

レコメンデーションエージェントによる多機能検索モデル

＊徳本 真平，タネヴ イヴァン，下原 勝憲

同志社大学大学院工学研究科

〒610-0394 京都府 京田辺市 多々羅都谷 1-3 香知館 219

E-mail : {tokumoto, itanev, kshimoha}@sil.doshisha.ac.jp

Abstract: 本研究では、「レコメンデーションエージェント」というシステムを作り、その正当性を検証する。複数用意したエージェントにレコメンドの基準を独自に持たせ、各エージェントがユーザの嗜好を分析し、ユーザにレコメンドを行う。この検索モデルで得られた結果をユーザに評価してもらい、その結果から、レコメンデーションシステムの改善案を考察し、発展性や展望を考える。

1. はじめに

音楽鑑賞は、世界中で広く親しまれている趣味の一つである。自分の嗜好にあった曲やアーティストを発見することは、この上ない喜びである。しかし、無数に存在している曲やアーティストの中から好みのものを探すことは、時間がかかる作業である。そこで、ユーザの嗜好を分析し、その人の好みにあったものを紹介する、「レコメンデーションシステム」というサービスが存在する。音楽においても、このサービスは利用されており、情報収集の大きな手助けとなっている。しかしながら、レコメンデーションシステムを提供しているサイトやソフトが多様化していくにつれ、「どのレコメンデーションシステムが自分にあっているのか」という問題が生じる。同じ内容を入力しても、レコメンデーションシステムを提供しているサイトやソフトによって、異なる結果が得られることが多い。

そこで、レコメンデーションシステムがユーザに合わせて仕様を変えることができるのではないかと考えた。本研究の目的は、レコメンデーションシステムがユーザに合わせて変化できるような検索モデルを作り、どのような方法が有効かを探し、レコメンデーションシステムの改善策を提案し、今後の発展性を考察することである。

2. 多機能検索のためのレコメンデーション

多機能検索のためのレコメンデーションシステムの基本的な考え方は次のとおりである。「あるユーザと近い嗜好と持つユーザのお気に入りには、そのユーザも気に入る可能性が高い」という推測を元に、レコメンドする基準を変えながら、嗜好の近い人を検索し、そのユーザから曲を紹介してもらう、という検索モデルを実装する。推薦アルゴリズムには、内容ベースフィルタリングと協調フィルタリングがあるが、本研究は、協調フィルタリングによる推薦システムの改良である。

2.1. 嗜好の評価方法

推薦する曲は、比較するユーザ間の嗜好がある程度合致しており、かつユーザにとって未知の曲でなければならない。この条件を満たすためには、ユーザの嗜好を定量化する必要がある。

ユーザの嗜好を定量化するための手段として、内藤^[1]の「ユーザの嗜好情報を利用した音楽検索モデルの提案」を参考に、iTunes で使用しているプレイリストを提示してもらう方法をとった。iTunes のプレイリストに含まれている情報には曲名やアーティスト名の他に、「再生回数」があり、そのユーザがどの曲を何度聞いたのかがわかるようになっている。これを利用してユーザの嗜好を数値化する。

2.2. レコメンドする曲の決定

ユーザの嗜好を定量化した後は、ユーザ同士の嗜好を比較し、各ユーザから見て最も嗜好の近いユーザを検索し、そのユーザのプレイリストから音楽をレコメンドする、という方法をとる。これは、「あるユーザと近い嗜好を持つユーザのお気に入りには、そのユーザも気に入る可能性が高い。」という推測に基づいている。

2.3. レコメンデーションエージェント概要

異なる評価関数を持つ、次の4つのエージェントを用意する。

● 偏差値エージェント

再生回数から、他の曲と比べてどの程度好きなのかを判断するために、偏差値を算出する。偏差値は、ユーザ間で一致した曲がプレイリスト全体の中での位置づけを示したものである。そこから、ユーザ間の嗜好がどれほど似通っているかを判断するための類似度を式1、式2、式3により計算する。類似度は、一致した曲の偏差値からプレイリスト間の類似性を算出したものである。

$$S = \frac{\frac{Sx}{Nx} + \frac{Sy}{Ny}}{2} \quad \dots(\text{式1})$$

$$Sx = \sum_{x\alpha=y\beta} D\alpha, Nx = \sum_{\alpha=1}^n D\alpha \quad \dots(\text{式2})$$

$$Sy = \sum_{x\alpha=y\beta} D\beta, Ny = \sum_{\beta=1}^n D\beta \quad \dots(\text{式3})$$

ユーザ x とユーザ y を比較する場合、ユーザ x の曲の偏差値の合計が Nx であり、ユーザ x のプレイリストからユーザ y のプレイリストと一致する曲の偏差値の合計が Sx である。 Ny , Sy も同様に算出する。類似度 S は最大値1、最小値0をとり、数字が大きいほど類似性がある。すなわち、 S が1に近いユーザほど嗜好が近いと言える。

● 一致数エージェント

基本的な方法論は偏差値エージェントと同じであるが、このエージェントは再生回数による差異を考慮しない。これには、偏差値による判断が正しいかどうかを判別する実験的な目的がある。

● 代表偏差値エージェント

基本的な方法論は偏差値エージェントと同じであるが、このエージェントはアーティスト毎に偏差値を算出する。そのユーザのプレイリストの中で、同じアーティストの曲を比較し、最も再生回数の高い曲を代表として選択する。偏差値の計算は代表の曲のみを使用し、類似度の算出はアーティスト毎に行う。

● 時間偏差値エージェント

基本的な方法論は偏差値エージェントと同じであるが、このエージェントは曲ごとに時間的な差異を加味する。再生回数を、最初にプレイリストに追加した曲の日付から集計日までの日数を乗算し、

各曲を追加した日付から最終日までの日数を除算する．これによって、プレイリストに乗っている期間が長い曲は少なく評価され、期間が短い曲は多く評価される．

3. システムの詳細

3.1. データベース

ユーザ間のプレイリスト内の曲のうち、どれが一致しているかを判断する必要があるため、「曲名」及び「アーティスト名」はユーザ間のプレイリスト内で共有する必要がある．本研究では、Paradox を用いて、2つの関係性のあるデータベースを作成した．

3.2. ファイルとシステム構造

取り扱うファイルは、XML である．iTunes のプレイリストは XML として簡単にエクスポートが可能で、テキスト化することで、Excel 等で簡単に加工が可能である．テキストから Paradox データベースの操作、類似度の計算までを簡単に操作できるよう、Delphi でシステムを開発した．

4. 実験

4.1. 実験の目的と評価方法

この実験の目的は、検索モデルにおけるレコメンデーションエージェントの評価を行うことである．評価方法は、プレイリストを提供してくれたユーザにアンケートに答えてもらうことで行った．

4.2. 実験の検証

● 検証(1) システムの正当性の検証

各エージェントによる判断において、最も嗜好が近いユーザから、重複していない再生回数上位の 10 曲をレコメンドする．レコメンドされた曲が嗜好に合った曲であるかどうかを質問し、システムの正当性を検証する．評価方法はそれぞれの曲に対して 5 段階で評価してもらう．

● 検証(2) 類似度の基準の検証

最も嗜好が近いユーザと、最も嗜好が遠いユーザで各 10 曲レコメンドし、5 段階で評価してもらい、その差を比較する．

● 検証(3) エージェントの特徴の検証

どのエージェントが最も嗜好にあったレコメンドをしたかを質問し、傾向を分析する．

5. 結果と考察

5.1. 実験結果

7 人の被験者に対して実験を行った．4 つのエージェントは、全ての実験において、「最も近いユーザ」及び「最も遠いユーザ」として同一のユーザを示すという結果となった．それぞれが全く同じユーザを示すという結果になった．したがって、検証(1)と検証(2)については、エージェントを統一した結果を参照する．

● 検証(1) システムの正当性の検証結果

7 人各 10 曲の合計 70 曲の評価を、図 1 に示す．最も嗜好が近いユーザからのレコメンドは、7 人各 10 曲の合計 70 曲中、「非常に気に入った」が 8 曲、「気に入った」が 12 曲、「どちらでもない」が 16 曲、「あまり気に入らなかった」が 21 曲、「気に入らなかった」が 13 曲となった．

● 検証(2) 類似度の基準の検証結果

7 人各 10 曲の合計 70 曲の評価を、図 2 に示す．最も嗜好が近いユーザからの結果と比較すると、「非常に気に入った」が 6 曲少なく、「気に入った」が 7 曲少なく、「どちらでもない」が 1 曲多く、

「あまり気に入らなかった」が6曲多く、「気に入らなかった」が6曲多い、という結果になった。

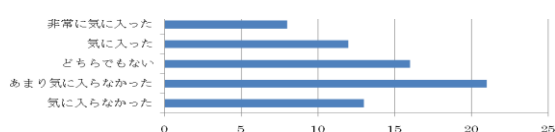


図 1 検証(1)

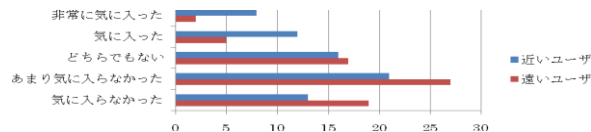


図 2 検証(2)

● 検証(3) エージェントの特徴の検証結果

検証(3)については、4つのエージェントが全ての実験において、「最も近いユーザ」及び「最も遠いユーザ」として同一のユーザを示したため、検証不可能となった。

5.2. 考察

検証(1), (2)より、最も嗜好が近いとされるユーザからの Recommend であるにも関わらず、「あまり気に入らなかった」、「気に入らなかった」が多かったが、最も嗜好が遠いユーザよりも、気に入ったという評価を下す傾向にある、ということがわかる。Recommend の精度は低いが、各エージェントの嗜好の近さを計る評価関数は適切であると考えられる。検証(3)については、各エージェントが似た値を示したために検証不可能となった。一方、ユーザによっては類似度の順位がエージェントによって違う結果も得られた。ユーザの類似度をランク付けし、被験者にその結果を評価してもらうような実験が可能ならば、エージェント間の差異を評価し、検証できると考えられる。今回用意した4つのエージェントは、結果の差別化を図るにあたっては、近すぎる値を示すということがわかった。

5.3. 問題点と改善案

検証(3)の結果を得るためには、各エージェントがそれぞれ違ったユーザを示すようにしなければならない。その為には、フィルタリングを変更するエージェントを用意することが必要である。フィルタリングには、協調フィルタリングの他に、項目に付随する特徴を元に Recommend する手法である内容ベースフィルタリングがある⁴。もう一つの問題点は、被験者とサンプル数の不足である。被験者には、多くの質問に答えてもらう予定であったため、時間的な問題から、少人数で実験を行った。この問題を改善する方法は、被験者とは別に、プレイリストを提供してもらうユーザを多数用意することである。

6. おわりに

「嗜好の近いユーザ」及び「嗜好の遠いユーザ」を導き出す基準としては妥当であるという結果は得られたものの、エージェント間の差異や、エージェントごとの結果の特徴などは検証できなかった。エージェントの挙動、評価方法、実験に多くの改善の余地があるという結果になった。理想的には、あるエージェントは曲と曲の関連性を重視し、あるエージェントは曲調の近いものを重視する、といったように、楽曲に付随する様々な属性を参照して Recommend を行う、というものが考えられる。

参考文献

- [1] 内藤正英, ユーザの嗜好情報を利用した音楽検索モデルの提案, 2007 年度同志社大学工学部情報システムデザイン学科卒業論文, 2008.
- [2] 鈴岡裕也, 2つのフィルタリングを用いたハイブリッド型 Recommendation System, 2008 年度同志社大学工学部情報システムデザイン学科卒業論文, 2009.