

ソーシャルタグを用いた類似楽曲検索システム

Content-based music information retrieval using SNS tags

萩原 智彰¹ 山田 誠二^{1,2,3}
Tomoaki HAGIHARA¹ Seiji YAMADA^{1,2}

¹ 東京工業大学

¹ Tokyo Institute of Technology

² 国立情報学研究所/総合研究大学院大学

² National Institute for Informatics/SOUKENDAI

Abstract: Recently, increasing digital music collections make us difficult to find new favorite music which we are really interested in. To overcome this problem, content-based music information retrieval has been studied. It enables users to search similar music with users choose, however few method can improve search result interactively. In this paper, we propose a new method that allows users to reflect their search intention. By selecting SNS tags, users can naturally inform their intention to a music retrieval system, and the system utilize user feedback to modify a similarity function. we employ learning to rank technique to utilize tag information to modify a feature vector space.

1 序論

1.1 研究背景

近年、定額楽曲配信サービスの登場などにより、個人で取り扱える楽曲の量は爆発的に増えているが、ユーザが自分の好む楽曲を見つけることはますます困難になっている。ユーザの楽曲発見を支援するためにいくつかの技術が研究されている。その中でもっとも一般的な手法はユーザの購買履歴を元に推薦を行うものである。このような手法は協調フィルタリングと呼ばれ、amazon.comなどのサービスで実際に利用されている[神島 2007]。しかしながら協調フィルタリングをベースとした推薦システムには、購買履歴が十分な量集まらないと適切な推薦が行えないなどいくつかの問題点が指摘されている[神島 2007]。また、コンテンツの内容を一切考慮せずに推薦楽曲を決定するため、楽曲の特性が推薦結果に反映されないという問題がある。

一方、楽曲の音響信号を利用してユーザの楽曲発見を支援する手法として類似楽曲検索が挙げられる[Casey 2008]。この技術を用いることで、ユーザは楽曲を指定してその楽曲に類似した曲を獲得することができる。類似楽曲検索では、楽曲の音響信号をもとに複数の特徴量を抽出し、特徴ベクトルを作成する。楽曲と楽曲の類似の度合いは、特徴ベクトル同士の距離を事前に定めた類似度関数に基づいて定義する。しかしながらユーザの検索意図は多様であるため、あるひとつの類似度関

数を利用するだけでは、ユーザの求める検索結果が得られない場合がある。そのため、類似度関数を各ユーザごとにパーソナライズする手法が求められる。

本研究では、SNS上のタグから作成したランキングを教師データとする学習手法を利用し、インタラクティブに検索結果の改善を行う類似楽曲検索手法を提案する。提案手法により、ユーザは表示されたタグの中から自分の検索意図を反映したタグを選択することによって検索結果の改善を行えるようになる。

1.2 本研究の貢献

本研究では、タグから得られるランキングが、音響信号から得られるランキングよりも優れていると仮定し、ランキング学習手法によって、優れたランキングに検索結果が近づくように学習を行う。

類似楽曲検索システムの検索結果改善をランキング学習問題として定式化した研究はMcFeeらによって既に行われている[McFee 2010]。しかしながらこの研究は事前に学習を行っている。それに対して本研究ではユーザにクエリが与えられてから学習するという、より難しい課題に取り組み、実時間での応答で検索結果改善が可能であることを示した。また、学習の効果を、計算機実験、参加者実験の両方で確認しており、関連分野の研究者にとって有益な知見を提供するものである。

2 関連研究

本章では、まず本研究で取り組む課題と似た目的をもつ研究を紹介する。その後、次節で提案手法に取り入れられている手法について解説する。

2.1 似た課題に取り組んでいる研究

類似楽曲検索に関する研究の多くは楽曲の音響信号からの特徴抽出手法の改善を行っている。楽曲間の類似の度合いの良し悪しは特徴抽出がどの程度うまくいっているかに強く依存するためである。しかしながらそのようなアプローチに限界があることが Aucouturier らの研究 [Aucouturier 2004] によって示されている。この研究は当時利用されていた特徴抽出手法とそれらに必要なパラメータについて大規模な比較調査を行ったものである。この研究によって、いかにパラメータの調整を注意深く行ったとしてもそれ以上性能の向上しない限界点があること、また、どの楽曲にも似ていないと判断される曲やどの楽曲とも似ていると判断される曲が出てきてしまうことを発見した。今日、この問題は hubs and orphans 問題として知られている。このような限界を超えるために特徴抽出とは別の部分で問題の解決に取り組む研究が行われている。このような方向性の研究にはユーザのフィードバックを利用するものや、ソーシャルタグのような楽曲に付随する情報を利用するものがある。

斎藤らの研究 [斎藤 2011] では、楽曲の集合を可視化した上でユーザに表示し、適合する楽曲を選択してもらい、対話的遺伝的アルゴリズム手法を利用して次に表示する楽曲の選択を行っていくことでユーザの新たな楽曲発見の支援を試みている。また、類似楽曲検索に対して検索結果の個人化を試みた研究として小林らの研究 [?] があげられる。小林らは楽曲と楽曲の類似度をマハラノビス距離の形式で定義し、行列の値をパラメータとし、非適合とされる楽曲が適合とされる楽曲よりも類似度が低くなるよう学習を行うことにより改善を行う。しかしながらこれらの手法はユーザが適合不適合の情報を与える必要がある。実際に曲を聞いて評価しなければならぬためユーザの負担が大きくなる。

類似楽曲検索の分野においては、画像検索と異なりユーザが検索結果を評価するためには実際に楽曲を耳にする必要があるためフィードバックを与えることがより困難である。そこで SNS 上で得られるソーシャルタグを利用することが検討されている。Karydis らの研究 [Karydis 2013] ではタグを元に抽出した特徴量と、音響信号を元にした特徴量とで性能の比較をし、タグを元にした特徴量の方がより性能が高いことを示した。しかし、ソーシャルタグはすべての楽曲に付けられて

いるものではないので、タグのない楽曲に対してもタグによる検索を実現するような技術が必要となる。そのような技術として楽曲から自動的にタグを推定する研究が行われている [Turnbull 2008]。また、タグから得られる特徴量の特徴空間と、音響信号から得られる特徴空間との間を対応付ける関数を学習する手法が提案されている [Ioannis 2011]。しかしこれらの手法は、検索結果として得られるランキングそのものを最適化しているわけではない。ランキングそのものに着目して学習を行う手法は McFee らによって提案されており [McFee 2010]、本研究ともっとも関連性が強い。しかしながら先述の通り、実時間で学習を行っている、参加者実験による性能確認を行っているなどの違いがある。

2.2 提案手法に取り入れられている手法

2.2.1 楽曲の特徴抽出

類似楽曲検索では、楽曲を特徴ベクトルで表し、特徴ベクトル間のユークリッド距離の大小で類似の程度を評価する。そのため、どのような特徴量を用いるかは検索結果に大きな影響を与える。楽曲や音声の特徴量としては MFCC (Mel Frequency Cepstrum Coefficient) が広く用いられている。これは人間の聴覚特性に合わせた各周波数成分のスペクトル密度である。MFCC は短時間の信号から抽出するものなので、楽曲から MFCC を抽出する場合、楽曲の時間に比例した複数個の組が出力される。この組をどのようにまとめるかで様々な手法が提案されている。

本研究では Levy らによって提案された手法 [Levy 2006] を採用した。この手法では二つの異なる楽曲間の距離を次式のような形で表す。

$$D(p, q) = (\vec{\mu}_p - \vec{\mu}_q)^T \Sigma_{\mu}^{-1} (\vec{\mu}_p - \vec{\mu}_q) + (\vec{s}_p - \vec{s}_q)^T \Sigma_s^{-1} (\vec{s}_p - \vec{s}_q) \quad (1)$$

ここで、 $\vec{\mu}_p$, $\vec{\mu}_q$ はそれぞれ楽曲 p , q の MFCC の各成分の平均ベクトルであり、 \vec{s}_p , \vec{s}_q は各成分の標準偏差である。 Σ_{μ} , Σ_s はデータセットの楽曲を表す特徴ベクトル $\vec{\mu}$, \vec{s} の共分散行列である。この手法は他の手法に比べ、特徴量の抽出、比較が高速に行えるという性質がある。我々はこのような性質が検索システムに望ましいと考え、この手法の採用を決断した。

2.2.2 タグの特徴抽出

本研究ではタグの情報を利用する。タグは、各タグを一つの単語とみなすことで Bag-of-Words のベクトルに変換できる。Bag-of-Words 表現のベクトルに対す

る重み付けとして TF-IDF がもっとも利用されているが本研究では IDF のみを各タグに対する重み付けとして採用した。本研究では、楽曲はその人気によってタグが付けられた回数の差が非常に大きく、 tf 値が大きいタグほどその楽曲の特徴を適切に反映しているとはいえないためである。

female vocalist と female vocalists のように、タグの中には意味的に非常に近いものがある。前述した Bag-of-Words の表現では、これは完全に別々の単語としてみなされてしまう。単語と単語の共起関係を考慮することでこれらの単語の間関係を考慮することができればより望ましい検索結果を導く事ができる。そのような要求に答える技術が Latent semantic Indexing(LSI)[Scott 1988] である。LSI では、高頻度で共起する単語が同一の要素に対応するよう特徴ベクトルの次元削減を行う。機械学習の分野で特徴ベクトルの次元を削減する場合、もっともよく用いられる手法が主成分分析である。しかしながら主成分分析は固有値分解を伴うため、単語-文書行列のような非正方行列には適用できない。そこで LSI では主成分分析の代わりに特異値分解を行うことで次元削減を実現する。

文書数が N 個で、それらの文書に含まれる単語の種類が M 種類であるとき、その単語-文書行列 C は大きさが $M \times N$ となる。この行列では一つの列が一つの文書を表す。このとき C は次式 (2) のように分解できる。

$$C = U\Sigma V^T \quad (2)$$

ここで U は CC^T の固有ベクトルを、対応する固有値が大きい順に並べた行列である。 V は C^TC の固有ベクトルを同様に並べた行列である。このとき Σ は (i, i) 成分が、行列 C の i 番目に大きい特異値であり、そのほかの成分がすべて 0 となる対角行列である。行列 C が式 (2) のように分解可能であるとき、 k 次元の低階数近似行列 C_k は式 (3) で得られる。

$$C_k = U\Sigma_k V^T \quad (3)$$

ここで Σ_k は Σ の $j > k$ であるすべての (j, j) 成分を 0 に置き換えた行列である。

本研究では $k = 200$ として利用した。

2.2.3 ランキング学習

特徴ベクトルで表されたデータセットとクエリについて、理想的なランキングが明らかであるときに検索結果をその理想的なランキングに近づけようとする手法としてランキング学習と呼ばれるものがある [Li 2009]。これは何らかの方法で理想的なランキングを機械学習の教師データとして扱えるよう変換し、教師あり学習手法で検索結果の改善を行う。理想的なランキングを

どのように教師データとして利用するかで、いくつかの枠組みが存在する。本論文では、分類手法を用いる枠組みを採用したので、それについて説明する。

あるデータの特徴ベクトルを \vec{v} とする。そして、あるデータの評価値が $\vec{w}^T \vec{v}$ のような特徴ベクトルの重みつき和で表されるとする。ここで \vec{w} は重みベクトルを表す。ここで理想的なランキングにおいて順位が i 番目のデータの特徴ベクトルを \vec{v}_i 、 j 番目の特徴ベクトルを \vec{v}_j とすると $i < j$ である i, j の組に対して次式の制約を可能な限り満たす重みベクトルを学習する。

$$\vec{w}^T \vec{v}_i > \vec{w}^T \vec{v}_j \quad (4)$$

この重みを学習する問題は $\vec{w}^T (\vec{v}_i - \vec{v}_j) > 0$ と変換すると二値分類問題と考えることができる。重みの学習には分類問題に用いられるパーセプトロンや SVM のような手法が利用可能である。本研究ではパラメータを調整する必要がない Passive Aggressive(PA) 法 [Koby 2006] を中心に使用するアルゴリズムの検討を行った。法については次節で学習手法を紹介する際、詳しく解説する。

3 提案手法

3.1 提案手法の利用を想定している局面

提案手法がどのような局面での利用を想定しているのかを述べる。提案手法は、ユーザが検索したいクエリ楽曲と、クエリ楽曲に対応するタグの情報が入力として必要になる。タグの情報と音響信号の情報を比較したとき、どちらの方がより類似楽曲検索の役に立つかは先行研究で調査されている [?][Ioannis 2011]。これらの研究はタグの情報の方がより精度の良い検索結果をもたらすことを示している。もし検索対象のデータセットに、楽曲そのものだけでなくタグも含まれるのであれば、そのタグを用いて検索することができる。しかしながら、タグの情報は SNS 等から得られるごく一部の楽曲にしかつけられていない。そのため、データセットを、楽曲の音響信号のみから検索しなければならない状況が発生する。本研究はそのような状況で利用される。つまり、検索したい対象となるデータは楽曲のみしか用意できないが、別の楽曲の集まりについてはタグと楽曲の両方を用意できるという状況である。提案手法では、楽曲の音響信号をから得られる特徴ベクトルを元に類似度を算出する際、ある重みパラメータを用いて類似度の算出の仕方を修正することで検索結果の改善を実現する。次節以降でシステムの詳細を記述する。

3.2 教師データ

提案手法ではランキング学習の手法を用いて類似度関数の重みを調整する。ランキング学習を行うためには教師データとなる理想的なランキングが必要となる。本手法では楽曲につけられているタグをもとに得られる検索結果を教師データとして利用した。タグから類似楽曲を得る手続きについて説明する。まず、事前にデータセット中の楽曲について、そのタグをもとにBag-of-Wordsで表現されたベクトルを作成し、関連研究の項で解説した IDF による重み付け、LSIによる次元圧縮を行って 200 次元の特徴ベクトル \vec{f} に変換した。そしてこのベクトルを記録した。検索する際は、まず、クエリとして与えられたタグに対し同様の処理を行ってベクトル \vec{f}_q を作成する。その後コサイン類似度でデータセット中の楽曲との類似度を算出する。

コサイン類似度は大きな値であれば大きな値であるほどより類似していることを表す指標であるので、コサイン類似度の値が大きい順にソートすることによって、類似した楽曲のランキングを作成することができる。

3.3 学習手法

教師データとなるランキングが得られたならば次の課題は重みの学習である。本研究ではオンライン学習手法の一つである Passive-Agressive 法 [Koby 2006] を改良した手法で重みの学習を行った。この改良法の紹介をする前、まず PA 法を本研究に応用するときどのようにすべきかの解説を行う。次に改良法との変更点を示す。

3.3.1 Passive-Agreessive

クエリとなる楽曲を q 、タグをもとに検索して得られる理想的なランキングにおいてある順位である楽曲をその順位を表す記号 i, j を用いて表現する。今、 $i < j$ であるとき、検索結果を理想的なランキングに近づけるためには式 (5) の制約ができる限り満たされるように重み \vec{w} の学習を行えばよい。

$$D_n = \vec{w}^T \vec{v}_j > D_p + 1 = \vec{w}^T \vec{v}_i + 1 \quad (5)$$

ここで D_n は楽曲 j と楽曲 q の距離を表し、 D_p は楽曲 i と楽曲 q の間の距離を表す。クエリ q との距離が小さい楽曲ほどより似ていると判断される。 \vec{v}_p, \vec{v}_n はそれぞれクエリとの差の大きさを表すベクトルであり、楽曲の音響信号より得られる特徴ベクトル \vec{x}_i, \vec{x}_j を用いて次式 (6)(7) で定義される。

$$\vec{v}_p = (\vec{x}_i - \vec{x}_q) \circ (\vec{x}_i - \vec{x}_q) \quad (6)$$

$$\vec{v}_n = (\vec{x}_j - \vec{x}_q) \circ (\vec{x}_j - \vec{x}_q) \quad (7)$$

\circ はアダマール積を表し、成分ごとに積を取る処理を表す。制約 (5) は、より上位にランキングされるべき楽曲 i と q との間の距離が、より下位にランキングされるべき楽曲 j と q との間の距離よりも 1 以上小さくなることを意味している。

ある楽曲 i, j に対してこの制約を満たす重みは無数にある。しかしながら、制約が満たされる中で、重みの更新量を最小にしようとするとき、その重みの値はただ一つに定まる。このことは次の制約付き最小化問題 (8) を解くことで求められる。

minimize

$$O(\vec{w}) = \frac{1}{2} |\vec{w} - \vec{w}_t|^2 \quad (8a)$$

subject to

$$l_t(\vec{w}, \vec{v}_p, \vec{v}_n) = 0 \quad (8b)$$

ここで l_t は式 (9) で定義される。

$$l_t(\vec{w}, \vec{v}_p, \vec{v}_n) = \begin{cases} 0 & (\text{if } D_n - D_p \geq 1), \\ 1 - (D_n - D_p) & (\text{otherwise}), \end{cases} \quad (9)$$

このとき最適化問題を解くと次の更新式が得られる。

$$\vec{w}_{t+1} = \vec{w}_t + \tau_t (\vec{v}_n - \vec{v}_p) \quad \text{where } \tau_t = \frac{l_t}{|v_n - v_p|^2} \quad (10)$$

以上のアルゴリズムをまとめると Algorithm 1 となる。ここで d は特徴ベクトルの次元数を表す。ones(d) は値がすべて 1 である d 次元の列ベクトルを出力する関数である。

3.3.2 提案手法

前節で説明したアルゴリズムを本研究に適用するといくつかの問題が生じる。それぞれの問題とその解決策を次に示す。

まず、もっとも大きな問題点は計算時間である。楽曲が N 個あるデータセットから順序を考慮せず 2 つ取り出した組み合わせは $\frac{N(N-1)}{2}$ 通りとなる。この組み合わせすべてを学習に利用すると計算時間のオーダーは $O(N^2)$ となり実時間での応答が難しくなる。これは検索システムとしてふさわしくない。このような問題に対処するために提案手法では学習に利用する楽曲を、ランキング上位 k 個と下位 m から得られる組に限定した。これはランキングで順位がそれほど変わらない楽

Algorithm 1 Passive Aggressive

Require: \vec{x}_q, R

```

 $\vec{w} = \text{ones}(d)$ 
for  $i = 0 : N - 1$  do
  for  $j = i : N - 1$  do
     $\vec{v}_p = (\vec{x}_i - \vec{x}_q) \circ (\vec{x}_i - \vec{x}_q)$ 
     $\vec{v}_n = (\vec{x}_j - \vec{x}_q) \circ (\vec{x}_j - \vec{x}_q)$ 
     $D_p = \vec{w}^T \vec{v}_p$ 
     $D_n = \vec{w}^T \vec{v}_n$ 
    if  $D_n - D_p < 1$  then
       $l_t = 1 - (D_n - D_p)$ 
       $\vec{v} = (\vec{v}_n - \vec{v}_p)$ 
       $s = \vec{v}^T \vec{v}$ 
       $\vec{\Delta} = \frac{l_t}{s} \vec{v}$ 
       $\vec{w} = \vec{w} + \vec{\Delta}$ 
    end if
  end for
end for

```

Ensure: \vec{w}

曲同士を比べても汎化性能の向上に繋がる適切な学習が行えない、という考えに基づいている。このように学習データを限定することで学習にかかる計算時間は $O(k \cdot m)$ となる。このように学習データを制限する手続きを以後 Algorithm 2 とする。本研究では $k = 50$, $m = 200$ と設定した。

二つ目の問題点は汎化誤差に関するものである。PA法は与えられたデータに対して必ず正常に分類できるように学習するものなので、はずれ値が与えられると、それを正常に分類できるように学習することによって汎化誤差が悪化する可能性がある。この問題に対処するために、学習によって重みが更新されるたびにその値を保存し、出力の際にそれらの成分ごとの平均を取るという処理を考案した。これは [Carvalho 2006] の手法から着想を得たものである。このような手続きを以後 Algorithm 3 とする。

三つ目の問題点は、特徴量の次元数が低いと調整できる重みが限られるというものである。この問題に対処するため、特徴量をそのまま利用するのではなく、代わりに各特徴量の積を利用するという処理を考案した。これは SVM の多項式カーネルから着想を得たものである。この工夫を導入するとき、解くべき最適化問題は次のようになる。

minimize

$$O(\mathbf{W}) = \frac{1}{2} |\mathbf{W} - \mathbf{W}_t|^2 \quad (11a)$$

subject to

$$l_t(\mathbf{W}, \vec{v}_p, \vec{v}_n) = 0 \quad (11b)$$

ここで l_t は式 (9) であるが、クエリとの距離 D_p , D_n の定義が異なり、式 (12)(13) のように定義される。

$$\vec{v}_p = \text{abs}(\vec{x}_p - \vec{x}_q) \quad (12a)$$

$$D_p = \text{sum}(\mathbf{W} \circ \vec{v}_p \vec{v}_p^T) \quad (12b)$$

$$\vec{v}_n = \text{abs}(\vec{x}_n - \vec{x}_q) \quad (13a)$$

$$D_n = \text{sum}(\mathbf{W} \circ \vec{v}_n \vec{v}_n^T) \quad (13b)$$

この工夫を取り入れた手法を Algorithm 4 とする。

また、考案したすべての工夫を取り入れた場合、手続きは Algorithm 5 となる。ここで S_p^k は LSI より得られるランキングの上位 k 個の楽曲を集めた集合、 S_n^m は下位 m 個の楽曲を集めた集合を表し、 $S_w.append(\vec{w})$ は配列 S_w に \vec{w} を追加する処理、 $\text{mean}(S_w)$ は S_w に含まれる重みベクトル \vec{w} の各成分の平均を取る処理を表す。 $\text{diag}(d)$ は d 次元で対角成分がすべて 1 である対角行列を出力する関数を表す。 \mathbf{T}_r は非零成分が 1 である大きさ $(d \times d)$ の上三角行列であり、 \mathbf{V} の非対角成分が二重に足し合わされるのを防ぐ役割を担っている。

Algorithm 5 Proposed Method

Require: \vec{x}_q, S_p^k, S_n^m

```

 $\mathbf{W} = \text{diag}(d)$ 
 $S_w = []$ 
for  $\vec{x}_p$  in  $S_p^k$  do
  for  $\vec{x}_n$  in  $S_n^m$  do
     $\vec{v}_p = \text{abs}(\vec{x}_p - \vec{x}_q)$ 
     $\vec{v}_n = \text{abs}(\vec{x}_n - \vec{x}_q)$ 
     $D_p = \text{sum}(\mathbf{W} \circ \vec{v}_p \vec{v}_p^T)$ 
     $D_n = \text{sum}(\mathbf{W} \circ \vec{v}_n \vec{v}_n^T)$ 
    if  $D_n - D_p < 1$  then
       $l_t = 1 - (D_n - D_p)$ 
       $\mathbf{V} = (\vec{v}_n \vec{v}_n^T - \vec{v}_p \vec{v}_p^T) \circ \mathbf{T}_r$ 
       $s = \text{sum}(\mathbf{V} \circ \mathbf{V})$ 
       $\mathbf{\Delta} = \frac{l_t}{s} \mathbf{V}$ 
       $\mathbf{W} = \mathbf{W} + \mathbf{\Delta}$ 
       $S_w.append(\mathbf{W})$ 
    end if
  end for
end for
 $\mathbf{W}_{output} = \text{mean}(S_w)$ 

```

Ensure: \mathbf{W}_{output}

我々はこれら Algorithm 2, 3, 4, 5 について、それぞれ計算機実験により汎化性能を確認した。また、もっともすぐれた性能を示したアルゴリズムを用いて参加者実験を行った。実験の詳細について次節に示す。

4 実験

提案手法の性能を確認するためにふたつの実験を行った。一つ目の実験は学習による汎化性能を確認する計算機実験である。そしてもう一つは実際にユーザの感性和照らし合わせても望ましい検索結果が得られることを確認する参加者実験である。まず、実験に利用したデータセットの構築について述べ、その後それぞれの実験の詳細を記述する。

4.1 実験に用いるデータ

本研究の実験は参加者による評価を含む実験であるため、楽曲のタグ、特徴ベクトルのほかに、楽曲そのものが必要となる。そのようなデータセットは筆者の知る限りでは存在しない。そこで、我々は本実験のために last.fm¹、7digital² の二つの API を利用してデータセットの構築を行う。last.fm は音楽に特化した SNS であり、サービスを通して得られた楽曲のメタデータを公開している。本実験では API を利用し、楽曲 3046 曲のメタデータを獲得した。このメタデータ内にはタイトル、アーティスト名などの一般的な情報のほかに各楽曲につけられたタグの情報を含んでいる。タグそれぞれには、そのタグが対応する楽曲に対して何度付けられたかを示す数が付けられている。本研究では、不正確なタグを除外するために、一つの楽曲に対して前述した数が大きいものから上位 20 個のタグのみを採用した。データに含まれる楽曲の再生可能な音響信号情報を獲得するために 7digital で公開されている API を利用した。7digital API を利用することで各楽曲ごとに約 30 秒の長さの試聴用データを入手できる。本実験ではこの試聴用データをもとに、特徴抽出および参加者実験を行う。特徴抽出は先述のとおり先行研究 [Levy 2006] で用いられている手法を利用した。

4.2 計算機実験

一つ目の実験では、テストデータを学習データとテストデータの半分に分け、汎化性能を確認した。クエリとしては、データセット中の楽曲とタグを利用した。学習の際は、学習データについて LSI より得られるランキングの上位 50 曲と下位 200 曲から得られる順序関係の組み合わせ 10000 個を教師データとして採用した。評価指標としては、テストデータの上位 50 曲と下位 200 曲から得られる順序関係の何割が満たされているかで評価した。はじめの 50 個のサンプルで学習時間を計測したところ平均学習時間は 0.502 秒だった。実験

のサンプル数は 3046 である。実験結果を図 1 に示す。Baseline は学習なしで検索した結果を表している。図 1 のエラーバーは標準誤差を表している。図から Alg 5 がもっとも汎化性能において優れていることが明らかになった。

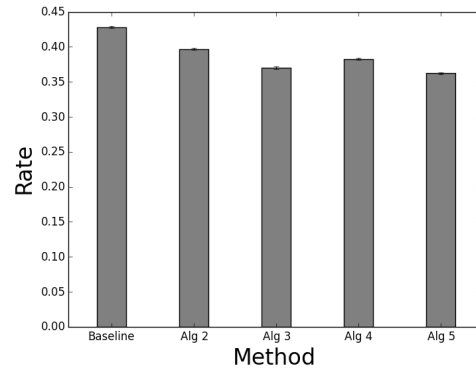


図 1: 計算機実験の結果

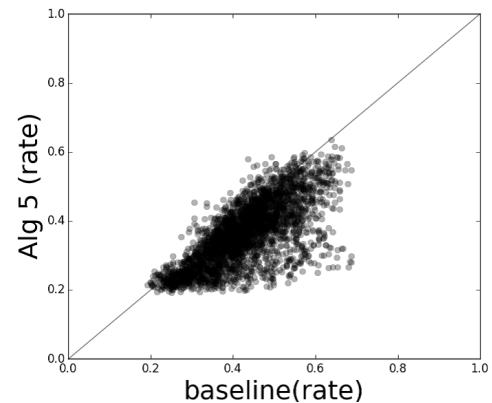


図 2: Baseline との比較

統計量だけでなく、一曲単位で見たとき、Baseline に比べ提案手法が優れているかは図 2 で確認できる。図 2 は学習の前後での、破られる制約の割合をプロットしたものである。左下から右上にの線は境界線を表しており、この境界線より右下にプロットされているならば、学習後の方が破られる制約の割合が低いこと、すなわち学習によってより望ましいランキングに近づいていることを示している。図 2 より多くのクエリで、学習後の結果が改善していることが明らかになった。

¹The Last.fm Dataset — Million Song Dataset <http://labrosa.ee.columbia.edu/millionsong/lastfm>

²7digital Developers <http://developer.7digital.com/>

4.3 参加者実験

4.3.1 実験手続き

提案手法に、ユーザに実際に聞いて望ましいと感じる結果が得られていることを確認するため参加者実験を行った。実験参加者は男性 15 名 (平均年齢 25, 最小 23, 最大 32, 標準偏差 2.68) である。また、参加者には実験報酬として 1500 円相当の筆記用具を贈与した。実験の手続きを説明する。参加者には各々が好む楽曲を用意してもらった。実験ではその楽曲をシステムに入力してもらい、事前にこちらが用意したタグの中から、楽曲を表すのにふさわしいと思うものを任意の数 (ただし一つ以上) 指定してもらった。用意したタグを表 1 に示す。これらのタグは実験者が楽曲を選ぶのに役に立つと判断して決定したものである。

表 1: 選択肢として提示したタグ

Tag	
instrumental	relax
rock	classic
pop	male vocalist
jazz	female vocalist
Ballad	beautiful
metal	guitar
alternative	piano
electronic	sad
dance	happy

システムにタグと楽曲が入力されると、学習なしで検索した結果 (Baseline) と、Algorithm 5 で学習をした検索結果からそれぞれ上位 5 曲が無作為な順番で図 3 のように表示される。楽曲名やアーティスト名のようなメタ情報にユーザが惑わされず、楽曲そのものについて評価できるようにするため、検索された楽曲に関する情報は表示されない。ユーザは UI を通して、楽曲の再生、一時停止、停止を行うことができる。検索結果はシャッフルされてから表示されるので、提案手法より得られる楽曲を高く評価するといった恣意的な評価を行うことはできないようになっている。

参加者にはそれぞれの楽曲を聞いてもらい、その楽曲が検索結果としてどの程度ふさわしいかを評価してもらった。評価値は 0 から 1 の間の得点を 0.1 点刻みで付けてもらった。参加者より得られた評価値を元に、学習前後の結果に対して、ランキングの評価指標である nDCG(normalized Discount Cumulative Gain) [Jarvelin 2000] を算出した。

nDCG は情報検索の分野で用いられる指標で検索結果のランキングが理想的なランキングにどれぐらい近

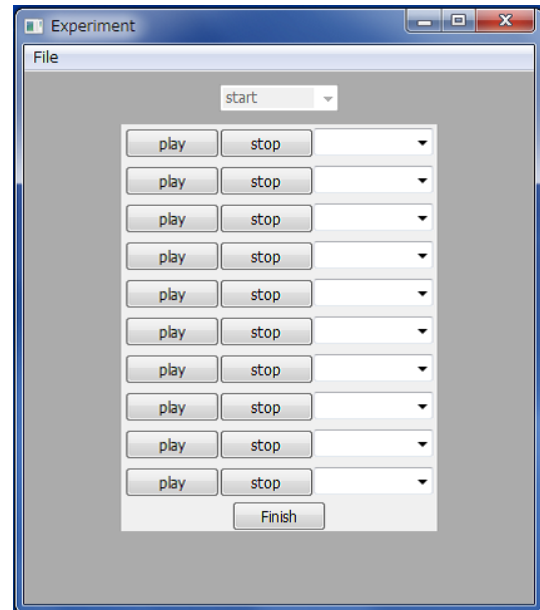


図 3: システムのユーザーインターフェース

いかをあらわす指標である。nDCG には、ユーザのアイテムに対する選好順序の情報を必要とせず、各アイテムについての評価さえ得られれば算出できるというメリットがある。本研究では、楽曲集合に対して選好の順序を記述させるよりも、各楽曲をそれぞれ評価する方がユーザにとって負担が少ないと考え、評価指標として nDCG を採用した。

4.3.2 実験結果

実験結果を図 4 に示す。図 4 のエラーバーは標準誤差を表す。図より、参加者が提案手法から得られる楽曲をより高く評価する傾向にあることが確認された。

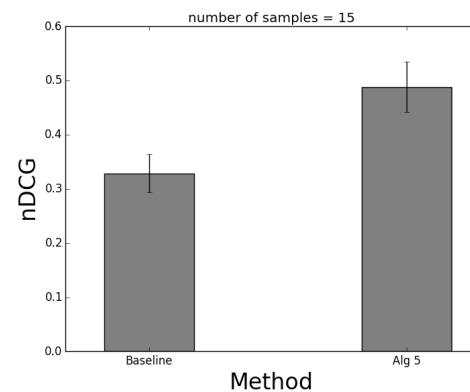


図 4: 参加者実験結果 棒グラフ

5 まとめと今後の課題

本論文では、類似楽曲検索において、ユーザがインタラクティブに検索結果の改善を行える方法を考案した。計算機実験の結果、学習によって重みを修正することで、出力をタグより得られるランキングに近づけることができることが確認された。また、参加者実験によって、このように学習して得られるランキングは、学習前のランキングに比べ、ユーザにとってより好ましいものであることがわかった。

一方提案手法は実際にユーザからクエリを受け取ってから学習を行うために、応答に時間がかかるという問題が残っている。また、本研究ではあらかじめ実験者が選択すべきタグの候補を決定していたが、実際にどのタグを用いると検索結果に大きな影響を与えるのかは未調査である。さらに有益なシステムのためには、今後このような課題に取り組むことが求められる。

参考文献

- [神寫 2007] 神寫 敏弘 : 推薦システムのアルゴリズム (1), 人工知能学会誌 22 巻 6 号 (2007)
- [Casey 2008] Casey, Michael A., et al. : Content-based music information retrieval: Current directions and future challenges, Proceedings of the IEEE 96.4 668-696 (2008)
- [Li 2009] Li, Hang. : Learning to rank, Tutorial given at ACL-IJCNLP, August. research.microsoft.com/enu/people/hangli/li-acl-ijcnlp-2009-tutorial.pdf (2009).
- [Jarvelin 2000] Jarvelin, K., Kekalainen, J. : IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 41-48). ACM. (2000)
- [斉藤 2011] 斉藤 優理 伊藤 貴之 : 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース, 2011 年度人工知能学会全国大会 (2011)
- [Levy 2006] Mark Levy, Mark Sandler : Lightweight measures for timbral similarity of musical audio, AMCM '06 Proceedings of the 1st ACM workshop on Audio and music computing multimedia Pages 27-36
- [Ioannis 2011] Karydis, Ioannis ; Nanopoulos, Alexandros : Audio-to-Tag mapping: A novel approach for music similarity computation, Multimedia and Expo (ICME), 2011 IEEE International Conference on
- [Carvalho 2006] Vitor R. Carvalho, William W. Cohen : Single-Pass Online Learning: Performance, Voting Schemes and Online Feature Selection, KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining Pages 548-553
- [Koby 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, Yoram Singer : Online Passive-Aggressive Algorithms, Journal The Journal of Machine Learning Research archive Volume 7, 12/1/2006 Pages 551-585
- [Scott 1988] Scott Deerwester and Susan T. Dumais and George W. Furnas and Thomas K. Landauer and Richard Harshman : Indexing by latent semantic analysis, JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE, Volume 41, Issue 6, pages 391-407, September 1990
- [McFee 2010] Brian McFee, Luke Barrington, Gert Lanckriet : LEARNING SIMILARITY FROM COLLABORATIVE FILTERS, 2010 International Society for Music Information Retrieval.
- [Karydis 2013] Ioannis Karydis, Katia Lida Kermanidis, Spyros Sioutas, Lazaros Iliadis : Comparing content and context based similarity for musical data, Neurocomputing 107 (2013) 6976
- [Aucouturier 2004] Jean-Julien Aucouturier, Francois Pachet : Improving Timbre Similarity: How high's the sky? : Journal of Negative Results in Speech and Audio Sciences (2004)
- [Turnbull 2008] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet : Semantic Annotation and Retrieval of Music and Sound Effects, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 16, NO. 2, FEBRUARY 2008