3B3-4

ユーザの嗜好を反映したプレイリスト生成のためのエージェントの設計

The agent's for playlist generation that reflects the user's preference design

溝口文雄[*1*2*3] Fumio Mizoguchi　竹内康夫[*1] Yasuo Takeuchi　山崎航[*2] Wataru Yamazaki　平石広典[*2*3] Hironori Hiraishi

[*1]東京理科大学理工学部 Faculty of Sci.and Tech,Tokyo University of Science

[*2]東京理科大学総合研究所 Research Institute for Science and Technology

[*3]ウィズダムテック株式会社 Wisdomtex,Inc.

Recently, the music data in PC increases remarkably. It is very difficult for the user to choose the music suitable for own favor from among a lot of musics. We design the music management tool of new type that makes the playlist corresponding to the preference of the individual and a lot of owned musics are visualized,and the entire prasp is facilitated.

1. はじめに

近年、Apple 社の iPod や iTune などのモバイルデバイスや、管理ソフトの登場により、パーソナルコンピュータで扱う音楽データの数は増加している。ユーザは自分の音楽ライブラリの中から、個人の趣向や状況に合わせて楽曲を選択・視聴するが、PC 中の楽曲数が多いため、ユーザが多くの楽曲の中から自分の好みやシチュエーションに適した楽曲の選択は困難になってきている。既存の音楽管理ツール等ではランダム再生機能や楽曲の付加情報を利用した楽曲選択機能 (例えばリリース年月日から、『70 年代の曲』をリストアップするなど) を備えているが、ある特定の嗜好やシチュエーション、また楽曲間の情報が利用されておらず、ユーザのニーズに合っているとはいえない。そこで、本論では大規模楽曲リスト中から個人の嗜好に応じた楽曲プレイリストを自動生成することにより楽曲選択難化問題の解決を図ることを目的とし、楽曲データや視聴履歴などを用いて個人の嗜好をプレイリストに反映するためのルールを導きだすエージェント機能を備えた音楽管理ツール iDJ の設計を行う。

2. iDJ

本章では iDJ の設計について述べる。iDJ は個人の嗜好を反映したプレイリストを作成するエージェント、及びユーザが持つ楽曲リストを視覚化する 2 つの機能を有している。

2.1 プレイリストの生成

iDJ では楽曲や視聴履歴などの情報を基にしてルールを導き出し、それらを反映したプレイリストを作成する。複数の事例や事実関係もとにルールの導き出す手法として、本論では帰納論理プログラミング (ILP) を用いた。背景知識には楽曲のタイトルや時間、収録アルバムなどの楽曲属性、及び視聴回数や時間、順番などの履歴属性がある。これらを利用することによって、既存の音楽管理ツールにはなかった、ある特定の視聴シチュエーションや楽曲間の関係性を利用したルールが導出でき、個人の嗜好を反映したプレイリストを作成することができる。プレイリスト生成のインターフェースを図 1 に示す。左側の GUI でプレイリストにおける各ジャンルの曲の割合を指定し、それらにルールを合わせて右側の GUI にリストを出力する。

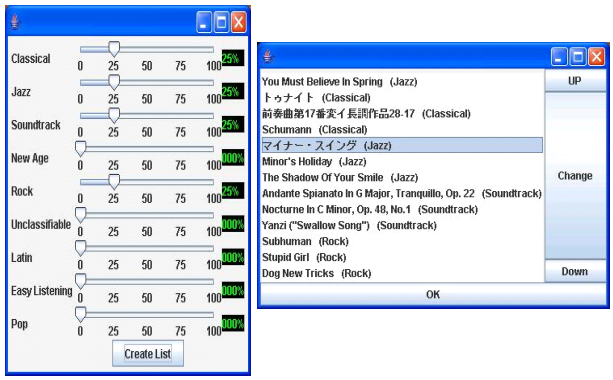

図 1: プレイリスト生成のインターフェース

2.2 楽曲リストの視覚化

既存の音楽管理ツールは所有している音楽を文字情報のみでリストアップしている為、ユーザは自分の音楽リストの状況を把握することは困難である。ユーザが自身の持つ日々増加していく楽曲リストをより直感的に全体を把握することを容易にするため視覚化を行うことは有効である。本論ではハイパボリックツリーと呼ばれる、情報階層構造全体を可視化する技術を利用することにより視覚化を行った。各ジャンル、アーティスト、楽曲にそれぞれノードを作成し、それぞれの所属関係を利用してツリー構造を構築する。例えば、楽曲 A が収録されているのがアルバム B であるなら楽曲 A はアーティスト B の子であるといえる。以下に実際に楽曲リストを視覚化した図 2 に示す。中心のノードから下層にいくにつれ、各ノードはジャンル、アーティスト、楽曲の順に示しており、楽曲データベースの全体像が把握が容易といえる。

3. ILP の適用

前節で述べた知識表現を用いて、個人の嗜好を反映したルールを生成する。本論では論理プログラミング形式の機械学習の手法である帰納論理プログラミング (ILP) を用いる。ILP は正例 (PE)、負例 (NE) の事例、また事例に関する背景知識 (BK) から構成され、これらを用いてルールを導き出す。iDJ

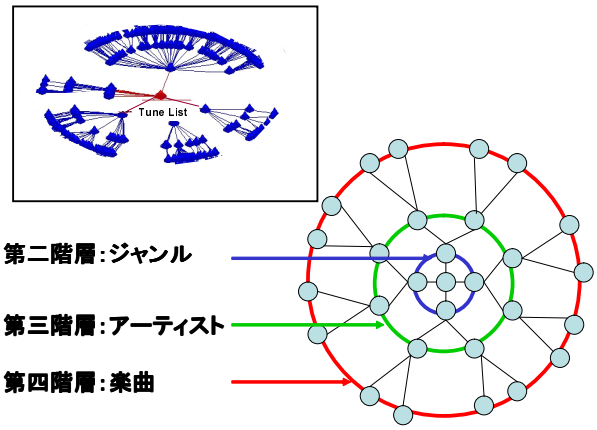

図 2: TuneMap

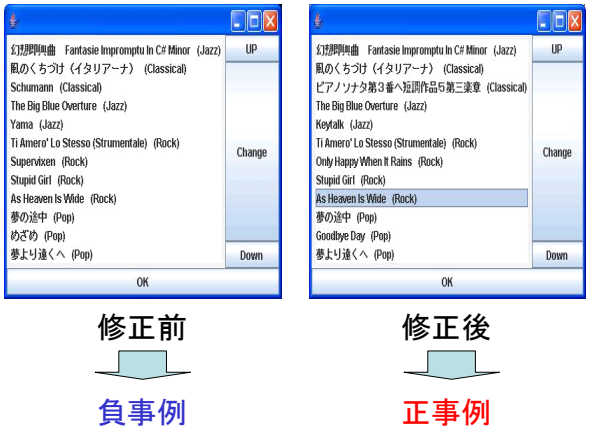

図 3: 正事例と負事例の生成

における PE とはユーザに好ましいプレイリスト、NE は好ましくなかったプレイリストとして定義する。例えばユーザ定義のプレイリストは常にユーザに好ましいので PE。またあるプレイリストに対しユーザが曲を修正した場合、修正前が NE となり修正後が PE となる。BK は楽曲属性やユーザの履歴属性を利用する。

例えば、楽曲 tune_a、tune_b、tune_c から成るプレイリスト list1 があり、ユーザは tune_b を tune_d に変更して list2 を作成したとする。この場合の PE、NE、BK は以下のように表現できる。

01 :- class(situation1, list1). // negative example
02 class(situation1, list2). // positive example.
// background knowledge
03 plist(list1, [tune_a,tune_b,tune_c]).
04 plist(list2,[tune_a, tune_new, tune_c]).
05 generate(list1,time(sunday,0900)).
06 generate(list2, time(Sunday,0905)).
07 removed(list1,tune_b,time(sunday,0905)).
08 title(tune_a, "titleOfA ").
09 genre(tune_a, "Jazz "),
10 length(tune_a, 300).

BK は楽曲属性や履歴属性などから得た事実をもとに構成される。例えば 5,6 行目はプレイリスト list1,list2 が日曜の AM9:00,AM9:05 に作成され、7 行目では tune_b が AM9:05 に list1 から削除されたことを示している。これらの節を利用することにより特定のシチュエーションを考慮したルールを導くことが可能である。

以上の要素から導きだしたルールは次のように表現できる。

class(Genre,List):- C1,C2, ⋯ *Cn*

ここで *C1,C2,* ⋯ *Cn* は背景知識を示している。

以上で述べたようなルールを利用したプレイリストをユーザに提案し、修正を加えられることにより正例、負例を得る (図 3)。これを繰り返すことにより事例が増え、よりユーザの嗜好を反映したルールを作成することが可能であると考えられる。

4. 実験

以上で述べてきた iDJ のシステムの実験を行う。実験に際し、s プレイリスト作成・修正を繰り返すことにより作成された 20 個ずつの PE と NE を用いることにより、プレイリスト作成の為のルールを導き出す。

リストの修正は、ジャンルの選択は Jazz が 50%、Rock が 50%の割合でリストに残し、Jazz の後に必ず Rock をセットするように行った。結果として以下に示すようなルールを取得できた。

class(default,List):-List=[Top—Rest],genre(Top,Jazz), rate(List,50,Rock),rate(List,50,Jazz)startsAfterEnd(A,B), genre(A,Rock),genre(B,Jazz)

ここで *rate(List,50,Rock)* はリスト中に 50%Rock の楽曲があることを示している。また *genre(A,Rock),genre(B,Jazz)* は楽曲 A,B がそれぞれ Rock,Jazz の楽曲であることを示し、*startsAterEnd(A,B)* では楽曲 A は B の直後にセットされることを示している。よってユーザの意思のルール化が出来たといえる。またこれらのルールは楽曲間の関係が考慮されていることがわかる。

5. おわりに

本論ではユーザの嗜好に応じたプレイリストを生成するエージェント機能を備えた音楽管理ツール iDJ の設計を行った。iDJ では楽曲情報や履歴情報を属性として作成した事例と背景知識をベースとして ILP を用いてルールの生成を行い、プレイリストに反映する音楽管理ツールである。実験により既存の音楽管理ツールの各機能の欠点であった、動的なプレイリスト、新しく追加された楽曲の反映、楽曲間の関係を考慮したルールを作成できたことがわかった。今後の展望としては、属性を増やし背景知識を充実させ、より豊かなルール表現を可能にすることでパーソナライズ化を進めて行きたいと考えている。

参考文献

[1] Muggleton, S, Inverse Entailment and Progol. New Gen-eration Computing, Vol.13, No.4-5, pp.245-286, 1995

[2] 沢井宏, 大和田勇人, 溝口文雄, " WebMap: Hyperboric-Tree を利用した WWW ブラウジングの支援 ", 情報処理学会第 58 回全国大会講演論文集,pp.3-67〜3-68,1999