

決定木を用いた 内容に基づく音楽情報フィルタリング

土方 嘉徳[†] , 岩濱 数宏[†] , 西田 正吾[†]

[†] 大阪大学大学院 基礎工学研究科

インターネット上で流通する情報の中から、ユーザの好みに適合するものを推薦するサービスとして情報フィルタリングがある。近年は、テキストで記述された情報だけでなく音楽などのマルチメディアデータに対しても、その必要性が高まりつつある。情報フィルタリングの一方式として、ユーザの興味に関する情報を表すユーザプロフィールと対象情報(コンテンツ)の内容をモデル化したコンテンツモデルを比較する内容に基づくフィルタリングがある。音楽データを対象とした内容に基づくフィルタリングに関する既存の研究として、コンテンツモデルとユーザプロフィールの両方を音楽の特徴量のベクトルで表し、ベクトル間距離を用いて推薦する手法に関する研究や、それらの特徴量に任意の評価関数を用いるフレームワークに関する研究などが存在する。本研究はこれらの方法とは性質が異なるアルゴリズムである決定木を用いた内容に基づく音楽情報フィルタリングシステムを提案する。決定木は、ユーザによって重視する音楽の特徴が違って、うまくそれを表現できると思われる。実際の音楽データとユーザを用いて実験を行い、他の手法に比較しての本システムの有効性を検証する。

Content-based Music Filtering System by Decision Tree

Yoshinori HIJIKATA[†] Kazuhiro IWAHAMA[†] Shogo NISHIDA[†]

[†] Graduate School of Engineering Science, Osaka University

Information filtering systems, which recommend appropriate information to users from enormous amount of information on the Internet, are becoming popular. One method of information filtering is content-based filtering that compares a user profile with a content model. Many systems using the content-based filtering deal with text data, and few systems deal with music data. We propose a content-based filtering system for music data by using decision tree. And we conduct an experiment by using real music data and users, and validate the effectiveness of our system compared with other filtering methods.

1 はじめに

従来から、ユーザの情報獲得を支援する目的で、情報フィルタリングシステム(または情報推薦システム)の研究が行われてきた¹⁾。情報フィルタリングシステムとは、膨大な情報の中からユーザの好みや興味に合致する情報をユーザに提供するシステムである。情報フィルタリングシステムを実現する方式には、内容に基づくフィルタリングと協調フィルタリングの二種類がある²⁾。両手法とも始めはテキストデータを対象として研究が行われてきたが、特に前者の内容に基づくフィルタリングについては、フィルタリングに用いる特徴量をマルチメディアデータから抽出する困難さもあり、依然としてテキストデータを対象とした研究が多く、マルチメディアデータを対象とした研究は少ない。

一方、実際のインターネット上では、近年音楽データ

の流通が広まってきている。音楽を対象とした内容に基づく情報フィルタリングの研究も、まだ数は少ないながらも研究が行われ始めている。一般に、音楽を対象とした内容に基づくフィルタリングでは、音楽データから曲の特徴を表す特徴量を抽出し、それらを用いて推薦対象の音楽コンテンツのモデル(以降、コンテンツモデル)を作成する。また、ユーザの嗜好を表すユーザプロフィールも、コンテンツモデルと同様の特徴量を用いて表現する。そして、コンテンツモデルとユーザプロフィールを比較することで推薦する音楽を決定する。

コンテンツモデルとユーザプロフィールをどのようにモデル化するかの、それらの比較方法にはいくつかの方法が考えられる。我々が調査したところ、コンテンツモデルとユーザプロフィールの両方を特徴量のベクトルで表し、ベクトル間距離を用いて推薦する手法に関する

研究や、それらの特徴量に任意の評価関数を用いるフレームワークに関する研究などが存在する。ベクトル間距離を用いる方法は、すべての特徴量を均等に扱うことに等しくなる。しかし、ユーザの嗜好は人によって重視する特徴が違っていると考えられ、必ずしも上記の比較手法が有効とは限らない。なぜなら、興味の分類に関係のない軸では、その方向に1つのクラスが広がってしまい、距離の計算に大きな影響を与えてしまうからである。評価関数を用いた場合は、個々の特徴量の重みをユーザの好みとの相関係数などにより変化させることができるため、ユーザの音楽の好みに影響を与えない特徴量による外乱を回避することができる。しかし、評価関数を用いた場合は1つの超平面でしか分類できないため、嗜好がいくつかの塊(クラス)に分散するような複雑な分類を行うことができない。

また、一般的に情報フィルタリングにおいては、ユーザプロファイルのカスタマイズの容易性が重要視されることがある³⁾。これは、機械学習したユーザプロファイルには誤って学習された部分が避けられず、それらを微修正する必要があるためである。そこで本研究は、決定木^{4, 5)}を用いた内容に基づく音楽情報フィルタリングシステムを提案する。ユーザごとにユーザプロファイルとなる決定木を構築することにより、そのユーザにとって重要な特徴量のみを用いて分類することが可能になる。さらに、決定木のカテゴリは人にとって可読であるため、誤って学習された箇所を修正することが可能である。実際の音楽データとユーザを用いて実験することで、距離に基づく方法と比較しての有効性を検証する。

本論文では、2章で本研究のアプローチについて述べる。3章で関連研究について述べ、本研究との違いを明確にする。4章で本研究で使用する特徴量について説明し、この特徴量に基づくフィルタリング方式を提案する。また、5章でシステムの実装と動作例を示す。そして、6章で評価実験を行い、本研究で提案する推薦手法の有効性を検証する。最後に、7章でまとめを述べる。

2 フィルタリングの概要

本研究では、音楽データの形式としてMIDIを用いる。また、対象とする音楽のジャンルとしてポップス(歌謡曲)を選択する。推薦する音楽のジャンルを限定するのは、音楽のジャンルによりパートや曲の構成が大きく異なるため、推薦に使う特徴量も大きく異なってしまうためである。本研究におけるフィルタリングの概要を図1に示す。ユーザはいくつかの音楽データを自分の好みに合うか合わないかの観点から評価付けを行う。評価付けされた音楽データからシステムは曲の特徴量(属性とその値から成る)を抽出し、ユーザの好みを表すユーザプロファイル(実際には、決定木)を作成する。次に、システムは推薦対象の音楽データから特徴量を抽出し、

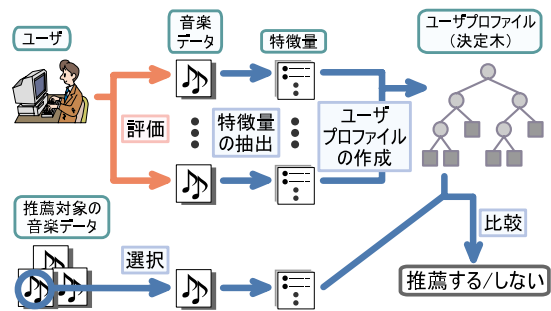


図 1: 本研究におけるフィルタリングの概要

それとユーザプロファイルと比較する。比較した結果から、システムはユーザが気に入りそうな音楽データを選択する。そして、選択した音楽データをユーザに提示する。

3 関連研究

コンテンツに基づくフィルタリングは、これまでテキストデータを対象に多くの研究が行われてきた¹⁾。多くの研究で用いられている手法は、ベクトル空間モデルに基づき、閲覧文書を用いて適合性フィードバックを行うものである⁶⁾。ここでのユーザプロファイルのモデル化は、単語の生起確率に基づいており、長期間ユーザが文書閲覧していると、もともと興味のある単語だけが高頻度なベクトル要素として残ることを利用している。音楽を対象としたコンテンツに基づくフィルタリングでは、テキストにおける単語のように、曲によって有無がはっきりと出る特徴量がなく、音楽には音楽に適したフィルタリング方法を考える必要がある。音楽情報処理の分野では、音楽を対象とした情報検索は数多く研究が行われている⁷⁾。これらの研究は、用いるデータがモノフォニック(単音)であるかポリフォニック(複数音)であるかや、パラメータ形式であるか時系列形式であるかなど、多くの観点から研究が進められている。これまで、音楽から多くの特徴量を取得することに成功しているが、これらをフィルタリングに応用した研究はまだ少ないと言える。

音楽データを対象とした推薦システムに関する研究として、Uitdenbogerdら⁸⁾の研究がある。この研究では、音楽推薦システムの成否に影響する社会的要素と音楽的要素を数種類挙げていますが、実際のフィルタリング方式を考案し、推薦システムを構築・評価するには至っていない。また、実際に音楽推薦システムを構築した例として、Chenら⁹⁾と黒瀬ら¹⁰⁾の研究がある。Chenらのシステムでは、音楽データから曲の特徴量を抽出し、音楽データを特徴量のベクトルで表現する。そのベクトル間距離に基づき音楽データを推薦する。黒瀬らのシステムは特徴量からなる多次元ベクトル空間にお

表 1: フィルタリングに用いる特徴量

拍子 (曲全体)
調性 (曲全体)
平均テンポ (曲全体)
リズム (曲全体)
メジャーコードの割合 (曲全体)
マイナーコードの割合 (曲全体)
sus4 コードの割合 (曲全体)
キー (曲全体)
音色 (メロディCH)
平均音高差 (M-CH, B-CH, C-CH)
平均音長 (M-CH, B-CH, C-CH, D-CH)
平均音長差 (M-CH, B-CH, C-CH, D-CH)
M-CH: メロディCH, B-CH: ベースCH C-CH: コードCH, D-CH: ドラムCH

いて、任意の評価関数を用いて推薦する音楽データを決定する。Chenらの方法では、すべての特徴量を均等に扱うことに等しくなる。そのため、ユーザごとに異なる特徴量を用いて推薦曲を決定することはできない。黒瀬らの方法では、重みを変化させることでユーザごとに重要な特徴量を選択できるが、1つの超平面で2分割することになるので、複雑な分類平面は構築できない。本研究では、ユーザごとにユーザプロファイルとなる決定木を構築し、そのユーザにとって重要な特徴量のみを用いて推薦曲を決定することにより、細かな識別能力をもつ推薦方式を提案する。

4 特徴量とフィルタリング方式

我々は先行研究¹¹⁾において、フィルタリングに用いる特徴量について調査を行った。具体的には、はじめに従来研究で用いられている特徴量や音楽の代表的な特徴量を列挙して、それらを本システムの特徴量の候補とした。次に、それらの特徴量を抽出するプログラムを実装した。そして、実際のMIDIデータにおける候補とした特徴量の分布を調べ、ばらつきの大いものをフィルタリングに用いる特徴量とした。決定した特徴量を表1に示す。ここで、特徴量はチャンネル(以下CH)ごとに抽出するものと、曲全体で抽出するものの2種類がある。ポップスは主にメロディ、コード、ベース及びドラムの4つのパートから構成されるため、これらのCHを対象としている。また、特徴量のうち、音高差とはある音符とその次の音符の音高の差のことである。音長差とはある音符とその次の音符の音長の差のことである。

次いで、決定した特徴量を用いたフィルタリング方式を考案する。分類アルゴリズムには、特徴量をベクトルで表現し、そのベクトル間距離に基づき分類する方法

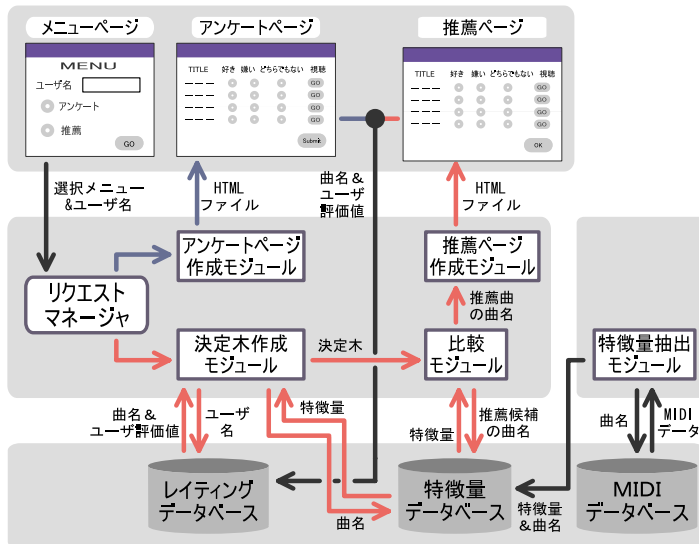
(K-means法¹²⁾、凝集法¹²⁾など)と、特徴量の1つ1つに注目して閾値による条件分岐を行うことで分類する方法(ID3⁴⁾やC4.5⁵⁾などの決定木)の2種類が考えられる。本研究では、後者の手法、すなわち決定木を用いることとする。その理由は1章で述べたとおりであるが、ユーザの嗜好は人によって重視する特徴が違っていると思われるためと、ユーザプロファイルのカスタマイズを可能にするためである。本方式では、ユーザごとにユーザプロファイルとなる決定木を構築する。決定木のアルゴリズムとしては、パラメータとして連続値とカテゴリ変数を扱うことのできるC4.5⁵⁾を採用することとする。

ユーザプロファイルは次のようにして作成する。ユーザは、システムが提示する音楽データに対して、“好き”、“嫌い”または“どちらでもない”の3段階で評価付けを行う。ここで、評価付けの方法として、5段階評価や7段階評価といった任意の離散値でユーザに評価付けをさせることも考えられる。しかし、決定木では等級を扱えないので、等級なしでユーザの好みを表すこととした。そして、ユーザの評価をもとにして、学習を行い、決定木を作成する。ここで、決定木のノードは曲の特徴量についての条件を持ち、葉ノードは“好き”、“嫌い”及び“どちらでもない”というクラスを持つ。クラスはその葉ノードに属する事例の最頻出クラスとする。

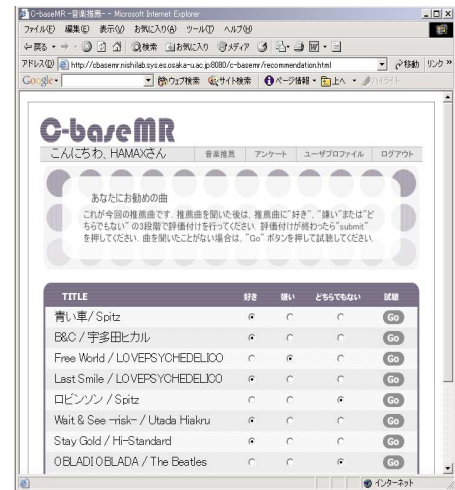
本研究では、コンテンツモデルを音楽データごとに作成する。システムが音楽データから自動的に抽出した特徴量がコンテンツモデルになる。フィルタリングを行う時は、各音楽データのコンテンツモデルを使って決定木の根ノードから探索する。順に探索を行い、葉ノードに達した時にその音楽データが“好き”というクラスになれば、システムはその音楽データをユーザに推薦する。“嫌い”または“どちらでもない”というクラスになれば推薦を行わない。また、ユーザはシステムが推薦した音楽データに対して評価を行う。システムは、この評価をもとに決定木を再学習させることでユーザプロファイルの更新を行う。

5 プロトタイプシステム

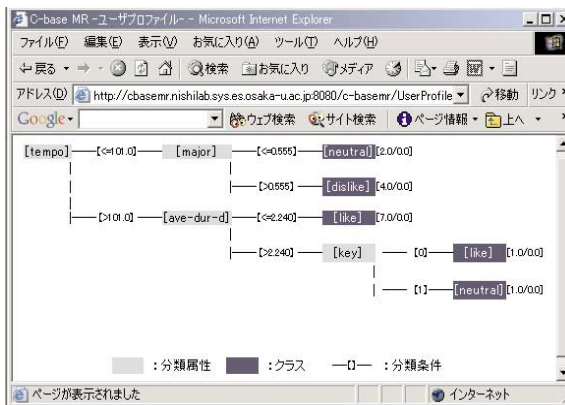
上述のフィルタリング方式を持つプロトタイプシステムをJavaとJavaサーブレットで実装した。このシステムを“C-base MR (Content-based Music Recommender)”と呼んでいる¹³⁾。プロトタイプシステムの構成を図2-(a)に示し、処理の流れを以下に示す。まず、メニューページでユーザがユーザ名を入力し、メニュー(アンケートを行うか、推薦を行うか)を選択する。そしてサーブレットのリクエストマネージャがユーザ名とユーザが選択したメニューを受け取る。ユーザが“アンケート”を選択した場合は、アンケートページ作成モジュールがアンケートページ用のHTMLファイルを作



(a) システム構成



(b) 推薦ページ



(c) ユーザプロフィール



(d) ユーザプロフィール編集

図 2: プロトタイプシステムの構成と画面例

成し、Web ブラウザに送信する。そして、ユーザが音楽データに評価付けを行った後、システムはユーザの評価値をレーティングデータベースに保存する。ここで、レーティングデータベースとは音楽データに対するユーザ評価値を保存するものである。

ユーザが“推薦”を選択した場合は、レーティングデータベースのユーザ評価値と特徴量データベースの対象音楽データの特徴量を用いて、決定木作成モジュールがユーザプロフィールとなる決定木を構築する。次に、比較モジュールが推薦対象の音楽データを用いて決定木を探索し、推薦すべきか否かを決定する。推薦ページ作成モジュールが、推薦する音楽データを表示する推薦ページ(図 2(b))用の HTML ファイルを作成し、Web ブラウザに送信する。また、特徴量の抽出は特徴量抽出モジュールが MIDI データベース中の MIDI データを用いて、オフラインで行う。

また、決定木は図 2(b)のメニュー中のユーザプロファ

イルをクリックすることで、確認することができる(図 2(c))。決定木中のノードとリンクには編集ページへのアンカーが埋め込まれており、これをクリックすることで、該当箇所のユーザプロフィール変更用の画面(図 2(d))が表れる。この画面では、中間ノードであれば特徴量の変更、特徴値の変更、葉ノード化が行える。葉ノードであれば、クラス変更、中間ノード化が行える。リンクであれば特徴値の変更が行える。また、選択している特徴量が連続値の場合は、その特徴量の最大値、最小値、平均と標準偏差を表示している。

6 システムの評価

本章では、プロトタイプシステムの評価を行い、決定木を用いた推薦方法が有効であるかどうかを定量的に検証する。6.1 節で実験方法について説明する。6.2 節でランダムに推薦する方法との比較を行う。これは、全く推薦に妥当性のない方法に比べてどの程度推薦の程度が良くなるのかを見るためと、学習用データとして最

表 2: ランダム推薦との精度比較

学習 データ数	精度		改善率
	ランダム	決定木	
25	0.399	0.433	1.09
50	0.399	0.448	1.12
75	0.399	0.568	1.42
100	0.399	0.543	1.36

低どの程度の数が必要かを確かめるためである。6.3 節と 6.4 節でベクトル距離に基づき分類する方法の代表的なアルゴリズムである K-means 法と凝集法を用いて推薦した場合との比較を行う。6.5 節でユーザプロファイルの編集機能がどの程度有効に働くかを検証する。最後に 6.6 節では、推薦対象の音楽データが少ない場合と大量に存在する場合を考え、提案する推薦方式の実用性について議論する。

6.1 実験方法

システムの評価にあたり、MIDI データ 200 曲を使用した。使用した曲の内訳は、RWC 研究用音楽データベース¹⁴⁾と、インターネット上で公開しているオリジナルの音楽データで作者に実験での利用許可をとったもの、JASRAC¹⁵⁾が著作権を持つ音楽を第 3 者が MIDI データ化したもので JASRAC と MIDI データの作者双方に利用許可をとったものを利用した。次に、ユーザ 10 人にこれらの 200 曲を試聴してもらい、すべての曲に対して好き、嫌いまたはどちらでもないの 3 段階で評価付けを行ってもらった。そして、MIDI データ 200 曲を評価値のばらつきが均等になるように 2 分割し、100 曲を決定木を構築するための学習用データ、残り 100 曲を決定木の評価用データとした。最後に、ユーザに自分のユーザプロファイルを見てもらい、ユーザプロファイル編集ページから編集を行ってもらう。ここで、ユーザがユーザプロファイルを上手く編集できそうにないと思った場合は編集を行わなくてもよいことにした。評価指標には、精度と再現率¹⁶⁾を用いる。

6.2 ランダムに推薦する方法との比較

ここでは、決定木を構築するための学習用データを、25、50、75 及び 100 と変化させ、各ケースにおいて提案手法とランダムに推薦する方法の比較を行う。ランダムに推薦する方法における精度は、評価用データに含まれるユーザが好きな音楽データの割合とする。また再現率は、次のようにして計算する。提案手法を用いて推薦した場合と同数の音楽データを推薦すると仮定し、それに精度を掛けることにより推薦する音楽データに含まれるユーザが好きなデータ数を求める。そして、求

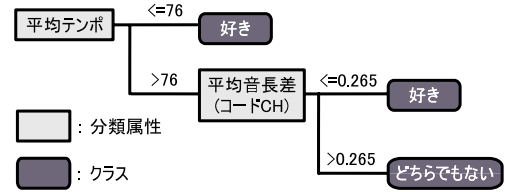


図 3: ユーザ A の決定木 (学習用データ 50 曲)

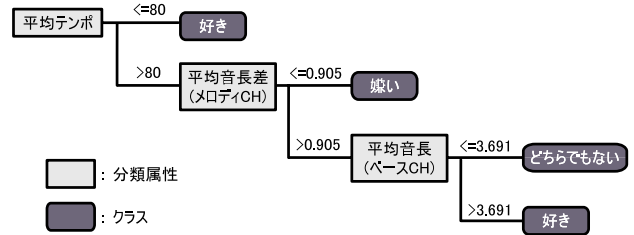


図 4: ユーザ A の決定木 (学習用データ 75 曲)

めた値と評価用データに含まれるユーザが好きな音楽データの数から再現率を計算する。

表 2 に精度とその改善率を示す (再現率に関しては、目立った変化は見られなかったので省略する)。精度の改善率は、学習データ数が少ない時は 1.1 程度であるのに対し、学習データ数が多くなると 1.4 程度になっている。これは、学習データ数が少ないときは決定木を構築するために十分な学習を行うことができなかったからだと考えられる。

図 3 に、学習用データが 50 曲のときのあるユーザ (6.3 節の表 4 のアンケート例におけるユーザ A) の決定木を示す。アンケートでは、このユーザはテンポが遅く、メロディがしっかりしている曲を好む傾向があった。図 3 では、1 つ目の分類属性はテンポであるが、2 つ目の分類属性はコード CH の平均音長差となっている。したがって、この決定木はテンポが遅いという評価基準は反映しているが、メロディラインがしっかりしているという評価基準は全く反映していないと言える。

図 4 に、学習用データが 75 曲のときの上記ユーザの決定木を示す。1 つ目の分類属性はテンポであり、2 つ目の分類属性はコード CH でなくメロディ CH の平均音長差となっている。この決定木から、テンポが遅いという評価基準とメロディラインがしっかりしているという評価基準が反映されていることが分かる。

75 曲 (及び 100 曲) の学習データを用いた場合に提案手法とランダムに推薦する方法の精度に有意差があるかどうかを調べるために t-検定 (片側) を行った。その結果、有意水準 $\alpha = 0.05$ で提案手法の精度とランダム推薦の精度で有意差が見られた。このことから、学習データ数が 75 曲以上の場合は提案手法の方がランダム

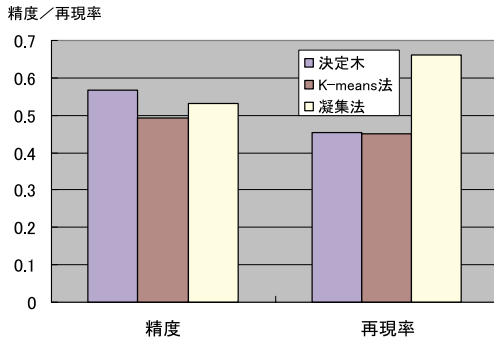


図 5: 提案手法 (決定木) とベクトル距離に基づく推薦方法 (K-means 法と凝集法) の比較

に推薦する方法よりも有効であるといえる。すなわち、本研究で提案する推薦手法が有効に働くには最低 75 曲程度の学習データが必要になると考えられる。

6.3 K-means 法を用いた推薦方法との比較

学習用データとして 75 曲を使用して、提案手法と K-means 法による推薦方法の比較を行う。なお、評価用データの数は 100 曲で変わらない。本研究で扱う音楽データの特徴量を K-means 法で用いるために、以下の方法でベクトルに変換する。特徴量が連続値である場合は、そのままの値をベクトルの要素とする。カテゴリ変数である場合は、各カテゴリを 0-1 の間に等間隔で配置することとする。生成するクラスタ数は決定木の葉ノードの数と等しくする。また、クラスタ間の距離はクラスタの重心間の距離とし (重心法)、距離の計算はユークリッド距離で行う。次に、各クラスタに含まれる音楽データに対するユーザの評価のうち、最も多い評価値をそのクラスタのラベルとする。

提案手法により音楽データを推薦した場合と K-means 法を用いて推薦した場合の精度と再現率を図 5 に示す。図 5 より、再現率は同等で、精度は提案手法を用いた場合の方が K-means 法を用いた場合よりも高くなったことが分かる。提案手法を用いた場合と K-means 法を用いた場合で、精度に有意差があるかを調べるために t-検定 (片側) を行った。その結果、有意水準 $\alpha = 0.05$ で有意差が見られた。ここで、推薦したデータの数に注目してみる。図 3 に、ユーザごとに各手法が推薦した音楽データの数と、評価データ中においてどれだけの曲を「好き」だったかを示す。図 3 より、提案手法を用いて推薦したデータ数が K-means 法を用いて推薦した場合よりも少なかったことが分かる。このことから、提案手法は K-means 法よりも、好きな曲をより絞り込んで推薦できたことが分かる。

次に、なぜ提案手法を用いて生成した分類ルールの方が精度が高くなったのかを調べるために、実験に参加したユーザ 10 人にどのような基準で曲に評価値をつけた

表 3: 各手法を用いた場合の推薦データ数

ユーザ	A	B	C	D	E	F	G	H	I	J	平均
提案手法	27	7	7	23	43	81	61	5	87	11	35.2
K-means法	35	3	3	52	46	85	47	11	100	13	39.5
凝集法	96	0	5	0	95	85	3	0	99	0	38.3
「好き」な曲数	43	27	36	34	45	54	36	39	54	38	40.6

表 4: アンケート結果の例

ユーザ	コメント
ユーザ A	<ul style="list-style-type: none"> テンポが遅く、メロディラインがしっかりしている曲に好きという評価を付けた。 テンポが速いと感じた曲やロック (低音が多い) やパンク色の強い曲には嫌いという評価を付けた。 どちらにも該当しないものに、どちらでもないという評価を付けた。
ユーザ B	<ul style="list-style-type: none"> ドラムが一定速度かつ速い曲に好きという評価を付けた。 同じベース音が繰り返している曲に好きという評価を付けた。 ミディアムテンポで普通な感じの曲に嫌いという評価を付けた。 どちらにも該当しないものに、どちらでもないという評価を付けた。
ユーザ C	<ul style="list-style-type: none"> テンポが速い曲に好きという評価を付けた。 テンポが普通かつ明るい感じの曲に好きという評価を付けた。 テンポが遅くかつ明るい感じの曲に好きという評価を付けた。 テンポが普通かつ暗い感じの曲にどちらでもないという評価を付けた。 テンポが遅くかつ暗い感じの曲に嫌いという評価を付けた。

かを尋ねるアンケートを行った。アンケート結果の例を表 4 に示す。表 4 から、今回実験に参加したユーザは曲の全体的な雰囲気ではなく、数種類の特徴に着目して評価を付けている傾向があることが分かる (「テンポが速い曲が好き」、「テンポの遅い曲が嫌い」など)。ユーザ A の決定木を図 4 に示す。図 4 より、1 つ目の分類属性としてテンポ、2 番目の分類属性として平均音長差 (メロディ CH) が選択されていることが分かる。アンケート結果からテンポとメロディ CH に関する特徴がユーザ A の音楽の嗜好に強い影響を与える特徴量になっていることが確認でき、このことが決定木に強く反映されていることが分かる。それに対して K-means 法を用いて分類ルールを生成した場合は、このように数種類の特徴量が音楽の好みに強い影響を与えていると、そうでない特徴量はベクトル間距離を計算するとき外乱になる。さらに表 4 から、テンポなど多くのユーザの音楽の嗜好に強い影響を与える特徴もあるが、メロディ

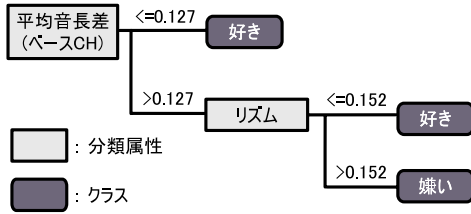


図 6: ユーザ B の決定木 (学習用データ 75 曲)

に関する特徴やベースに関する特徴などは、それに注目するかどうかはユーザによって異なることが分かる。具体的には、ユーザ A はテンポやメロディに関する特徴を重視しているが、ユーザ B はドラムやベースの音を重視している (表 4 参照)。ユーザ B の決定木を図 6 に示す。図 4 と図 6 より、各決定木が使用している分類属性はユーザによって異なり、各ユーザの好みを反映した分類属性を用いていることが確認できる。これらのことから、ユーザの嗜好に特定の特徴が強く影響を与え、なおかつユーザによって重要な特徴が異なる音楽には、K-means 法よりも提案手法の方がユーザの好みをよく表すと言える。

6.4 凝集法を用いた推薦方法との比較

学習用データとして 75 曲を使用して、提案手法と凝集法による推薦方法の比較を行う。6.3 節の方法により特徴量をベクトルに変換した後、凝集法によりクラスタリングを行う。そして、クラスタリングによって生成された樹形図から、決定木の葉ノードの数と等しくなるようにクラスタを決定する。クラスタのラベル付けは 6.3 節の K-means 法の場合と同様である。距離の計算には、平方ユークリッド距離を用いる。

凝集法は他の手法と違って、10 人中 4 人のユーザにおいて、一曲も推薦されない結果となった (図 3 参照)。凝集法は、ボトムアップにクラスタリングを行うため、上位ノードにおいては、多次元空間において離れた空間に位置する 1 つまたは少数のデータからなるクラスタを分岐していきがちである。決定木の葉ノードの数になるように上位ノードで区切るため、ユーザの評価における最も多い評価値が”好き”でない場合、極端なケースでは少数のデータからなる $n - 1$ 個のクラスタと、多数のデータからなる 1 個の一般的なクラスタ (評価値の分布から”好き”以外のクラスタとなる) ができることがある。この場合は、1 つもデータが推薦されないことも有り得る。データが推薦されたユーザは、再現率が高くなっているが (図 5)、これはユーザの評価における最も多い評価値が”好き”の場合、上記で説明した 1 個の一般的なクラスタの最頻出のクラスが”好き”であったためである。図 3 から分かるように、推薦されているユーザには非常に多くの曲が推薦されている。

表 5: ユーザプロフィール編集後の精度

ユーザ	編集の有無	精度 (編集前)	精度 (編集後)
ユーザ A		0.851	0.956
ユーザ B	×	-	-
ユーザ C	×	-	-
ユーザ D		0.522	0.500
ユーザ E	×	-	-
ユーザ F		0.617	0.700
ユーザ G		0.426	0.444
ユーザ H	×	-	-
ユーザ I		0.426	0.596
ユーザ J		0.545	0.632

このように凝集法は、学習データのクラス分布に依存してしまうため、適用が困難であると言える。

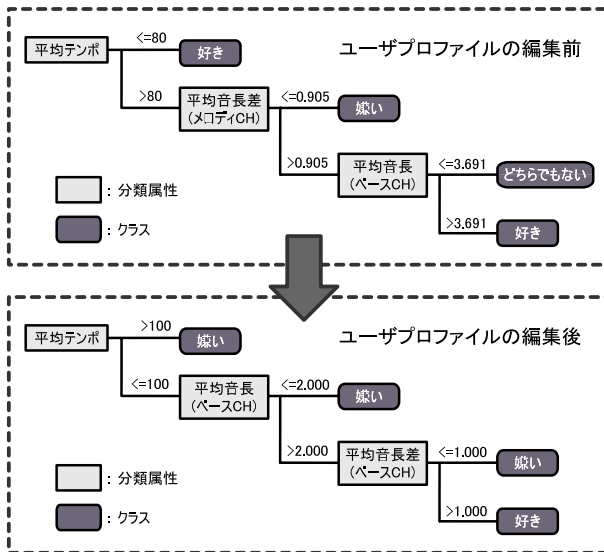
6.5 ユーザプロフィールの編集

ユーザプロフィール編集前と編集後で精度に差があるかどうかを調べる。ユーザプロフィールの編集結果を表 5 に示す。10 人中 6 人が編集を行い、そのうち 5 人に関して編集後の方が精度が向上していることが分かる。図 7 に、ユーザ A のユーザプロフィールの編集例を示す。ユーザ A は、テンポが速く、ロックやパンク色の強い曲を嫌う傾向がある。そこでユーザ A は、編集前のユーザプロフィールを参考にして、嫌いな曲を落とすという戦略を採った。最初のカテゴリ属性としてテンポを使用し、次にロックやパンク色に影響を与えるベース音の音長に関する特徴量を使用して編集を行った。その結果、編集前のユーザプロフィールよりもユーザの好みを反映したプロフィールが生成され、精度が向上した。

しかし、ユーザ D については編集前の方が精度が高く、4 人のユーザに関してはユーザプロフィールの編集を行っていない。これは、ユーザプロフィールを編集するためには、自分の曲の好みをよく理解していないと書き換えができないことや、ある程度の音楽の知識がないとどの特徴量に注目すればよいか分からないためと考えられる。これらのことから、ユーザプロフィールの編集は必ずしも容易ではなく、また下手に編集すると逆に精度が下がる危険性はあるが、うまく編集できれば、精度向上が可能であると言える。

6.6 推薦数と実用性に関する議論

今回の評価では、100 曲の中から音楽データを推薦するという条件で実験を行った。このように少ないデータ数からは、ある程度の数の音楽が推薦できた方が良いと思われる。しかし、ユーザによっては数曲しか推薦されない結果となった (図 3 参照)。この場合、信頼性の高



- ・テンポが遅く、メロディラインがしっかりしている曲に好きという評価を付けた。
- ・テンポが速いと感じた曲やロック(低音が多い)やパンク色の強い曲には嫌いという評価を付けた。
- ・どちらにも該当しないものに、どちらでもないという評価を付けた。

図 7: ユーザプロフィールの編集例

い決定木による推薦に加えて、多少信頼性が落ちたとしても、好みの音楽である可能性のあるデータを試行的に推薦することも一つの方式として考えられる。

また逆に、本実験では 100 曲中平均して 35.2 曲の音楽データを推薦している。これは、学習用データに入っていた「好きな」音楽データをなるべく多く推薦するという目的には良いが、現実の世界では膨大な数の音楽データを持っている Web サイトやシステムもあり、これらを一度に推薦してしまうと、ユーザの負担となる。今回の実験では、決定木では各葉ノードにラベル付けを行う際に、単純に葉ノード中で最もデータの多いクラスを用いている。このことが多くの音楽データを推薦するに至った原因と言える。推薦データ数を少なくすることを考えた場合、決定木のラベル付けに葉ノード中のデータの割合に関する閾値を設け、この閾値を高く設定すれば、推薦データ数の絞込みは可能と思われる。

7 むすび

本稿では、音楽情報を対象とした決定木を用いた内容に基づくフィルタリング方式を提案し、Java と Java サブレットを用いてプロトタイプシステムを構築した。従来手法である、ベクトル距離に基づく方法と比較したところ、決定木の方がユーザの重視する音楽の特徴量をうまく表現でき、ユーザの好む曲をより絞り込んで推薦することが分かった。また、決定木はユーザにとって可読であるため、ユーザが好む曲の特徴をよく理解している場合で、なおかつある程度の音楽の知識がある

場合は、学習したユーザプロフィールを編集し、より自分にあったユーザプロフィールに書き換えられることも確認した。しかし、ユーザの学習用データによっては、推薦するデータ数が少ないケースも見られ、別の観点から推薦する代替手法が必要となる。また大量の音楽データを持つ環境では、推薦数を制限するような工夫が必要である。今後は、これらの音楽データ数が大きすぎる場合や小さすぎる場合に対する研究を進めていく予定である。

参考文献

- 1) Resnick, P. and Varian, H.R.: Recommender Systems, *Comm. of the ACM*, Vol.40, No.3, pp.56-89 (1997).
- 2) Riecken, D.: Personalized Views of Personalization, *Comm. of the ACM*, Vol.43, No.8, pp.26-158 (2000).
- 3) Vassileva, J.: A Practical Architecture for User Modeling in a Hypermedia-Based Information Systems, *In Proceedings of 4th International Conference on User Modeling*, pp.115-120 (1994).
- 4) Quinlan, J.R.: Induction of Decision Trees, *Machine Learning*, vol.11, pp.81-106, (1986).
- 5) Quinlan, J.R.: C4.5 Programs for Machine Learning, Morgan Kaufmann (1993).
- 6) 土方嘉徳: 情報推薦・情報フィルタリングのためのユーザプロフィール技術, *人工知能学会学会誌*, Vol.19, No.3, pp.365-372, (2004).
- 7) ISMIR: The International Conferences on Music Information Retrieval, <http://www.ismir.net/>
- 8) Uitdenbogerd, A. and Schyndel, R.V.: A Review of Factors Affecting Music Recommender Success, *In Proceedings of International Symposium on Music Information Retrieval (ISMIR 2002)*, (2002).
- 9) Chen, H. and Chen, A.L.P.: A music recommendation system based on music data grouping and user interests, *In Proceedings of the tenth international conference on Information and knowledge management*, (2001).
- 10) 黒瀬崇弘, 梶川嘉延, 野村康: 感性情報を用いた楽曲推薦システム, 第 14 回データ工学ワークショップ (DEWS2002), 8-P-6 (2003).
- 11) 岩濱数宏, 土方嘉徳, 西田正吾: 内容に基づく音楽情報フィルタリングシステム, データベースと Web 情報システムに関するシンポジウム (DBWeb2003), pp.69-76 (2003).
- 12) Berry, M.J.A. and Linoff, G.: *Data Mining Techniques, For Marketing, Sales, and Customer Support*, Wiley Computing Publishing (1997).
- 13) インターメディアフォーラム 2003, C-base MR : 内容に基づく音楽情報フィルタリングシステム, 大阪大学西田研究室, 2003 年 10 月, マイドーム大阪, <http://www.jma.or.jp/imf/>
- 14) RWC 研究用音楽データベース, <http://staff.aist.go.jp/m.goto/RWC-MDB/index-j.html>
- 15) 社団法人日本音楽著作権協会 JASRAC, <http://www.jasrac.or.jp/>
- 16) 北研二, 津田和彦, 獅々堀正幹: 情報検索アルゴリズム, 共立出版, (2002).