UniversalPlaylist: 利用者の嗜好に動的に適合するメディア再生機構

暦本純一

株式会社ソニーコンピュータサイエンス研究所
インタラクションラボラトリー
141-0022 東京都品川区東五反田 3-14-13
Phone: +81-3-5448-4380, Fax: +81-3-5448-4273
rekimoto@acm.org
http://www.csl.sony.co.jp/person/rekimoto/

要旨

音楽などの多量のデジタルコンテンツを、利用者の操作負担を最小限に抑えつつ、利用者のその時々での嗜好に合致して選択・再生するための手法, UniversalPlaylist を提案する。「複数のプレイリストを集合として扱い、プレイリストの評価重み付けを利用者の嗜好に合わせて動的に変更し、プレイリストの重みから各楽曲の優先度を計算する」という考え方を基本としている。利用者が、再生される楽曲に対し YES/NO などの簡単な指定を繰り返すだけで、コンテンツの属性情報や検索パラメタなどの詳細情報に立ち入ることなく、選択・再生される楽曲の傾向が変化していく。操作インタフェースが非常に単純なので、携帯音楽プレイヤーのリモコンなどへの搭載が容易である。本論文では UniveralPlaylist の計算アルゴリズム、および提案方式に基づいて試作した音楽プレイヤーとその使用経験について報告する。

UniversalPlaylist: A Dynamic Adaptation Mechanism for Media Playback

Jun Rekimoto

Interaction Laboratory
Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda, Shinagawa-ku Tokyo 1410022 Japan
Phone: +81-3-5448-4380, Fax: +81-3-5448-4273
rekimoto@acm.org
http://www.csl.sony.co.jp/person/rekimoto/

Abstract

This paper proposes a mechanism for dynamically adapting to user's preference while selecting digital contents, typically music contents, during playback. This mechanism, named "UniversalPlaylist", uses a collection of contents, known as playlists, and assign weight parameter to each playlist. Users only required to replay "Yes" or "No" for presented music item and the system automatically refines weight parameters of playlists which contain the presented item. Since this interaction is very simple, it should be easy to install this interface to portable music players or remote commanders. This paper reports selection algorithm, and pilot music player systems based on this mechanism, and initial user experience.

1 はじめに

音楽や写真などのメディアをデジタル形式で取り扱うことが一般化している。それに伴い、音楽の再生や写真の閲覧などにおいて、従来の物理媒体に依存していたものとは異なった操作スタイルが発生しつつある。

従来の音楽再生では、音楽ソースはCDやカセットテープなどの物理媒体に記録されており、利用者はそのどれかを選んで再生するのが普通であった。一方、すべての楽曲がデジタルデータとしてハードディスクなどの大容量ストレージに格納されている場合、適当なプレイリストを作成しておき、それを選んで再生する、全体をシャッフルしてランダムに演奏する、などが可能になる。

従来は、メディア交換の操作が煩雑なので、音楽再生は事実上メディア単位であったが、ハードディスクなどの大容量ストレージに格納されている場合、その制限はなく、再生すべき楽曲はもっぱら利用者の嗜好によって決めることができる。

しかし、格納されている楽曲の件数が膨大なものになった場合、毎回再生すべき楽曲を指定する方式は煩雑になってしまう。そのときの気分に合った楽曲を、膨大な楽曲の集合から検索することは容易ではない。

こういった理由から、再生装置（再生ソフトウェア）が、利用者の嗜好に合った楽曲を選択する機能を提供することが注目されている。たとえば、MoodLogic[7]では、楽曲の特徴（種類、テンポなど）をネットワーク上のデータベースに格納しておき、プレイヤーソフトがそのデータベースにアクセスすることで、自動的にプレイリストを生成する機能を提供している。たとえば、「ジャンルはJazzでテンポは120以下」のような指定をして楽曲を選択することができる。

しかし、このような指定方法は、以下の問題を持っている。

- プレイリストの生成にあたり、検索パラメタの入力などのため、通常かなり複雑なユーザ操作が必要となるので、入出力手段の限られた携帯機器での利用が難しい。携帯機器で利用する場合は、PC環境等であらかじめプレイリストを生成しておき、それを携帯機器に転送することになり、自由度が著しく阻害される。
- 利用者が、自分の聞きたい音楽をパラメタの形式で正確に指定するのは簡単ではない。たとえば、自分が現在聞きたい音楽をテンポ速度の範囲や曲のジャンルなどの組み合わせで表現することは直感的な作業ではない。
- 聞きたい音楽の雰囲気は、場所や時間、あるいは音楽を聴く行為自体によっても変化するが、そのたびに毎回パラメタを設定しなおすのは非常に煩雑である。
- 楽曲の選択が、あらかじめ決められた属性値によって決定されるので、それより細かな嗜好がある場合の指定をする手段がない。指定したパラメタで生成された曲順が好みに合わないとわかった場合、また最初から指定をしなおさなければならない。
- 楽曲を選択するアルゴリズムを利用者が簡単に拡張することができない。

本論文では、利用者に明示的なパラメタ指定を要求することなく、そのときの雰囲気にあった楽曲を自動的に選択する手法を提案する。

本手法の要旨は以下の通りである。

- **プレイリストによる再生を基本とする。**プレイリストはすでに広く利用されており、「あるテーマに基づいて曲をグループ化しておく」という概念は利用者にとって理解しやすい。プレイリストは後述するように、利用者が自分で曲をグループ化して作成するもの以外に、タグや楽曲分析によって自動生成されたものなどを含む。
- ただし、従来のプレイリスト再生と異なり、**プレイリストを集合として扱う。**すなわち、単一のプレイリストに属する曲を再生するのではなく、すべてのプレイリストの集合体から、適当な重み付け関数によって、曲の優先度を決定する。この優先度の高い曲がより優先して再生されるようなアルゴリズムを定義する。
- 利用者は、再生された曲が好みにあうかどうかを, 再生時にYes・Noをボタンで指示するなどの簡単な方法で再生システムにフィードバックする。
- フィードバックされた情報に基づいて、その楽曲を含むプレイリストの重み付け係数を更新する。単純に曲そのものの優先度を変えるのではなく、プレイリストの重みを変化させることで、プレイリストに所属する他の楽曲の再生優先度に間接的に影響を与えることになる。

このようにして、最初はほぼランダムな再生の状態から開始しても、提示される曲に対してYes

(気分に合っている)、No (合っていない) などの簡単な指定を繰り返すことで、利用者の好みに動的に適合していく。No は、曲再生のスキップ（飛ばし再生）を兼ねることが可能で、気に入らない曲を飛ばして聞いていると自然に曲の優先度が変化していく。操作の際、画面表示は必須ではないので、携帯音楽再生機器やリモコンなどに搭載することが可能である。さらに、再生中に気分が変化していった場合でも、それに動的に追随することが可能である。

以上のようにプレイリストを集合体として扱う本論文の提案方式を UniversalPlaylist と呼ぶことにする。本論文では需要が特に高いと思われる音楽再生を中心に議論するが、同様のアルゴリズムがデジタル画像の自動巡回表示など、さまざまな場面で適用できるものと期待している。

2 UniversalPlaylist

2.1 プレイリスト

前節で説明したように、UniversalPlaylist ではプレイリストの集合体によって再生する楽曲を選択する。本手法で用いるプレイリストは、以下の 5 種類を含む。

通常のプレイリストファイル: 利用者が複数の楽曲を選択しファイルに保存したもの。winamp やその他のプレイヤーソフトで広く利用されている “.m3u” 形式のファイルなど。利用者が好みに従って曲をグループ化しておくことができる。

タグ情報: 楽曲のタグから抽出されたリスト。たとえば MP3 ファイルには ID3 とよばれる形式で楽曲情報が格納されている。これらから、曲の種別（Jazz, Rock, Classical, Opera, など）にごとにリストを生成する。

ディレクトリ構造: 楽曲を格納するディレクトリ構造。利用者が何らかのポリシーでディレクトリを構成している場合に、それをグループとみなす。また、プレイヤーソフトによっては、ファイルのタグ情報から自動的にディレクトリ階層を生成するものがあるが、これもプレイリストの形で曲選択に反映することになる。

楽曲分析情報: 楽曲を分析した結果得られた属性情報。たとえばビート成分の強度、テンポ、曲調（長調か短調か）などの情報に基づきリストを生成する。

（本論文の試作システムではビート成分強度をを自動生成リストとして利用している。ビート成分は、楽曲を短時間フーリエ変換し、低周波成分をフィルタリングしたものの自己相関係数を求め、相関係数が高い（音量の周期的変動が規則的である）ものを「ビート性が高い」ものとしている)。

再生履歴: 過去の一定期間に再生された曲をリストとして保存しておく。

本手法の特徴は、これらの異なった素性の情報を、プレイリストという形式統一的に扱える点にある。たとえば MoodLogic のように独自のデータベースを構築して、ID3 よりも細かな楽曲属性情報を提供しているサービスがあるが、これらから得られた情報もプレイリストの形式で本提案の枠組みで取り扱うことができる。同様に楽曲内容分析による「類似した楽曲の集合」もリストの一種として扱うことが可能である。

2.2 重み付けアルゴリズム

今、全プレイリストの集合を $L_i(i = 1, \ldots, n)$, 全楽曲の集合を $M_j\,(j = 1, \ldots, m)$ とする。各プレイリストは（後述するアルゴリズムで決定される) 重み係数 $W(i)(i = 1, \ldots, n)$ を持つ。$W(i)$ の初期値は 0 とする。また、プレイリストと楽曲ファイルの関係は、

$$C_{i,j} = \begin{cases} 1 & L_i \text{が} M_j \text{を含むとき} \\ 0 & L_i \text{が} M_j \text{を含まないとき} \end{cases} \tag{1}$$

とする。

このとき、各曲の優先度 $P(j)$ を以下で決定する：

$$P(j) = \sum_{i=1}^{n} W(i) C_{i,j}$$

この優先度に従った確率で次に再生すべき曲を選択する。

プレイリストの重み係数は、以下のようにして決定する。今、利用者がいくつかの楽曲について Yes,No で嗜好を答えたとする:

$$\{\{X_1, Y_1\}, \ldots \{X_k, Y_k\}\}$$

ここで $X_1, \ldots X_k$ は楽曲 (M の添え字)、$Y_1, \ldots Y_k$ はそれに対するフィードバック（Yes のとき $+1$, No のとき -1) とする。

プレイリストの重みづけは以下のように決定される ($N(L_i)$ はプレイリスト L_i の要素数)：

$$W(i) = \frac{1}{N(L_i)} \sum_{j=1}^{k} C_{i,X_j} Y_j$$

この重み付けでは、プレイリストに含まれる楽曲数が多いほど、ひとつの曲を Yes/No 判定したことによる影響が少なくなる。たとえば楽曲タグから生成されたおおまかなジャンルを示すプレイリストは要素数が多いので、曲選択に対しては低めの影響度を持つ。一方、人手で作成したリストは曲数が少ない傾向にあるので、もしそのリストに含まれている楽曲に対して Yes/No 判定が行われた場合は、リストの他の楽曲に対する影響は高い。

2.3 アンサンブル学習 (AdaBoost) による最適化

以上が基本的な曲選択アルゴリズムであるが、Yes/No で件数が増え、嗜好の指定が複雑になってくると、（1）で計算された重みづけ係数による曲の優先度計算が嗜好の指示と矛盾する場合が出てくる。

そこで、アンサンブル学習（AdaBoost[4]）による最適化計算を行う。アンサンブル学習は機械学習の手法であり、データを分類する関数を、与えられた例題を元に学習していく。その際、複数の弱仮説を生成し、その組み合わせによってより高い精度での推定を行う。AdaBoost では、各例題に重み $D_t(i)$ を与え、重み係数を含めた計算により弱仮説を生成する。そして、前段の弱仮説でうまく判定できなかった例題に相当する D を増やすことで、次段の弱仮説の計算にフィードバックを与える。

本方式では、Yes/No の履歴が例題に相当し、全楽曲を Yes か No に分類する作業と考えると、上記のアンサンブル学習の考え方を適用することができる。その際、前節で説明した重み付け計算方法を弱仮説として利用する。

今、前節と同様　$\{\{X_1, Y_1\}, \ldots, \{X_K, Y_K\}\}$ という嗜好のリストが与えられたとき、

ステップ１：

例題の重みを $D_1(i) = 1/K, i = 1, \ldots, K$ によって初期化する。

ステップ２：

$t = 1, \ldots, T$　に対して

$$W_t(i) = \sum_j D_t(j) C_{i,X_j} Y_j$$

と重み係数を計算し、各楽曲の優先度を

$$P_t(j) = \sum_{i=1}^{n} W_t(i) C_{i,j}$$

とする。弱仮説 H_t は

$$H_t(j) = \begin{cases} +1 & P_t(j) > 0 \text{ のとき} \\ -1 & P_t(j) <= 0 \text{ のとき} \end{cases} \quad (2)$$

その誤り率 ϵ_t を

$$\epsilon_t = \sum_{H \text{ が間違えた } i} D_t(i)$$

とし、信頼度 α_t を計算する：

$$\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$$

以下の式で分布 D_{t+1} を計算する：

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t Y_i H_t(X_i))}{Z_t}$$

(Z_t は $\sum_i D_{t+1}(i) = 1$　と規格化するための因子)

ステップ３：

最終的な楽曲優先度は、すべての仮説を信頼度で重み付けしたもの、すなわち

$$\sum_{t=1}^{T} \alpha_t P_t(x)$$

となる。

3 試作システム

以上で説明した方式に基づいた音楽再生システムを試作した。試作システムは二種類で、PC 上で動作するものと、情報家電に組み込まれたものである。

図 1 に PC 版試作システムの画面を示す。通常の音楽再生ソフトと同様、「再生」「停止」「ポーズ」などのボタンがあるが、それに加えて「Yes」「No」という嗜好を示すボタンがある。

利用者はまず適当なプレイリストを選択し、再生を開始する。最初の曲は選択されたプレイリスト中からランダムに選択される。プレイリストを選択せずに再生を開始すると、格納されている全楽曲からのランダム再生を開始する。利用者は、適宜 Yes, No ボタンを押して、現在再生されている曲

図 1: システム画面例

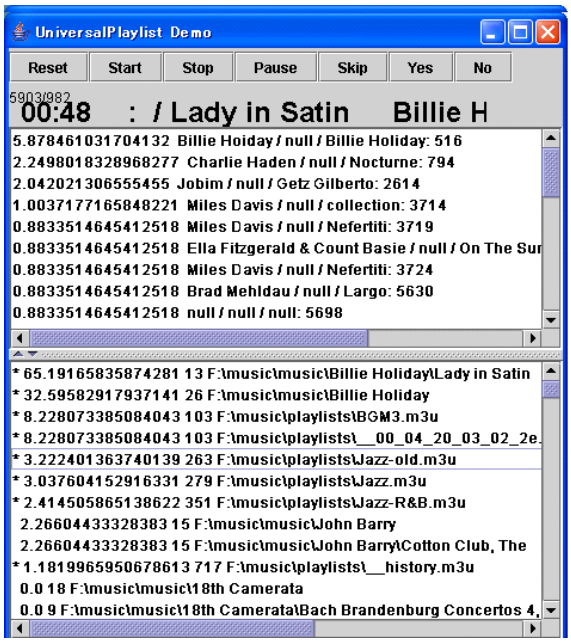

図 2: 嗜好調整の例１：ジャズ系の音楽に対して Yes, それ以外に対して No と答えた結果（15 件の判定後の画面例）

が気分に合っているかどうかをシステムにフィードバックする。No ボタンはスキップを兼ねているので、気分に合わない曲が再生された場合は、曲の終了を待つことなく次の曲に進むことが可能である。

このようにして Yes,No ボタンを押すことで、利

図 3: 嗜好調整の例２：クラシック系の音楽に対して Yes, それ以外に対して No と答えた結果（15 件の判定後の画面例）

用者の嗜好リストが形成されていく。リストが更新されるごとに、前節で説明したアルゴリズムによってプレイリストの重み付けを再計算する。嗜好リストは最近 30 件の Yes/No 判定を履歴として保存するようにしている。

曲の選択は、前節で説明したアルゴリズムによって動的に変更されるので、選ばれる曲は最初に指定したプレイリストに含まれるものに限定されない。格納されている全楽曲の中で優先度の高くなったものが選択されていく。

画面の上部リストには、本アルゴリズムによって選択された楽曲が、下部リストには各プレイリスト一覧と、その重みが表示されている。利用者は、これらのリストから楽曲ないしプレイリストを選択し、再生開始、Yes/No の指定などを指示することもできる。

図 1 は、再生開始直後の様子を示している。プレイリストを指定せずに再生を開始しているので、楽曲が単にランダムに選択されているのがわかる。図 2 は、Yes/No 判定を 15 回程度繰り返した後の画面である。この場合は、ジャズ系の音楽が再生されると Yes, それ以外の場合 No と指示するようにした。画面上でもその効果が確認できる。図 3 は、クラシック系の楽曲を嗜好して Yes/No 判定を行った結果である。図 2 と明らかに異なった傾

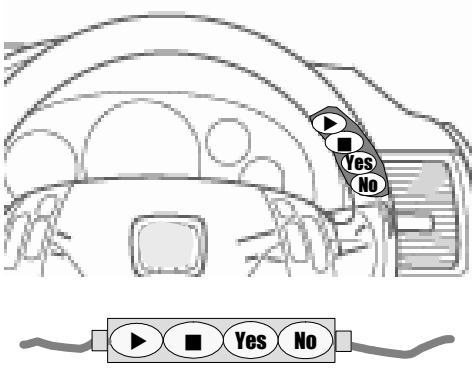

図 4: 音楽再生インタフェースの搭載可能性 (上：車載用、下：携帯プレイヤーのリモコン)

向の選曲がなされている。

なお、聞いている途中で気分が変化した場合は、RESET を押して過去の Yes/No 履歴をフラッシュして再生を継続することができる。また RESET をせずに、単に Yes/No の判定基準を変えて再生を継続しても、再生楽曲の選択は次第に適応していく。

3.1 情報アプライアンスとしての実現

また、リビングルームでの音楽再生に対応させるために、Slim Devices 社のネットワーク音楽再生アプライアンスである SqueezBox[3] に本方式を適用した。SqueezBox はネットワークに接続して MP3 などの音楽フォーマットをストリーム受信して音楽を再生するアプライアンス（情報家電）製品である。楽曲ファイルはネットワーク上のサーバーシステムに格納されており、楽曲のストリーミングもサーバーシステム上のプロセスが担当する。本体にも簡易ディスプレイが搭載されており、リモコン信号を受信してサーバーに送る機能を持っているので利用者はサーバーを直接意識することなく、SqueezBox に対して直接リモコンを使った操作が可能である。

本試作では、リモコンの使用されていないボタン二つを Yes、No に割り当てた。SqueezBox のサーバーでは、リモコンキーに対応した Perl スクリプトが起動されるような構成をとっているので、ここにプレイリスト計算アルゴリズムを実装した。

他の適用例としては、嗜好を指示するインタフェースが極めて単純なので、数個のボタンのみで操作可能な携帯音楽再生デバイス用のリモコンや、車載のオーディオシステムのインタフェースへの搭載が可能だと考えている（図 4）。

楽曲総数	5903
プレイリスト総数	982
(利用者が生成)	67
(タグ情報より生成)	37
(ディレクトリ構造より生成)	868
(楽曲分析より生成)	10
プレイリスト中の平均楽曲数	28.1

表 1: 使用環境

4 利用者経験

上記のシステムを筆者の所有する楽曲ファイルについて適用し、日常の利用環境で試用を行っている。利用する場面は、主に移動中（ノート PC による操作）、自宅（情報アプライアンスによる操作）である。表 4 に使用環境を示す。

現状では複数の利用者による公式な利用者評価は行っていないが、以下で挙げるような知見が得られている。

- プレイリストの作成意識の変化： 従来型のプレイリストは、それに含まれる楽曲を直接再生する目的で作成するので、曲目数は少なめなものが多かったが、本システムで利用に慣れるに従い、ある「観点」で楽曲をまとめたものを作成するようになった。とくに音楽ジャンルをまたがったグループを作成することで、タグによるジャンル情報を補完することが可能である（クラシックもジャズも含む「ピアノ曲」「ボーカル」といったグループ）。

- 曲の嗜好はポジティブなものよりネガティブなものが明確であった。提示された曲に対して、「これは今聞きたくない」というネガティブな判定は（たとえば夜なのでうるさい曲は聞きたくない、時間が限られているのに長い曲は聞きたくない、などの理由で）かなり明確に下せたが、「これを聞きたい」という判定基準はそれほど明確ではなかった。 UniversalPlaylist のような嗜好調整型のインタフェースに対して、人はどのようなタイミングどんな指示を与えるのか、は今後継続して研究する必要性を感じている。

- 選曲のランダム性に関する調整： 上記のアルゴリズムを厳密に適用し、優先度の最も高いものから再生するようにすると、過去の Yes/No 判定に正確に追従しすぎて、逆に「選曲の意外性」が感じられなくなる場面があった。たとえば特

定のアルバムの曲のみが連続再生されるような場合などがそれに相当する。これに対しては、現在、以下のような楽曲選択のランダム性の導入を検討している：

プレイリストの重み係数が計算された後、全楽曲から N 曲をランダムに選び出し、その中から前節の計算手法で最も高い優先度を持つものを次に再生すべき曲とする。N が全楽曲数の場合は全楽曲中で最も優先度の高い曲になり、N が１の場合は単純なランダム選曲になるので、N の値を変化させて選曲の「ばらつき」を調整することができる。この値を、たとえばボリュームやスライダー、あるいは携帯機器全体をシェイクするなどの操作で利用者が制御することを想定している。

5 関連研究

本研究で提案する UniversalPlaylist は、ネットワーク販売などで広く使われているソーシャル・フィルタリング [8] の応用であると考えられる。本方式の特徴は、すでにデジタル音楽再生で利用されているプレイリストの概念との親和性が高いこと、再生中に嗜好のパラメタが動的に変化し、それに伴って選曲の戦略も動的に適合していく点にある。また、ソーシャル・フィルタリングと異なり、「他人の好み」との比較ではなく、自分の好みの様々な局面が、異なったプレイリストとして保管されており、その組み合わせによって現時点での嗜好を表現している点も特徴である。

MoodLogic[7] のように、楽曲情報データベースをさまざなな検索パラメタを解して利用者の嗜好に合ったプレイリストを生成するサービスが提供されている。パラメタにはジャンル、テンポ数、また各種の「気分」メニュー (“Agressive”, “Upbead”, “Happy”, “Sad” など）が含まれる。これらのシステムでは、検査パラメタを調整するためのユーザ操作の複雑性、再生中に動的に嗜好を調整するための手段などが課題となる。

楽曲自体の特徴量から、楽曲間の類似度を計算し、音楽検索に応用する研究が多く行われている [1, 6]。類似度を利用することで、幾つかの楽曲の例を示して、それに「似た」曲のリストを生成することが可能である。本方式は、このようにして生成されたリストも含めて利用することが可能である。現時点で提案されている類似度計算の水準では、一定以上の品質を超えることが困難であるという調査結果 [2] もあり、現状では自動生成リストと人手によるリストを組み合わせられる枠組みがより現実的であると考えている。

MeltingSound[9] は、楽曲を 2 次元平面上に配置しておき、マウスの位置に近い音楽が自動的に再生されるインタフェースを提供している。趣味にあった音楽を 2 次元平面上で近くに配置していくことで、マウスの位置に応じて再生される音楽の種類を選別することができる。この方法は、2 次元平面に楽曲を配置する判断を利用者に委ねているので、配置すべき曲の数が増えるに従い、その作業が負担となる。また、複数の異なる観点からの配置が困難である。さらに、ある程度の面積の画面表示と 2 次元ポインティングデバイスが必要なので、リモコンなどへの搭載が困難である。

6 議論・今後の課題

プレイリスト間の類似度

提案方式では、楽曲ファイルとプレイリスト間の関係のみによって優先度を計算している。したがって、ある曲に対して判定を行った場合、それを含まないプレイリストの重み係数は変化しない。一方、プレイリスト間でも含まれる楽曲には重複があるので、たとえば以下のような計算式によってプレイリスト間の「類似度」を判定することができる:

$$Sim(L_i, L_j) = \frac{1}{2}(\frac{N(L_i \cap L_j)}{N(L_i)} + \frac{N(L_i \cap L_i)}{N(L_j)})$$

($N(L)$ は L に含まれる楽曲の個数)。計算された類似度を使って、あるプレイリストの重みが変化した場合に、それに「似ている」プレイリストの重みも類似度に従って変化させることが可能になる。この方式では、曲を明示的に含まないプレイリストの重みも変化するので、曲の優先度がより広範囲で変化することになる。

ネットワーク上でのプレイリスト流通

オンライン音楽販売の普及に伴い、ネットワーク上で自分のプレイリストを公開・交換することが可能になった。通常、他人のプレイリストに含まれる楽曲をすべて所有しているとは限らないので、このプレイリストをそのまま利用することは必ずしも容易ではないが、本方式では他人のプレイリストを曲間の関連づけ情報として利用することが可能である。

携帯機器の自動音楽インストール

携帯音楽再生機器の最大容量を超える音楽を所有している利用者の場合、単純に機器に「すべてをダウンロード」することはできない。何らかの取捨選択が必要である。通常、これは人手を要する作業だが、本提案方式によって「趣味に合った10 ギガバイト分をダウンロード」という自動化が可能になる。

また、携帯電話や PDA のように外部との通信機能を有する携帯機器では、利用者の嗜好に適合して動的に楽曲をダウンロードすることが可能である。本体にはプレイリスト情報と、ある程度の楽曲格納領域を用意しておき、利用者の嗜好の変化に追従して自動的にダウンロードを行うことで、仮想的には全楽曲を保有しているのと似た再生環境を提供することができる。

他のメディア再生への適用可能性

最後に、提案方式を音楽再生以外の場面で適用する可能性について考察する。デジタルカメラの普及にともない、従来とは比較にならないほど簡便に写真を撮影・記録することが可能になったが、その再生（表示）方式についてはあまり発展がない。「記憶する住宅」[5] では、住居の壁面に、蓄積された画像を常時ランダムに表示する方式を提案しているが、そのときの気分に合った画像をどうやって表示するか、についての手段は提供されていない。

本提案方式を写真再生に適用すると、Yes/No などの簡単な指定のみで表示戦略を利用者の興味に適合させることができる。指示にはディスプレイにボタンを装備する、リモコンで指令する、などが可能だが、ディスプレイに人感センサを装備しておき、人がディスプレイの前に立ち止まっていると、そのときに表示されている写真に関するプレイリストの評価が上がるようにすることも可能ではないかと考えている。

7 結論

音楽などの多量のコンテンツを、利用者の操作負担を最小限に抑えつつ、利用者のその時々の嗜好に合致して再生するための手法、UniversalPlaylist について提案した。「複数のプレイリストを集合として扱い、プレイリストの評価重み付けを利用者の嗜好に合わせて動的に変更していく」という発想に基づいている。利用者は Yes/No などの簡単な指定を繰り返すだけで、コンテンツの属性情報や検索パラメタなどの詳細に立ち入ることなく気分にあった再生を行うことができる。インタフェースが非常に単純なので、携帯音楽プレイヤーのリモコンなどでも容易に操作することができる。

提案方式を実現した音楽再生システムを試作し、非公式ではあるが良好な結果を得た。今後は、複数の利用者によるユーザ評価、および、嗜好の変化の長期的な記録によって、人がどのような場面でどのように嗜好を変化させていくのか等の研究を行っていきたい。

参考文献

[1] J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR 2002*, pp. 157–163, 2002.

[2] J. Aucouturier and F. Pachet. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, Vol. 1, No. 1, 2004.

[3] SLIM DEVICES. SqueezeBox: WiFi for your HiFi! http://www.slimdevices.com.

[4] Y. Freund and R.E. Shapire. A desision-theoretic geeralization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119–139, 1997.

[5] Y. Kono and K. Misaki. Remembrance Home: Storage for re-discovering one's lifetxt. In *Proc. of Pervasive 2004 Workshop on Memory and Sharing of Experiences*, 2003.

[6] B. Logan, D. Ellis, and A. Berenzweig. Towards evaluation techniques for music similarity. In *Workshop on the Evaluation of Music Information Retrieval Systems at SIGIR-03*, 2003.

[7] MoodLogic. http://www.moodlogic.com.

[8] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating " word of mouth ". pp. 210–217, 1995.

[9] 神原啓介, 安村通晃. Meltingsound：なめらかなオーディオブラウジング. HIS (ヒューマンインタフェースシンポジウム) 2003 論文集, pp. 817–820, 2003.