インターネット上で音楽再生のプラットフォームが出現するとともに、プレイリスト生成の技術が広く利用されるようになってきた。近年、音楽推薦の研究分野では、ユーザの再生した複数の楽曲からその次に再生するのに相応しい楽曲を推薦するプレイリスト生成が取り組まれている。既存手法は、線形和を用いて楽曲やアーティスト等複数の要素を統合していた。しかし、線形和の重み付けには予めその重み付けの最適値を計算する必要があるという欠点がある。そこで、本稿ではプレイリストの持つ複数の要素をグラフで表現し、グラフにおける遷移確率から推薦対象楽曲を決定する。これにより、要素間の重要度の比重を調整することなく、画一的に、複数の要素の情報を統合したプレイリスト生成手法を提案し、検証を行う。

キーワード プレイリスト生成, 音楽推薦, グラフモデル, ランダムウォーク, コンテキストualレコメンデーション

1. はじめに

spotify^(注1)に代表される音楽再生プラットフォームの利用が進み、ユーザはプラットフォーム内の楽曲の中から自分の好みに合った楽曲を集めプレイリストを生成するようになった。しかし、膨大な楽曲群の中から好みの楽曲を選ぶことはユーザにとってコストの高い作業であり、ユーザの好みに合わせて楽曲のリストを生成するプレイリスト自動生成技術の必要性が高まってきた。

近年、プレイリスト生成に関する研究として評価尺度 Hit rate [1] と呼ばれるタスクが取り組まれている。Hit rate は、プレイリストの一部の楽曲を、残りの楽曲及び他のプレイリスト群の情報を基に予想するタスクである。Hit rate は、単にユーザの好みに合致する楽曲を推薦するだけでなく、ユーザのコンテキスト (プレイリスト) に適切な楽曲を推薦する必要がある、より挑戦的かつ実用的な課題である。

Hit rate に関する研究では、同一のプレイリスト内で楽曲が共起する関係を用いた相関ルールベースの手法 [2] や、プレイリストやアーティスト同士の類似度等を考慮した手法 [1] [2] [3] [4] が提案されてきた。相関ルールを用いた手法は、同じプレイリストに含まれやすい楽曲を推薦し、アーティスト同士の類似性を考慮した手法は、関連度の高いアーティストの楽曲を推薦できる。しかし、これらの手法は、楽曲の共起やプレイリスト同士の類似度等、一つの要素のみに着目した生成手法である。ユーザのプレイリストの用途は多様であるため、ユーザの好む楽曲を推薦するには楽曲やプレイリストの持つ一つの要素だけでなく、複数の要素を取り入れたプレイリスト生成を行うことは、より有用であることが期待される。実際、著者らが過去に

提案した、楽曲の共起関係と楽曲とアーティストの関係を考慮したスコアリング手法 [5] において、それぞれ個別のスコアリング手法よりも線形和で統合することで推薦精度は向上した。しかしながら、線形和を用いたスコアリングでは、データセット毎の線形和の最適な重みの設定を必要とする欠点が存在する。この問題を解決するために、重み付けの設定を不要とする画一的なモデルが必要である。

これまでユーザやアイテムといった複数のコンテキスト情報を含む情報推薦システムでは、グラフモデルのフレームワークとしてランダムウォーク [6] [7] を用いた手法が存在する。なお、プレイリストに付随するコンテキスト情報として、プレイリスト内の楽曲、各楽曲に対応するアーティスト、プレイリストを作成したユーザが参照可能である。これらのプレイリストの基本情報をグラフモデルを用いて統合することで既存手法と比べてより高精度な推薦を可能に出来ると考えられる。

そこで、本稿では、プレイリストや楽曲、アーティスト、ユーザの情報から構成されるグラフに対してランダムウォークを適用する。その際、グラフ内のノード (コンテキスト情報) 間のエッジの重み (遷移確率) の設定方法と、各楽曲へ遷移する期待値を基にしたプレイリスト生成手法を提案する。

2. 関連研究

Hit rate において用いられる評価尺度^(注2)と、これまでに提案されてきたプレイリスト生成手法、コンテキスト情報におけるグラフモデルを用いた推薦手法について述べる。

(注1): <http://www.spotify.com>

(注2): 先行研究において、Hit rate はタスク及び評価尺度の両方に対して用いられるため、本稿においてもこれに倣うこととする。

2.1 評価尺度

2.1.1 Hit rate

プレイリスト自動生成の実験・評価方法として Hit rate [1] を用いた評価方法が提案されている。Hit rate は、楽曲数 n のプレイリストの一番最後の楽曲 t_n を隠し、残りの $n-1$ 曲 (t_1, t_2, \dots, t_{n-1}) の楽曲に関連すると考えられる k 件の楽曲のリストを生成し、その k 件の中に楽曲 t_n が含まれるかどうかを測る尺度である。

プレイリスト群 P に対して、推薦手法 R の精度を Hit rate で測る場合の計算方法について述べる。プレイリスト群 P 内の各プレイリストの最後の楽曲を正解楽曲 rel 、それ以外の楽曲を楽曲リスト L とする。 L に対してプレイリスト生成アルゴリズムを適用し、推薦リスト $R(L)$ を生成し、その $R(L)$ に正解楽曲 rel が含まれれば正解とし、正解率を計算する。計算式は式 (1) に示す。

$$HitRate(P) = \frac{1}{||P||} \sum_{(L, rel) \in P} 1_{R(L)}(rel) \quad (1)$$

ここで、本稿では推薦リストの長さ k は、 $k = 100$ と設定し、 $1_{R(L)}(rel)$ は式 (2) で計算される。

$$1_{R(L)}(rel) = \begin{cases} 1 & (rel \in R(L)) \\ 0 & (rel \notin R(L)) \end{cases} \quad (2)$$

2.1.2 MRR(mean reciprocal rank)

MRR はリスト内で適合する要素の順位を比較するための評価尺度である [8]。プレイリスト群 P の各プレイリストにおける、推薦リスト内の正解楽曲の順位である reciprocal rank (RR) の逆数の平均値が MRR である。一つの L 当たりの RR の計算式を式 (3) に示す。

$$RR(L, rel) = \begin{cases} \frac{1}{rank_{R(L)} rel} & (rel \in R(L)) \\ 0 & (rel \notin R(L)) \end{cases} \quad (3)$$

ここで、 $rank_{R(L)} rel$ は推薦リスト $R(L)$ における楽曲 rel の順位である。最終的な MRR の計算式は以下の式 (4) の通りである。

$$MRR(P) = \frac{1}{||P||} \sum_{(L, rel) \in P} RR(L, rel) \quad (4)$$

2.2 既存のプレイリスト生成手法

これまでの Hit rate に関する既存研究において楽曲の popularity [2] [3] やプレイリスト内の楽曲の頻出パターン [2]、プレイリスト同士の類似度を基にした近傍推薦手法 [1] などが提案されてきた。これらの手法と線形結合を用いたモデル [5]、情報推薦で多用される協調フィルタリングを用いた手法 [9] について述べる。なお、推薦リスト生成のタスク設定ではプレイリスト群を参照可能である。以後このプレイリスト群を学習データと呼ぶ。

2.2.1 popularity ベース

Same artists - greatest hits(SAGH)

ユーザが再生した楽曲リスト L に出現するアーティストをユーザが好むと考え、これらのユーザの楽曲を人気度順に並べて推薦する手法である。特定のアーティストの楽曲しか推薦しないため、推薦する楽曲が偏ってしまうが、ユーザからの信頼性を高められることが報告されている [3]。楽曲リスト L を与えた時、楽曲 t の SAGH のスコアは式 (5) で計算される。

$$score_{SAGH}(t, L) = count(t) \cdot 1_L(a) \quad (5)$$

ここで、 $count(t)$ は学習データにおける楽曲 t の出現回数を表し、 $1_L(a)$ は楽曲 t のアーティスト a が楽曲リスト L に含まれていれば 1 を返す。

Collocated artists - greatest hits(CAGH)

CAGH [2] は SAGH を拡張した手法であり、 L と類似度の高いアーティストの楽曲も推薦候補に加える手法である。同じプレイリストに出現するアーティスト同士は関連性が有ると仮定し、プレイリスト群内のアーティストの共起度からアーティストの類似性を計算している。アーティスト a と b の類似度は学習データを用いて以下の式 (6) で計算される。

$$sim_a(a, b) = \frac{\sum_p (1_p(a) \cdot 1_p(b))}{\sqrt{\sum_p 1_p(a) \cdot \sum_p 1_p(b)}} \quad (6)$$

ただし、 $1_p(a)$ はプレイリスト p にアーティスト a が出現すれば 1 を返し、そうでなければ 0 を返す。楽曲 t のアーティストを a とし、 L のアーティストの集合を A_L とすると、CAGH のスコアは式 (7) で計算される。

$$score_{CAGH}(t, L) = \sum_{b \in A_L} sim_a(a, b) \cdot count(t) \quad (7)$$

2.2.2 頻出パターン

Association Rule(AR)

プレイリストに相関ルールを適用した推薦手法である。楽曲数 n の楽曲リスト L の楽曲を t_1, t_2, \dots, t_n と表す。 L 内の楽曲から $(t_1), (t_1, t_2), (t_2), (t_1, t_2), \dots, (t_{i-1}, t_i)$ という楽曲数が一曲あるいは二曲の組合せの集合が生成できる。上記の様な楽曲の組合せを ω 、推薦候補となる楽曲を t としたとき、相関ルール $\omega \rightarrow t$ の確信度を用いて楽曲 t の AR のスコアは式 (8) で計算される。

$$score_{AR}(t, L) = \sum_{\omega \in \Omega} confidence(\omega \rightarrow t) \quad (8)$$

ここで、 Ω は L の中から得られる高々楽曲数二曲の組合せの集合である。この集合は、 L の先頭の楽曲から長さ w の楽曲のリストを、先頭から一曲ずつずらして生成していき、得られた複数の楽曲のリスト群中の楽曲同士の組合せの和集合である。このため、 w のサイズや L の長さによっては、 L 内での全ての楽曲同士の組合せを考慮する訳ではない。 $confidence(\omega \rightarrow t)$ は相関ルール $\omega \rightarrow t$ の確信度を表す。確信度は、例えば $(t_1, t_2) \rightarrow t$ の相関ルールでは t_1 と t_2 を含むプレイリストの内 t も含むプレイリストの割合に当たる。従って、AR では二曲又は三曲の楽曲の共起関係から楽曲のスコアを計算している。

2.2.3 近傍推薦

k-Nearest Neighbors(kNN)

プレイリストの類似度を用い L の近傍のプレイリストを抽出し、それらのプレイリストに含まれる楽曲を類似性のある楽曲として推薦する手法 [1] である。 L とプレイリスト p の類似度は式 (9) で計算される。

$$sim_p(L, p) = \frac{||L \cap p||}{\sqrt{||L|| ||p||}} \quad (9)$$

ここで、 $||L||$ は L 内の楽曲数を表す。 L の N 個の近傍のプレイリスト群 N_L が与えられたとき、楽曲 t の kNN のスコアは式 (10) で計算される。

$$score_{kNN}(t, L) = \sum_{p \in N_L} sim_p(L, p) \cdot 1_p(t) \quad (10)$$

ただし, $1_p(t)$ はプレイリスト p に楽曲 t が含まれれば 1 を返し, 含まれなければ 0 を返す. 文献 [10] で kNN の k の値を 300 に設定することで優れた精度を得られることが報告されており, 本稿でも $k = 300$ として検証を行う.

2.2.4 線形結合モデル

kNN で求めた楽曲のスコアに対してユーザの嗜好情報から求めた楽曲のスコアを線形結合で統合する手法 [10] [11] について述べる. また, 楽曲とアーティストの情報を線形和を用いて統合する手法 [5] についても述べる.

Content Preferences and single Artist heuristic(CPA)

Jannach ら [10] は, 二種類の楽曲のスコアを kNN で求めた楽曲のスコアと統合している. 一つ目のスコアは, ユーザが生成した全てのプレイリストから楽曲のタグ情報を用いて TF-IDF [12] ベクトルを生成し, 推薦候補の楽曲とのコサイン類似度から算出した楽曲のスコアである. 二つ目のスコアは, 楽曲リストのアーティストが一組のみの場合, そのアーティストの楽曲の出現回数から算出した楽曲のスコアである. 二組以上の場合の楽曲のスコアは 0 となる. 線形和で統合した結果, 高い推薦精度を得ている.

Multi-dimensional Long-Term Preferences(MLTP)

Kamehkhosh ら [11] は, プレイリストデータだけでなくセッションデータに対しても Hit rate を適用している. セッションデータはユーザの楽曲再生時の時間と楽曲情報を保持したリストである. 楽曲リスト L をユーザの短期的嗜好と見なし, 学習データ内のユーザが生成したプレイリストをユーザの長期的嗜好の情報 (長期履歴) として利用している. 以下の五つの要素でそれぞれ楽曲のスコアを計算している.

- (1) 長期履歴においてより高頻度により最近再生された楽曲に高いスコアを付与
 - (2) 長期履歴内のアーティストに CAGH を適用
 - (3) 長期履歴から楽曲のタグ情報を用いて TF-IDF ベクトルを生成し, 推薦候補の楽曲との類似度を考慮
 - (4) 長期履歴の各プレイリストに対して kNN を適用
 - (5) SNS 上の友人が高頻度に再生する楽曲にスコアを付与
- データセット毎に楽曲のスコアを算出できる要素は異なるがこれらの要素から算出した楽曲のスコアと kNN のスコアを線形和を用いて統合を行っている. 全ての情報を統合した結果, kNN から大幅に精度が向上している.

Linear Combination of Track and Artist(LCTA)

プレイリスト内の楽曲同士の共起関係と, 楽曲とアーティストの関係性をそれぞれグラフで表現し, グラフ構造から算出した楽曲のスコアを線形和で統合する手法である. 図 1 に学習データ内のプレイリストの楽曲とアーティストの関係性を表現したグラフを示す. 楽曲間のエッジは学習データにおける共起回数を表し, 楽曲とアーティスト間のエッジは楽曲の出現回数を表している.

AR は L 内の全楽曲間の共起関係を考慮しておらず, 二曲もしくは三曲間の楽曲の共起関係のみに着目している. 一方, SAGH は, 楽曲を出現回数順にソートしており, アーティスト毎の楽曲の人気度の違いを考慮できていない.

そこで LCTA ではこれらの問題を解決するために, L 内の

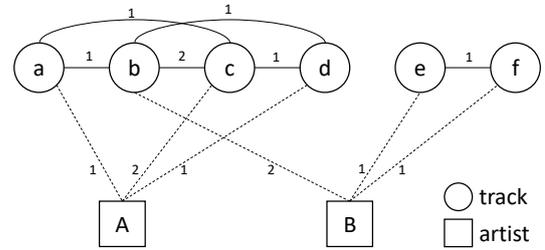


図 1 LCTA におけるグラフ

楽曲と共起する全ての楽曲に対する楽曲のスコア (以後楽曲スコアとする) と, L 内のアーティストの全ての楽曲に対する楽曲のスコア (以後アーティストスコアとする) を提案し, それらを線形和で統合する. この楽曲スコアと, アーティストスコアの詳細について述べる.

楽曲スコア

L 内の一つの楽曲を t_L とし, t_L と共起する楽曲 t の楽曲スコア $score_t$ は式 (11) で計算される.

$$score_t(t_L, t) = \frac{cooc(t_L, t)}{total_cooc(t_L)} \cdot \log\left(\frac{max_count}{count(t_L)} + 1\right) \quad (11)$$

ここで, $cooc(t_L, t)$ は学習データにおける楽曲 t_L と t の共起回数を表し, 図 1 での楽曲間のエッジの重みに当たる. $total_cooc(t_L)$ は楽曲 t_L が共起する全ての楽曲との共起回数の総和を表している. $count(t_L)$ は学習データにおける楽曲 t_L の出現回数を表し, 図 1 の楽曲とアーティストのエッジの重みに当たる. max_count は学習データのプレイリスト群内の楽曲の出現回数の最大値である. 式 (11) の第一項で楽曲 t_L の共起回数に占める楽曲 t の割合を計算し, 第二項で楽曲 t_L の出現回数によるスコアを補正を行っている. これは情報検索における TF-IDF の計算に相当する. L と楽曲 t から最終的な t の楽曲スコアは式 (12) の通りである.

$$score_{track}(t, L) = \sum_{t_L \in L} score_t(t_L, t) \quad (12)$$

アーティストスコア

L に含まれるアーティスト a の楽曲 t のアーティストスコア $score_a$ は式 (13) で計算される.

$$score_a(a, t) = \frac{count(t)}{total_count(a)} \cdot \log\left(\frac{max_listcount}{listcount(a)} + 1\right) \quad (13)$$

ここで, $total_count(a)$ は学習データにおけるアーティスト a の全楽曲の出現回数の総和を表している. $listcount(a)$ は学習データ内でアーティスト a が出現するプレイリストの数を表し, $max_listcount$ は学習データ内でアーティストが出現するプレイリストの数の最大値を表している. 楽曲スコアと同様に TF-IDF のアイデアを取り入れたスコア計算を行っている. L と L 内のアーティストの楽曲 t のアーティストスコアは式 (14) の通りである.

$$score_{artist}(t, L) = \sum_{a \in A_L} score_a(a, t) \quad (14)$$

楽曲スコアとアーティストスコアは正規化を行った上で線形和で統合を行う. 事前に最適な重みを求めた結果, 楽曲スコアに 0.55, アーティストスコアに 0.45 の係数を付ける. L を与えたときの楽曲 t の楽曲スコアを $score_{track}(t, L)$, アーティストスコアを $score_{artist}(t, L)$ とし, それぞれの最大値を

max_score_{track} , max_score_{artist} とすると最終的なスコアは式 (15) で計算される.

$$score_{LCTA}(t, L) = \frac{score_{track}(t, L)}{max_score_{track}} * 0.55 + \frac{score_{artist}(t, L)}{max_score_{artist}} * 0.45 \quad (15)$$

2.2.5 協調フィルタリング

Collaborative Filtering(CF)

情報推薦分野の基本的な手法である協調フィルタリングをベースとした手法であり, 従来のユーザ-アイテム間の推薦をプレイリスト-楽曲間に置き換えて楽曲のスコア計算を行っている. 計算ではプレイリスト (ユーザ) 同士の類似度をアーティスト情報から計算し, 推薦する楽曲 (アイテム) を推薦する形で, ユーザベースの協調フィルタリングをプレイリスト生成に適用している. 事前に検証を行った結果, プレイリストの類似度計算にプレイリストに含まれる楽曲のアーティスト情報を用いた場合が最も精度が高くなったため, プレイリストの類似度計算にはアーティスト情報を用いることとする. L とプレイリスト p の類似度は Jaccard 係数を用いて式 (16) で計算される.

$$sim_p(L, p) = \frac{|A_L \cap A_p|}{|A_L \cup A_p|} \quad (16)$$

ここで, A_L , A_p はそれぞれは L と p のアーティストの集合を表している. 学習データのプレイリスト群を *Playlists* とすると, 楽曲 t の CF のスコアは式 (17) で計算される.

$$score_{CF}(t, L) = \sum_{p \in Playlists} sim_p(L, p) * count_p(t) \quad (17)$$

ここで, $count_p(t)$ はプレイリスト p 内の楽曲 t の出現回数を表す.

2.3 グラフモデルを用いた推薦手法

ユーザやアイテム等の複数のコンテキスト情報を含む情報推薦では, グラフモデルを用いた手法が用いられてきた. これまでにランダムウォーク [6] [7] を用いた手法が提案されている. ユーザやアイテム等のコンテキスト情報からなるグラフ G にランダムウォークを適用する方法について述べる. $G = (V, E)$ と表わすことができ, V は複数のコンテキスト情報の集合, E は V の各ノードを結ぶエッジである.

ランダムウォークでは初期状態として例えばユーザに対してアイテムを推薦する場合は, V 内の一つのユーザノードに初期値 1 を与え, 他のノードに初期値 0 を与える. 各ノードから別のノードへは予め設定した遷移確率 X に基づいて遷移を行う. n 回遷移をした時の状態を S_n とすると, V のノード i (t ステップ) からノード j ($t+1$ ステップ) への遷移確率 $X_{i,j}$ は $X_{i,j} = P(S_{t+1} = j | S_t = i)$ と表現される. 複数回遷移した後に V の各ノードへ遷移する期待値を計算し, 期待値の高い特定のコンテキスト情報のノードを推薦する.

Christoffel ら [6] は映画や書籍等のコンテキスト情報に対してランダムウォークを適用し, 数回の遷移ステップ後の各ノードの期待値を計算することで効率的で高精度の推薦を行っている. Bogers [7] は映画のコンテキスト情報を用いて推薦において, ユーザ, 映画, タグ, 俳優, ジャンルの五つのコンテキスト情報のグラフに対してランダムウォークを適用する手法を提案した. また, ハイパーグラフを併用することで高精度に推薦を行う手法 [13] も提案されている.

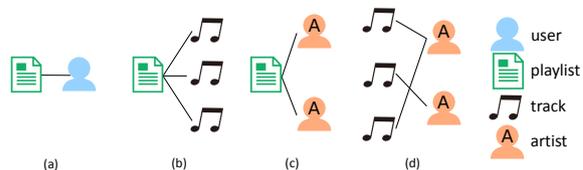


図 2 プレイリスト情報間の関係情報

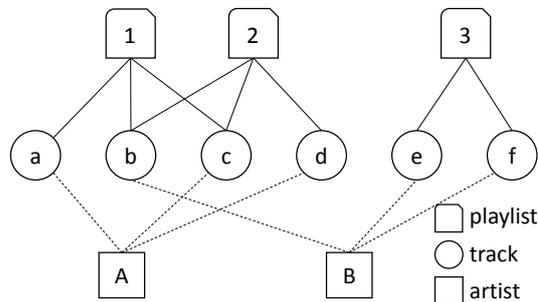


図 3 RW-base のグラフ

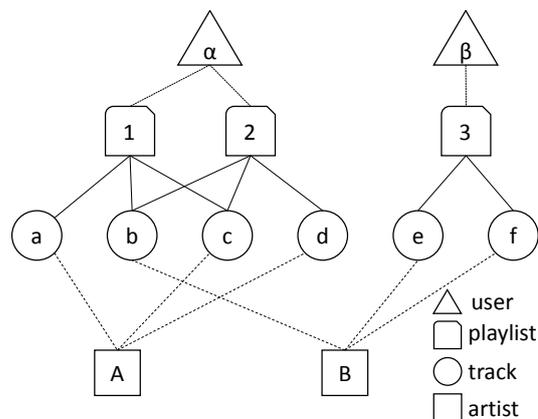


図 4 RW-user のグラフ

3. 提案手法

線形和を用いてプレイリストに付随する複数の要素を統合することは推薦精度の向上に寄与してきた. しかしながらそれぞれの要素を異なる情報として線形和で統合するのではなく, 画一的なグラフで表現することで各情報の重みのチューニングを省略し, 高精度な推薦を行うことを目指す.

そこで, 本稿ではプレイリストに付随する, 楽曲, アーティスト, ユーザのコンテキスト情報をグラフを用いて表現し, ランダムウォークを用いて各楽曲の期待値を計算するプレイリスト生成手法を提案する. また, Hit rate のタスクで与える楽曲リスト L 内の楽曲や L を作成したユーザの情報等, ランダムウォークの各ノードの初期値として複数の値の設定が考えられる. そこで, ノードの初期値の設定方法についても提案を行う.

以降, 本稿で考慮した四種類のグラフについて説明し, その後各グラフにランダムウォークを適用した際の遷移確率の設定とノードの初期値の設定について述べる.

3.1 コンテキスト情報から成る四つのグラフ

最初にプレイリストデータのコンテキスト情報間の関係性について説明し, その後これらの関係性を組み合わせることで生成できる複数のグラフの設定方法について述べる.

プレイリストの持つ, ユーザ, 楽曲, アーティストのコンテ

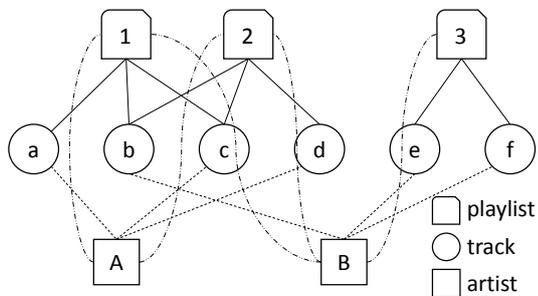


図 5 RW-topic のグラフ

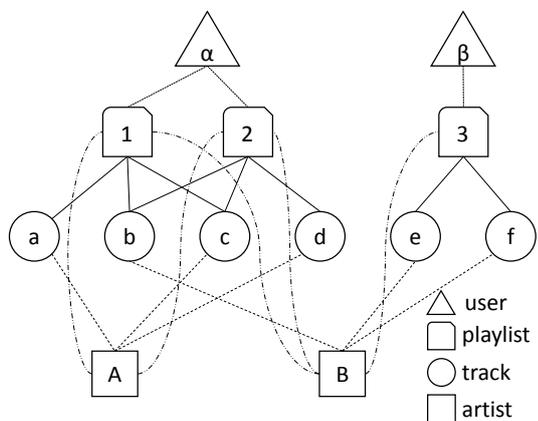


図 6 RW-hybrid のグラフ

キスト情報では、これらの要素間に複数の関係性が考えられる。プレイリストにおける要素同士の関係情報を図 2 に纏めた。(a) はユーザがプレイリストを作成した生成関係である。(b) はプレイリストと、プレイリストに含まれる複数の楽曲との間の参照関係である。(c) はプレイリストと、プレイリストに含まれる複数の楽曲のアーティストとの間の包含関係である。(d) は楽曲とその楽曲のアーティスト間の対応関係である。これらのコンテキスト情報間の関係を組み合わせることで、複数の関係性を画一的に統合したグラフの生成が可能である。

これまでに提案された LCTA 手法は同一のプレイリスト内の楽曲同士の共起関係と、楽曲とアーティストの関係性を用いた手法である。この楽曲間の共起関係とはプレイリストと楽曲の包含関係と置き換えることができる。そこで、プレイリストと楽曲の包含関係と、楽曲とアーティストの対応関係を統合したグラフを生成する。楽曲同士の共起関係を取り除いたことで、楽曲のコンテキスト情報における直接的な楽曲への遷移を取り除き、楽曲からプレイリストを経由して他の楽曲への遷移を行う形に変更する。コンテキスト情報から成るグラフでは同一のコンテキスト情報内で直接遷移は行わず、異なる種類のコンテキスト情報を介して遷移を行うようにエッジを結ぶこととする。このグラフを基に、プレイリストとユーザの生成関係やプレイリストとアーティストの包含関係を組合せ、四つのグラフを設定した。その詳細について述べる。

RW-base

LCTA を基に、プレイリストと楽曲、楽曲とアーティストの関係を取り入れたグラフである。図 3 にグラフを示す。

RW-user

RW-base のグラフに更にプレイリストとユーザの生成関係を取り入れたグラフである。ユーザの生成したプレイリストの

表 1 遷移確率の計算方法

遷移	遷移確率
楽曲 プレイリスト	$\frac{1}{\text{楽曲が含まれるプレイリストの数}}$
プレイリスト 楽曲	$\frac{1}{\text{プレイリスト内の楽曲数}}$
楽曲 アーティスト	1
アーティスト 楽曲	$\frac{\text{楽曲の出現回数}}{\text{アーティストの全楽曲の出現回数}}$
ユーザ プレイリスト	$\frac{1}{\text{ユーザが生成したプレイリストの数}}$
プレイリスト ユーザ	1
プレイリスト アーティスト	$\frac{\text{プレイリスト内のアーティストの楽曲の数}}{\text{プレイリストの楽曲数}}$
アーティスト プレイリスト	$\frac{1}{\text{アーティストが含まれるプレイリストの数}}$

情報を用いることで、ユーザの生成したプレイリストからユーザが好むと考えられる楽曲の期待値を高めることを目的としている。図 4 にグラフを示す。

RW-topic

RW-base のグラフに更にプレイリストとアーティストの参照関係を取り入れたグラフである。プレイリストではプレイリスト、アーティスト、楽曲の関係は文書集合における文書、トピック、単語の関係と見なすことができ、プレイリストからアーティストを介した遷移により、プレイリストと関連すると考えられる楽曲の期待値が高めることが目的である。図 5 にグラフを示す。

RW-hybrid

プレイリストデータの持つ全ての関係情報を統合したグラフである。ユーザのプレイリストの生成情報と、プレイリストからアーティストを介した遷移をどちらも取り入れることで、ユーザの好む楽曲とプレイリストと関係性の高い楽曲の期待値を高めると共に、全てのコンテキスト情報の統合を目的としている。図 6 にグラフを示す。

3.2 ランダムウォークの遷移確率とノードの初期値

提案グラフのランダムウォークにおける、各ノード(コンテキスト情報)から、連結するノード(異なる種類のコンテキスト情報)へのエッジの重み(遷移確率)の設定方法について述べる。その後、各ノードの初期値の設定について述べる。

あるノードから連結するノードへの遷移確率の合計値を 1 と設定する。コンテキスト情報の組合せごとの、遷移確率の設定方法を表 1 に掲載する。なお、あるコンテキスト情報のノードから複数種類のコンテキスト情報のノードに遷移を行う場合、あるノードからの遷移確率は 1 を超えることになる。具体的には、図 3 に示す通り、楽曲ノードからはプレイリストノードとアーティストノードへの遷移を行うため、遷移確率の合計値は 2 となる。そこで、今回用いたグラフでは、ノードからの遷移確率の合計を 1 とするために、表 1 で計算した遷移スコアの値を接続するコンテキスト情報の種類数で割った値で補正する。図 3 の楽曲ノードから他のノードへの遷移確率は表 1 に基づいて計算した値を 2 で割った値になる。

次に、各ノードの初期値の設定方法について述べる。Hit rate を用いたプレイリスト生成問題では、複数の楽曲を含んだ楽曲リスト L を基に、 L と関連すると考えられる楽曲の推薦を行っている。そこで、グラフには楽曲リスト L 内の各楽曲のノードに対して初期値 1 を与え、 L の楽曲に基づいて関連する楽曲の探索を行う。

また、楽曲リスト L を生成したユーザのノードに対しても初

表 2 データセットの概要

	Aotm	Last.fm-2015	Last.fm-2016	30musicS	nowplaying	30musicP
プレイリスト数	1028	2824	2762	9000	9288	25139
ユーザ数	142	451	426	1000	1032	11004
ユーザ当たりのプレイリスト数	7.24	6.26	6.48	9	9	2.28
楽曲数	11411	18081	17813	123315	76652	150957
プレイリスト当たりの楽曲数	17.01	11.72	11.74	25.34	23.89	12.1
楽曲の平均出現回数	1.53	1.83	1.82	1.85	2.89	2.01
アーティスト数	2768	3272	3209	26122	15638	29834
アーティスト当たりの楽曲数	4.12	5.53	5.55	4.72	4.9	5.06
プレイリスト当たりのアーティスト数	12.77	4.56	4.55	9.76	12.56	8.56
アーティスト当たりの平均楽曲出現数	6.31	10.11	10.1	8.73	14.19	10.2
アーティストの再出現率	0.267	0.651	0.654	0.659	0.486	0.304

期値として 1 を与えることで、同一ユーザの別のプレイリストの情報も考慮し、ユーザの嗜好を取り入れた遷移を行うことを目指す。さらに、協調フィルタリングのプレイリスト同士の類似度から楽曲の推薦を行う考えを基に、楽曲リスト L と類似するプレイリストノードに対しても初期値として、式 (16) に基づく類似度を与えることとした。これにより、プレイリストとアーティスト間の遷移によってプレイリストから関連すると考えられる楽曲への遷移の取り入れが期待できる。ただし、プレイリストノードに対して L との類似度を初期値として設定したのは、プレイリストとアーティスト間の遷移を行う RW-topic, RW-hybrid のみである。それ以外では初期値として 0 を設定した。なお、その他のコンテキスト情報のノード、すなわち、アーティストノードには初期値として 0 を設定した。

4. 評価実験

本章では、提案手法の有用性の評価を行う。その際のデータセットとして、6 種類のプレイリストとセッションデータを用いる。セッションデータはユーザの楽曲再生時の時間と楽曲情報を保持したリストである。プレイリストと同様にこの楽曲のリストは特定の目的を持ち、その目的は再生時のユーザの気分や環境を反映したものの判断される。評価尺度には Hit rate 及び MRR を用いた。データセットの詳細と交差検証、比較を行った手法、実験結果について述べる。

4.1 データセット

文献 [10] で紹介されている Art-of-the-Mix(Aotm) と Last.fm のプレイリストデータと、文献 [11] で紹介されている Last.fm のプレイリストデータ、30music 及び nowplaying のセッションデータと、文献 [14] で紹介されている 30music のデータセットから作成したプレイリストのデータセットを使用する。以後、文献 [10] の Last.fm のデータセットを Last.fm-2015 とし、文献 [11] の Last.fm のデータセットを Last.fm-2016 とする。また、文献 [11] の 30music のセッションデータを 30musicS、30music のデータセットから作成したプレイリストのデータセットを 30musicP とする。特に、30musicP のデータはデータ数のばらつきが大きいため、それぞれリストの楽曲数に上限と下限を設けデータ数を制限している。今回使用した六つのデータセットの詳細を表 2 に示す。

アーティストの再出現率はプレイリストの最後の楽曲のアーティストが、既にそのプレイリストの中で出現しているかの割合である。Hit rate の楽曲リスト L と正解楽曲 rel を生成した際、 rel のアーティストが L 内に出現する割合を表してい

る。Aotm はプレイリスト数が一番小さく、楽曲の出現回数やアーティストの楽曲数が少ない。また、プレイリスト内に出現するアーティストの数が多のが特徴である。Last.fm-2015 と Last.fm-2016 はデータセットは類似性が高く、プレイリスト当たりのアーティストが少なく特定のアーティストの楽曲のみを含むプレイリストが多いのが特徴である。30musicS 及び nowplaying はセッションデータであり、ユーザ当たりのプレイリスト数が多い。また、プレイリスト当たりの楽曲数が多いがプレイリスト内でのアーティスト数の割合は Aotm 程高くない傾向にある。30musicP は最もプレイリストの数が多いデータセットであるがユーザ当たりのプレイリスト数が少なく、プレイリスト内のアーティストの数が比較的多いことが特徴である。

4.2 交差検証

六つのデータセットを用いて、プレイリストデータは四分分割交差検証、セッションデータは四分分割スライド窓検証で実験を行った。四分分割スライド窓検証では、初めにセッションデータ群を時系列順に並べ、九個のブロックに分割し、時系列で新しいブロックから連続した六つのブロックを取り、その六つのブロックの中で最も新しいブロックをテストデータ、それ以外の五つのブロックを訓練データとした。この六つのブロックを一ブロックずつ古いブロックの方向にずらし、四回の検証を行った。

4.3 比較した手法

実験では既存手法として SAGH, CAGH, kNN, AR, CF, CPA, MLTP, LCTA を用い、提案手法である RW-base, RW-user, RW-topic, RW-hybrid と精度の比較を行った。なお、CPA は Aotm と Last.fm-2015 における結果を載せ、MLTP は Aotm, Lastfm-2016, 30musicS, nowplaying における結果を載せる。ランダムウォークを用いた四つの手法では遷移ステップ毎に楽曲の期待値を計算し、最大 4 回まで遷移させ精度を計算した。これは、5 回目以降の遷移では 4 回までの遷移よりも精度が低下したためである。また、既存手法の中で MLTP が最も精度が高かったが、MLTP には存在しないデータセットが存在するため、代わりに LCTA を基準として符号検定を行い、 p 値が 0.05 未満であった手法には * を付けた。

4.4 実験結果

推薦リストの長さを 100 とした場合の各データセットの Hit rate を表 3 に示す。ランダムウォークの手法名の右側の数値は遷移ステップ数である。

表 3 の結果より、Hit rate において RW-hybrid が Aotm 以外のデータセットで最も高い精度を示した。特に、30musicP では、既存手法の中で最も精度の高かった SAGH, CF に対して、

表 3 Hit rate 精度結果

Algorithm	Accuracy@100(Hit rate)						
	Aotm	Last.fm-2015	Last.fm-2016	30musicS	nowplaying	30musicP	
SAGH	.072	.314	.292	.241	.247	.140*	
CAGH	.114	.329	.292	.219	.190	.134	
kNN	.099	.323	.269	.253	.285	.111	
AR	.093	.321	.267	.251	.268	.103	
CF	.125	.350	.296	.244	.255	.140*	
CPA	.130	.359					
MLTP	.143		.315	.269	.323		
LCTA	.116	.360	.307	.266	.297	.133	
RW-base	1	.132	.345	.290	.237	.270	.128
	2	.109	.356	.302	.280*	.320*	.149*
	3	.137	.357	.305	.247	.268	.148*
	4	.116	.356	.303	.276*	.312*	.152*
RW-user	1	.132	.345	.290	.237	.270	.128
	2	.109	.365	.311	.286*	.331*	.162*
	3	.139	.357	.306	.247	.275	.144*
	4	.113	.368	.315	.283*	.333*	.164*
RW-topic	1	.132	.345	.290	.237	.270	.128
	2	.116	.363	.313*	.285*	.322*	.160*
	3	.139*	.360	.311	.263	.294	.155*
	4	.129	.365	.313	.275*	.305	.161*
RW-hybrid	1	.132	.345	.290	.237	.269	.128
	2	.118	.373*	.323*	.289*	.336*	.170*
	3	.141*	.365	.314	.265	.302	.154*
	4	.129	.375*	.325*	.281*	.329*	.170*

RW-hybrid は 4 ステップ後の精度で最大となる 21%の精度向上を達成し、線形結合を用いた LCTA に対しては 27%の精度向上を達成した。線形結合を用いない既存手法である SAGH, CAGH, kNN, AR, CF の中では CF の精度が高かったが、線形和を用いた CPA や MLTP, LCTA はこれら手法の結果を上回った。

既存手法の中では CF の精度が比較的高い精度を示したことから、アーティスト情報に基づくプレイリストの類似度を用いた推薦は有効であることがわかる。これは同様の類似度をプレイリストノードの初期値に設定し、プレイリストとアーティスト間の遷移を取り入れた RW-topic や RW-hybrid の Hit rate の値が、RW-base や RW-user より向上していることから推薦精度を高める上で効果的な戦略であると推測される。

遷移ステップ数については殆どのデータセットにおいて 2 回の遷移で十分に高い精度が出ているが、Aotm のみ 3 回の遷移で最も精度が高くなる傾向が見られる。これは、Aotm はプレイリスト内のアーティストのばらつきが大きいため、ある程度遷移が進み L と異なるアーティストの楽曲への遷移確率が高まらなければ正解率が上がらないからであると考えられる。

楽曲間と楽曲-アーティスト間の関係を取り入れた LCTA と RW-base の精度を比較する。六つのデータセットにおいて LCTA と RW-base の 2,4 ステップ後の Hit rate の結果はほぼ同程度となった。この二つの手法の推薦精度は近い値であるが、ランダムウォークはコンテキスト情報の種類が増えても拡張が容易であり、線形和の最適な重みの設定も必要ないという利点がある。

CPA, MLTP と RW を比較する。CPA と MLTP は楽曲の

タグ情報を用いてユーザの嗜好情報を線形和で取り入れた手法であるが、タグ情報を用いずに、プレイリストの基本情報のみを用いた RW-hybrid は Hit rate においてこれらの手法と同等あるいは上回る結果となっている。ユーザの嗜好情報としてユーザが生成したプレイリストの情報も RW-hybrid では取り入れていることから、楽曲やアーティスト、ユーザの関係性を一つのグラフ内で上手く取り入れることができ、精度が向上したと推測される。

次に、各データセットの MRR の結果を表 4 に示す。

MRR の結果では線形結合を用いた CPA, MLTP, LCTA が高い精度を示した。これらの手法は線形和を用いて複数の戦略から算出した楽曲のスコアを統合しているため、複数の戦略に合致する楽曲ほどスコアが高まり順位が押し上げられると考えられる。一方で、RW-hybrid ではこれらの手法を Hit rate で上回っているが正解楽曲をより上位に配置できていない。ただし、今回のタスクでは正解楽曲以外に楽曲リスト h と関連する楽曲を評価の際に不正解 (不適合) と扱うため、推薦リスト内の順位付けが必ずしも推薦精度を表現しているとは限らない。順位付けの必要性については被験者実験による検証が必要である。

5. まとめ

本稿では、プレイリストの持つ複数の情報をグラフを用いて統合し、ランダムウォークを適用することで遷移確率に基づいた楽曲の推薦を行った。与えられた楽曲のリストから関連する楽曲を予想するというタスクである Hit rate において、これまで線形和を用いて楽曲やアーティストの情報を統合した手法が提案されてきた。しかし、線形和を用いた手法では線形結合の

表 4 MRR 精度結果

Algorithm	Accuracy@100(MRR)						
	Aotm	Last.fm-2015	Last.fm-2016	30musicS	nowplaying	30musicP	
SAGH	.009	.057	.054	.044	.039	.022	
CAGH	.010	.052	.049	.036	.027	.019	
kNN	.008	.047	.038	.051	.046	.011	
AR	.008	.047	.037	.048	.046	.012	
CF	.016	.059	.049	.040	.041	.020	
CPA	.022	.102					
MLTP	.022		.076	.053	.054		
LCTA	.012	.074	.058	.062	.051	.022	
RW-base	1	.016	.056	.048	.041	.043	.018
	2	.010	.059	.051	.054	.051	.020
	3	.013	.057	.051	.037	.040	.020
	4	.011	.059	.051	.049	.048	.021
RW-user	1	.016	.056	.048	.041	.043	.018
	2	.010	.058	.051	.052	.049	.021
	3	.013	.057	.051	.038	.041	.020
	4	.010	.058	.051	.049	.048	.022
RW-topic	1	.016	.056	.048	.041	.043	.018
	2	.010	.062	.054	.054	.050	.022
	3	.013	.061	.051	.043	.044	.021
	4	.011	.060	.052	.048	.046	.022
RW-hybrid	1	.016	.056	.049	.041	.043	.018
	2	.010	.060	.053	.053	.050	.022
	3	.014	.060	.050	.043	.045	.021
	4	.011	.060	.052	.048	.049	.022

重み付けを予め設定する必要があった。評価実験の結果、提案手法は従来の線形和を用いた手法に対して最大で 27% の精度向上を達成した。

今後の課題は、音源やタグ情報からより最適な遷移確率の設定を行うことと、ユーザがプレイリストを生成する際に考慮したと考えられる生成の意図をプレイリストから予想することで、例えば新しい楽曲を探索してる場合やこれまで聞いた曲の中から好きな楽曲を探すなどの状況に合った楽曲の予想を可能にすることである。また、実際に被験者実験を行い、ユーザからの満足度の向上を計測し、手法の有効性を検証することである。

謝 辞

本研究の一部は、JSPS 科研費 JP26280115, JP15H02701, JP16H02908, JP15K20990 の助成を受けたものである。ここに記して謝意を表す。

文 献

- [1] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latent topic sequential patterns. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 131–138. ACM, 2012.
- [2] Geoffroy Bonnin and Dietmar Jannach. A comparison of playlist generation strategies for music recommendation and a new baseline scheme. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
- [3] Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 909–916. ACM, 2012.
- [4] Geoffroy Bonnin and Dietmar Jannach. Automated generation of music playlists: Survey and experiments. *ACM Computing Surveys (CSUR)*, 47(2):26, 2014.
- [5] 植田聖司, 樺惇志, and 宮崎純. プレイリスト生成におけるグラフモデルを用いたスコアリング手法の提案. 第 8 回データ工学と情報マネジメントに関するフォーラム (*DEIM2016*) 論文集, 2016.
- [6] Fabian Christoffel, Bibek Paudel, Chris Newell, and Abraham Bernstein. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 163–170. ACM, 2015.
- [7] Toine Bogers. Movie recommendation using random walks over the contextual graph. In *Proc. of the 2nd Intl. Workshop on Context-Aware Recommender Systems*, 2010.
- [8] Ellen M Voorhees et al. The trec-8 question answering track report. In *TREC*, volume 99, pages 77–82, 1999.
- [9] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.
- [10] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. Beyond hitting the hits: Generating coherent music playlist continuations with the right tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 187–194. ACM, 2015.
- [11] Iman Kamehkhosh, Dietmar Jannach, and Lukas Lerche. Personalized next-track music recommendation with multi-dimensional long-term preference signals. 2016.
- [12] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.
- [13] Brian McFee and Gert RG Lanckriet. Hypergraph models of playlist dialects. In *ISMIR*, pages 343–348. Citeseer, 2012.
- [14] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 30music listening and playlists dataset. 2015.