

## 協調フィルタリングを用いた音楽推薦と マッピング手法

新美怜志<sup>†</sup> 濱川礼<sup>††</sup>

本論文では、再生履歴からユーザーが聴きたい音楽を推測、推薦するシステムについて述べる。システムは、再生履歴を蓄積し、協調フィルタリングを用いて音楽間のユーザーの嗜好情報を得て、ユーザーの所有している音楽の中から現在のユーザーに最適な音楽を推薦する。また、蓄積された再生履歴から個々のユーザーの音楽間の相関度を算出し、音楽マップを作成する。

## Music recommendation and mapping for collaborative filtering

Satoshi Niimi<sup>†</sup> and Rei Hamakawa<sup>††</sup>

In this thesis, the system that guesses, and recommends music to want to hear the user from the play history is described. The system stores the play history plays, with the music preference information among users for collaborative filtering to recommend music, best music, the current user from the user owns. Also, calculate the correlation between playing music stored on your own history, to create a musical map.

### 1. はじめに

我々は、再生履歴からユーザーの現在の気分に沿った音楽を推薦するシステムを開発した。近年、携帯音楽プレーヤの大容量化や iTunes Music Store 等の音楽配信サービスが登場したことによって大量の音楽を保存、再生することが可能になった。例として、160G の大容量を誇る iPod classic では約 40,000 曲もの音楽を保存することができる。また、iTunes Music Store では、2008 年 6 月 23 日時点で累計 50 億曲以上の音楽を販売したことを発表した。

これにより、ユーザーが大量の音楽の中から聴きたい音楽を取捨選択していくことが必要となった。そのため、ユーザーの所有している音楽からユーザーの好みを推測し、ユーザーの所有していない音楽を新たに推薦する手法についての研究が多く行われてきた [1][2]。しかし、ユーザーの所有している音楽からユーザーが現在聴きたい音楽を推薦する手法については上記の研究と比べ、あまり行われていない。ユーザーの所有している音楽も増えてきている中、ユーザーの聴きたい音楽は気分によって頻繁に変化していくと考えられるため、ユーザーの気分に沿った音楽を推測し、推薦する手法についての研究は需要があると考えられる。また、既存プレーヤではアーティスト名やアルバム名、ユーザーが評価したレーティング等を用いて選曲を手助けしているものもあるが、ユーザーの気分は一定ではないため、上記の方法で選曲した音楽が必ずしもユーザーの現在の気分に沿っているとは限らない。そのため、ユーザーが音楽を聴く際に、大量の音楽から聴きたい音楽を選択する手間が生じるといった問題が挙げられる。

そこで、本研究ではユーザーの現在の気分に沿った音楽を推薦することで、ユーザーの選曲の手助けになると考え、本システムを開発した。また、本研究ではユーザーの気分によって聴きたい音楽が変化し、気分が近い時には聴きたい音楽も類似すると仮定している。

### 2. 関連研究と本研究の特徴

本研究の関連研究として、ユーザーがスキップした音楽を「今聴きたくない音楽」と判断し、音響特徴の類似度が高い音楽を次から選曲しないことでユーザーが今聴きたい音楽を推薦する研究がある [3]。しかし、音響特徴を用いる手法では、スキップした音楽と類似度の高い音楽の中にユーザーの聴きたい音楽があった場合に推薦されにくくなるという問題がある。

そこで我々は、再生履歴を用いてユーザーの気分に沿った音楽の推薦を行う手法を提

<sup>†</sup> 中京大学大学院  
Chukyo University graduate school

<sup>††</sup> 中京大学  
Chukyo University

案する。現在の気分に近い過去の気分の時に気に入った音楽を推薦することでユーザの現在の気分に沿った音楽を推薦できると考えられる。また、音響特徴によってユーザの聴きたい音楽が推薦されにくくなる問題が解決できる。

本研究では、システム起動から終了までの再生履歴を蓄積しておき、システム起動から現在までの再生履歴（以後、現在の再生履歴と呼ぶ）と過去にシステムを使用した際に蓄積した再生履歴の集合（以後、過去の再生履歴群と呼ぶ）をユーザベース協調フィルタリングを用いて比較することで、ユーザの嗜好情報を得て、現在のユーザに沿った音楽を推薦する。本研究では再生履歴を元に音楽を推薦しているため、再生履歴が蓄積されるまでは効果的な推薦が行われないコールドスタート問題があると考えられる。そのため、現在の再生履歴に依存しない、過去の再生履歴群を用いた音楽マップを作成した。これによってユーザが現在聴きたい音楽を探しやすくなる。

### 3. システム構成

本システムは、ユーザインタフェース、データベース、音楽推薦モジュール、マップ作成モジュールの4つで構成される。

ユーザインタフェースでは、通常の音楽プレーヤ同様の操作(再生,停止,次の音楽,etc)で音楽を聴くことが出来る。また、マップ作成モジュールで作成されたマップの表示、音楽推薦モジュールで推薦された音楽の出力を行う。データベースでは、システム起動から終了までに聴いた音楽とそれぞれの音楽に対する評価値を蓄積する(4.1章)。音楽推薦モジュールでは、ユーザベース協調フィルタリングを用い、ユーザが現在聴きたい音楽を推薦する(4.2章)。マップ作成モジュールでは、アイテムベース協調フィルタリングを用い、個々のユーザの音楽間の相関度を算出し、音楽マップを作成する(4.3章)。システム構成図を以下に示す(図1)。

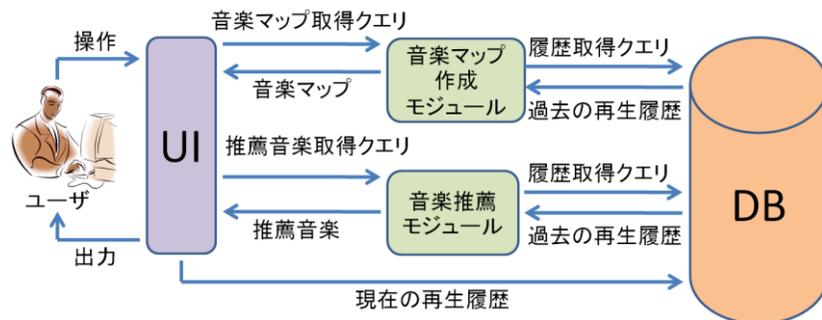


図1 システム構成図

システム起動時、ユーザインタフェースからマップ作成モジュールに対して、マップ取得クエリが送られる。マップ作成モジュールは、データベースに履歴取得クエリを送り、再生履歴と評価値を取得する。評価値については、4.1章で詳しく述べる。そして、マップ作成モジュールは取得したデータを元に音楽マップを作成し、ユーザインタフェースで音楽マップを表示する。また、再生される音楽が変わった場合、ユーザインタフェースが音楽推薦モジュールに対して、推薦音楽取得クエリを送る。音楽推薦モジュールは、データベースに履歴取得クエリを送り、再生履歴と評価値を取得する。そして、音楽推薦モジュールは取得したデータを元に推薦音楽を決定し、ユーザインタフェースで推薦音楽を表示する。最後に、システムを終了した時、現在の再生履歴をデータベースに保存することで、次回以降の音楽マップ作成と音楽推薦に反映させる。

### 4. システム詳細

#### 4.1 データベース

データベースでは、音楽マップの作成と音楽推薦に必要な再生履歴とそれぞれの音楽に対する評価値を蓄積する。評価値は、音楽がユーザの現在の気分に沿っているかを数値化したものである。ユーザの現在の気分に沿っているかの判断として、我々はユーザが音楽を聴き時間に着目した。気分に沿わない音楽に対し、ユーザは音楽を飛ばすという操作を行うためである。そこで、評価値  $e$  を以下のように定めた。

$$e = \text{PlayTime} / \text{MaxTime} \quad (0 < e \leq 1)$$

$\text{PlayTime}$  : 実際に再生した時間,  $\text{MaxTime}$  : 音楽全体の時間

#### 4.2 音楽推薦モジュール

現在の再生履歴と過去の再生履歴群を比較することで、ユーザが現在聴きたい音楽を推測し、推薦する。比較方法は、ユーザベース協調フィルタリングの相関係数法を用いる。相関係数法は以下の手順で行う。

##### 4.2.1 過去と現在の再生履歴と音楽の評価値を取得

データベースから過去の再生履歴群と評価値を取得する。例を以下に示す(表1)。空欄は未再生であることを表す。

表1 過去と現在の再生履歴と評価値

音楽\履歴	現在	過去1	過去2	過去3	過去4
音楽A	0.3	0.6	0.7	1	0.4
音楽B		0.1	0.8	0.6	1
音楽C	1	0.25		0.3	1
音楽D	0.3		0.1		0.3
音楽E		0.7	0.2		0.1
音楽F			0.7	1	0.1

#### 4.2.2 現在の再生履歴と過去の再生履歴群の相関度の算出

現在の再生履歴と過去の再生履歴群を比較し、再生履歴の相関度を算出する。相関度は以下の式を用いて算出する。

$$r_j = \frac{\sum_i (N_i - A)(P_{ij} - A)}{\sqrt{\sum_i (N_i - A)^2} \sqrt{\sum_i (P_{ij} - A)^2}}$$

$r_j$  : 過去の再生履歴 j と現在の再生履歴の相関度,  $A$  : 評価値の平均値

$N_i$  : 現在の再生履歴の音楽 i の評価値,  $P_{ij}$  : 過去の再生履歴 j の音楽 i の評価値

表1から相関度を求めた例を以下に示す(表2)。

表2 相関度の算出

音楽\履歴	現在	過去1	過去2	過去3	過去4
音楽A	0.3	0.6	0.7	1	0.4
音楽B		0.1	0.8	0.6	1
音楽C	1	0.25		0.3	1
音楽D	0.3		0.1		0.3
音楽E		0.7	0.2		0.1
音楽F			0.7	1	0.1
相関度		-0.91	0.44	-1	0.99

表2で、現在と過去4の再生履歴に対して、音楽Aと音楽Dの評価が低く、音楽Cの評価が高いことから、この2つの再生履歴は相関度の高い履歴であると言える。逆に、過去1、過去3は音楽Aの評価が高く、音楽Cの評価が低いことから、相関度の

低い履歴であると言える。

#### 4.2.3 予測値の取得

4.2.1章で取得した評価値と4.2.2章で算出した相関度を用いて、システム起動からまだ再生していない音楽の予測値を算出する。予測値は、ユーザの現在の気分でユーザが音楽に与えるであろう評価値を予測した値である。予測値は以下の計算式を用いて算出する。

$$F_i = A + \frac{\sum_j (P_{ij} - A) * r_j}{\sum_j |r_j|}$$

$F_i$  : 音楽 i に対する予測値,  $P_{ij}$  : 過去の再生履歴 j の音楽 i に対する評価値

$r_j$  : 過去の再生履歴 j と現在の再生履歴の相関度

表2から予測値を求めた例を以下に示す(表3)。

表3 予測値の算出

音楽\履歴	現在	過去1	過去2	過去3	過去4	予測値
音楽A	0.3	0.6	0.7	1	0.4	
音楽B		0.1	0.8	0.6	1	0.75
音楽C	1	0.25		0.3	1	
音楽D	0.3		0.1		0.3	
音楽E		0.7	0.2		0.1	0.2
音楽F			0.7	1	0.1	0.37
相関度		-0.91	0.44	-1	0.99	

表3で音楽Bは、相関度の高い過去4では高評価を得、相関度の低い過去1で低評価を得ていることから、現在の気分に沿っていると判断し、予測値が高くなっている。また、音楽Eでは、相関度の高い過去4で低い評価を得、相関度の低い過去1で高評価を得ていることから、現在の気分に沿わないと判断され、予測値が低くなっている。

これにより算出した予測値が閾値以上の音楽を推薦音楽とする。この場合では、音楽Bが最も良い音楽として推薦される。

#### 4.3 音楽マップ作成モジュール

4.2章の手法では、現在の再生履歴が少ない場合(システム起動直後)、現在の再生履歴と過去の再生履歴群の比較ができないため、音楽を推薦することができないといっ

た問題が挙げられる。そこで、音楽を推薦出来ない場合でも、ユーザが聴きたい音楽を探しやすくすることを目的とし、現在の再生履歴に左右されないアイテムベース協調フィルタリングを用いて音楽マップを作成する。

過去の再生履歴群から、個々のユーザの音楽間の相関度を算出し、相関度の高い音楽を接続する。相関度の算出にはアイテムベース協調フィルタリングを用い、マップの作成には Eades のばねモデルを用いる。Eades のばねモデルとは以下のようなモデルを作成し、物理シミュレーションすることで、物体を配置するアルゴリズムである。

- 各ノード(頂点)は、質量はあるが、体積が 0 の理想物体と仮定する。
- エッジ(頂点と頂点を接続する線)は、ある長さのばねでできていると仮定。ばねは元の長さより長いと縮まり、短いと伸びる。
- ノード同士には反発力が働く。

音楽マップ作成は以下の手順で行う。

#### 4.3.1 音楽間の相関度の算出

蓄積した過去の再生履歴群から、セットで再生された音楽の回数を算出する。ただし、評価値が閾値以下の音楽に対してカウントしない。これは、再生されてすぐに聴く音楽を変えた場合、その音楽は気分に沿っていないと判断できるためである。表 1 からセットで再生された音楽の回数をカウントした例を示す(表 4)。

表 4 セットで再生された音楽の回数

音楽\音楽	音楽 A	音楽 B	音楽 C	音楽 D	音楽 E	音楽 F
音楽 A		2			1	2
音楽 B	2		1			2
音楽 C		1				
音楽 D						
音楽 E	1					
音楽 F	2	2				
和	5	5	1		1	4

その後、表 4 の組み合わせの総和から、セットで再生される音楽の割合を算出する(表 5)。算出された割合が閾値を越えている場合、その音楽はユーザにとって相関度の高い音楽と判断し、ばねモデルで接続する。例として、閾値が 0.3 の場合、表 5 の青色で表示された音楽が相関度の高い音楽となる。

表 5 セットで再生される音楽の割合

音楽\音楽	音楽 A	音楽 B	音楽 C	音楽 D	音楽 E	音楽 F
音楽 A		0.4			1	0.5
音楽 B	0.4		1			0.5
音楽 C		0.2				
音楽 D						
音楽 E	0.2					
音楽 F	0.4	0.4				
和	1	1	1		1	1

#### 4.3.2 ばねの弾性力を適応

ノードとノードを接続するエッジはばねで出来ているため、ノードがばねから受ける力を算出する。計算式は以下ようになる。

$$\begin{pmatrix} F_x \\ F_y \end{pmatrix} = -k * (d - l) \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$$

$l$ :ばねの自然長,  $k$ :ばねの強さを表す比例定数,  $d$ :ノード間の距離

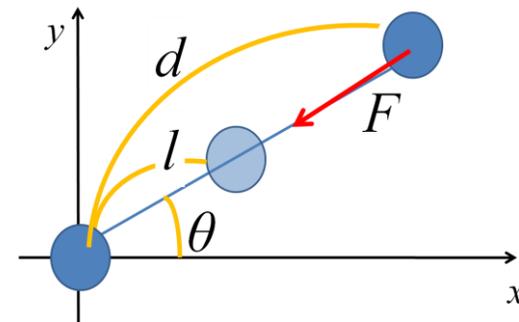


図 2 ばねの弾性力

#### 4.3.3 ノードの反発力を適応

反発力はばねで接続されているかどうかに関わらず、全てのノード間で働く。反発力は、逆二乗の法則に従う。つまり、力  $F$  の大きさが距離  $d$  の二乗に反比例する。計算式は以下ようになる。

$$\begin{pmatrix} F_x \\ F_y \end{pmatrix} = \frac{g}{d^2} \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}$$

$g$ :反発力の強さを表す比例定数

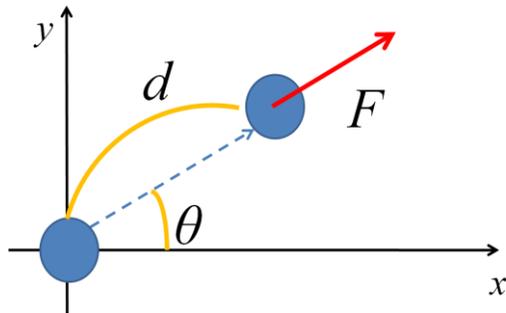


図3 ノードの反発力

#### 4.3.4 摩擦力を適応

上記だけでは、ノードが常に動きつづけてしまうため、摩擦力を加えることで速度を低下させる。計算式は以下になる。

$$\begin{pmatrix} F_x \\ F_y \end{pmatrix} = -\mu \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

$\mu$ :摩擦力を表す比例定数

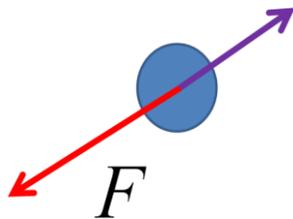


図4 ノードの摩擦力

上記3つをノードに適用させることで、下記のようなグラフを取得できる(図5)。

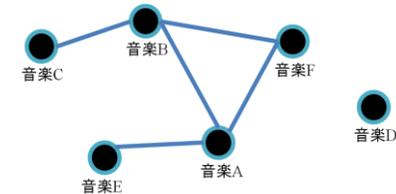


図5 音楽マップの例

#### 4.4 ユーザインタフェース

本システムのユーザインタフェースを以下に示す(図6)。図6は、音楽を19曲追加し、10回システムを起動した後(1回に約5~6曲再生)に作成されたマップである。

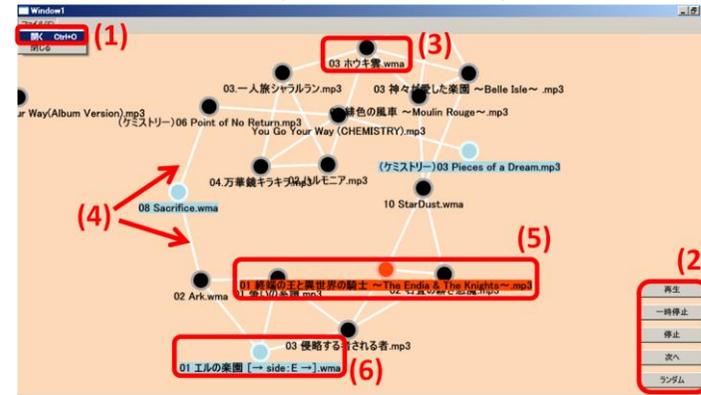


図6 ユーザインタフェース

- ユーザの入力
  - (1) 音楽の追加  
追加ボタン, またはドラッグ&ドロップで音楽を追加する。
  - (2) 操作ボタン  
「再生」, 「停止」, 「次の音楽へ」などのボタンで音楽を操作する。
- システムの出力
  - (3) ノード  
音楽ファイルをマップ上にノードとして表示する。

#### (4) エッジ

4.3 章で算出した相関度の高い音楽同士をエッジで接続する。

#### (5) 再生中の音楽

再生中の音楽のノードとファイル名を赤く表示する。

#### (6) 推薦された音楽

4.2 章で取得した推薦音楽のノードとファイル名を青く表示する。

### 5. 評価

本システムが研究の目的を達成できているかの調査のため、2 種類の評価実験を行った。

#### 5.1 本研究で推薦された音楽の評価

本研究の学生 22 名に対し、評価実験を行った。「本研究で推薦された音楽」、「タグ情報が同じ音楽」「ランダムで選択した音楽」をそれぞれ表示し、被験者に現在聴きたい音楽を選択してもらった。その結果、それぞれ選択した割合をグラフとして以下に示す(図 7)。タグ情報には、アーティスト、アルバム、年、ジャンルを用いている。

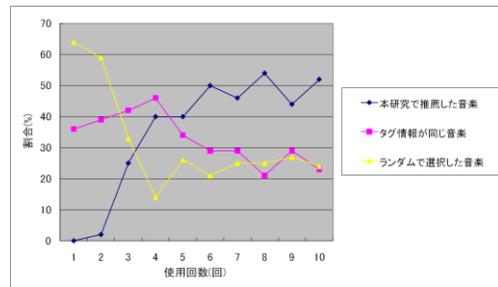


図 7 選択した音楽の割合考察

図 6 から、使用回数が少なく、再生履歴が蓄積されていない場合は選択されることが少ないが、使用回数が増え、再生履歴が蓄積されると本研究の推薦が最も多く推薦されている。上記から本システムの推薦は、最もユーザの気分に沿った音楽を推薦できたと言える。

#### 5.2 アンケート形式の評価

本研究の学生 15 名を対象に、実際にシステムを使ってもらい、アンケート形式の調査を行った。それぞれの項目に対し、1~4(4 が最も良い)の数字を割り振ってもらった。項目、結果(それぞれの項目の平均値)は以下に示す(図 8)

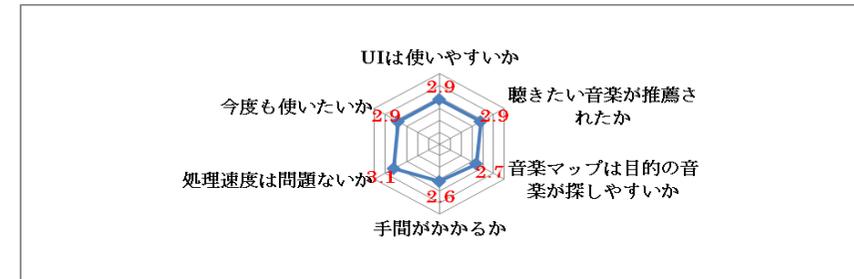


図 8 アンケート結果

図 8 から、全体的に良い結果が得られたと言える。また、「聴きたい音楽が推薦されたか」の項目が高評価なことから、本研究の目的である、ユーザの気分に沿った音楽の推薦が達成できたと言える。しかし、「ファイル名が長いと重なって見にくい」「エッジが重なりすぎて見にくい」といったコメントが多く得られたため、ユーザインタフェースの表示については課題が残る結果となった。

### 6. 考察

本研究では、ユーザベース協調フィルタリングを用いた音楽推薦とアイテムベース協調フィルタリングを用いた音楽マップの作成を行った。

音楽推薦では、現在の再生履歴と過去の再生履歴群に対して、ユーザベース協調フィルタリングを用いることでユーザの嗜好情報を得て、ユーザの気分に沿った音楽推薦を行うことができた。また、アイテムベース協調フィルタリングを用いた音楽マップの作成によって、現在の再生履歴が少なく、音楽推薦が行えない場合にもユーザが聴きたい音楽を探しやすくすることができた。

しかし、評価結果からユーザインタフェースや再生履歴が蓄積されていない場合などについての改善の必要性がでてきた。

### 参考文献

- 1) 梶 克彦, 平田 圭二, 長尾 確: 状況と嗜好に関するアノテーションに基づくオンライン楽曲推薦システム, 情報処理学会 音楽情報科学研究会 研究報告 2004-MUS-58, pp33-38 (2004)
- 2) 吉井 和佳, 後藤 真孝, 駒谷 和範, 尾形 哲也, 奥乃 博: ユーザの評価と音響的特徴との確率的統合に基づくハイブリッド型楽曲推薦システム, 情報処理学会 音楽情報科学研究会 研究報告 2006-MUS-66-8, Vol. 2006, No. 90, pp. 45-52 (2006)
- 3) 彦坂 健太郎, 谷口 徹, 菅田 雅彰, 白井 克彦: ユーザの嗜好に適應させた選曲システムの提案, 情報処理学会 音楽情報科学研究会 研究報告 2006-MUS-68, Vol. 2006, No. 133, pp. 19-24 (2006)