

楽曲内容に基づく楽曲推薦 -TwinVQを用いた-

鈴木 誠知 小早川 倫広 星 守 大森 匡

電気通信大学大学院情報システム学研究科

1 はじめに

現在、様々な音楽配信・検索サービスが展開されている。それらの多くの楽曲検索システムでは、楽曲を特定するためのキーワード、または楽曲の一部などを入力することで目的の楽曲を検索している [1, 2, 3]。さらに自分の好みに合った楽曲を検索したい、この曲のような楽曲を検索したいという要求もある [4, 5, 6]。この検索では“正しい答え(曲)”が、ユーザごとに異なるという難しさがある。

本稿では、TwinVQを用いた楽曲内容に基づく楽曲推薦システムの枠組を示し、楽曲推薦システムで用いるためのリズムによる検索手法について述べる。

2 楽曲推薦システムの枠組

図1に示す楽曲推薦システムの枠組を、データベース登録部、ユーザ入力部、楽曲検索部の順に説明する。

データベース登録部 データベース登録部では、データベースに登録する楽曲に対して、TwinVQ エンコーダを適用し圧縮データを生成する。圧縮時に算出されるパラメータ (MDCT 係数, 自己相関係数, 線形予測係数, LPC ケプストラム, LSP パラメータ, LPC スペクトル) から、リズム特徴, 音域特徴などの楽曲特徴量を抽出し、楽曲特徴量による索引を構築する。楽曲を登録する際に、Gracenote 社から取得したタイトル名などのメタデータもデータベースに登録する。

ユーザ入力部 ユーザは問い合わせとして2種類の情報をシステムに入力する。1つめに、ユーザが好む楽曲の気に入っている一部分の楽曲データを入力する。2つめに、リズム, 音域などのコンテンツごとに設けられたボタンを選択することで、その楽曲のどのような楽曲特徴を気に入っているかを入力する。

楽曲検索部 システムはユーザが入力した楽曲データに対してTwinVQ エンコーダを適用し、様々なパラメータを抽出する。そのパラメータからリズム特徴, 音域特徴などの楽曲特徴量を抽出し、各特徴量ごとにデータベース内の楽曲との類似度を算出する。最後に楽曲特徴量ごとに算出した類似度を元に、推薦曲リストを得る。

3 リズムによる検索

楽曲の基本的なリズムは、時間軸上の音の長さや強さの規則的な並びであり、それはパワーのピークの間隔と高さから読み取ることができる。そこで低次の自己相関係数が、楽曲のフレームのパワーの近似と見なせることから、リズム検索に用いるパラメータとして自己相関係数に着目する。

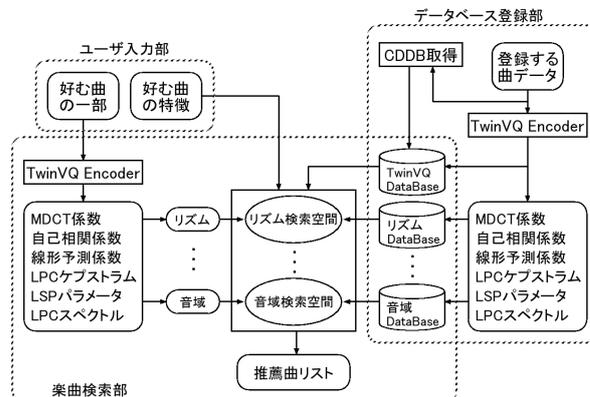


図1: 楽曲推薦システムの枠組

自己相関係数によるリズム検索の手続きを図2に示す。ユーザが入力した楽曲データにTwinVQ圧縮を適用することで得られる自己相関係数と、楽曲データベースの自己相関係数との非類似度を求め、昇順にソートすることで自己相関係数の概形が近い楽曲を検索し、推薦曲リストが得られる。

4 リズムによる検索実験

自己相関係数によるリズム検索の有効性について検証するため、図2の手続きにおいて、 $a = 10$ として検索実験を行った。実験データ集合として、23種類のジャンルからなる3059曲を用意した。次に、23種類のジャンルの内、CDセット6種類 (Chanson, Dance, Jazz, Tango, Thai(タイの音楽), TUBE(邦楽グループ)), リズムに特徴のあるジャンル4種類 (HipHop, R&B, Reggae, Rock)。ただし、これらの分類はGracenoteに依存)の10種類のジャンルの楽曲から10曲ずつ、計100(10×10)曲を問い合わせとして選択した。実際の実験では、楽曲100曲からサビと思われる部分10秒程度を手作業で抽出し、それら区間の自己相関係数列を入力した。また、非類似度関数として、次の2種類の非類似度, dis_sim_1 , dis_sim_2 を用いて検索を実行した。

$$dis_sim_1(\mathbf{R}_k^q, \hat{\mathbf{R}}_k^m) = d_2 \left(\frac{\mathbf{R}_k^q}{\frac{1}{W} \sum_{i=1}^W |R_{k,i}^q|}, \frac{\hat{\mathbf{R}}_k^m}{\frac{1}{W} \sum_{i=1}^W |\hat{R}_{k,i}^m|} \right),$$

$$dis_sim_2(\mathbf{R}_k^q, \hat{\mathbf{R}}_k^m) = 1 - \frac{\sum_{i=1}^W (R_{k,i}^q - \bar{R}_k^q)(R_{k,n+i-1}^m - \bar{R}_k^m)}{\sqrt{\sum_{i=1}^W (R_{k,i}^q - \bar{R}_k^q)^2 \sum_{i=1}^W (R_{k,n+i-1}^m - \bar{R}_k^m)^2}}$$

ただし、 $d_2(x, y)$ は x, y 間のユークリッド距離、 \bar{x} は x の平均を表す。

```

Procedure retrieval_by_rhythm( $D, q, a$ )

input  $D$ :  $M$  曲からなる楽曲データベースの自
己相関係数列  $R_k^m$  ( $m = 1, \dots, M$ ,  $k =$ 
 $0, 1, \dots, 20$ ) の集合
 $q$ : ユーザが入力する楽曲データ
 $a$ : 出力する楽曲数

output  $L$ : 推薦曲リスト

// Step 1 TwinVQ エンコード //
•  $q$  に対して, TwinVQ エンコーダを適用 .
• 自己相関係数列  $R_k^q = R_{k,1}^q, \dots, R_{k,W}^q$  を抽出 ( $W$ 
は自己相関係数列の長さ) .
// Step 2 非類似度算出//
for  $m \leftarrow 1$  to  $M$  do
  for  $n \leftarrow 1$  to  $N_m - W + 1$  do
     $R_k^q$  と  $\hat{R}_k^m = R_{k,n}^m, \dots, R_{k,n+W-1}^m$  間の
    非類似度  $dis\_sim(R_k^q, \hat{R}_k^m)$  を算出 .
     $Dis\_sim(R_k^q, R_k^m) \leftarrow$ 
     $\min_{n=1}^{N_m-W+1} dis\_sim(R_k^q, \hat{R}_k^m)$  .
  // Step 3 推薦曲リストの出力//
  非類似度  $Dis\_sim(R_k^q, R_k^m)$  を昇順ソートし, 上位
   $a$  曲の推薦曲リスト  $L$  を出力.
end procedure

```

図 2: リズム検索の手順.

表 1: 検索結果の評価

	dis_sim_1		dis_sim_2	
	+	×	+	×
Chanson	3.50	5.50	4.10	4.90
Dance	4.90	4.10	4.80	4.20
HipHop	4.40	4.60	3.90	5.10
Jazz	2.90	6.10	3.00	6.00
R&B	3.80	5.20	4.30	4.70
Reggae	4.60	4.40	5.00	4.00
Rock	3.90	5.10	4.50	4.50
Tango	1.10	7.90	2.50	6.50
Thai	2.80	6.20	2.70	6.30
TUBE	1.60	7.40	2.90	6.10
平均	3.35	5.65	3.77	5.23

5 評価方法

2 種類の非類似度により検索された楽曲がリズムによる推薦曲として妥当かどうか検証する. 各問い合わせに対する検索結果の上位 10 曲中, 2 位から 10 位が問い合わせ曲 (1 位) とリズムが似ているかどうか, 試聴により次の 3 通りで評価する. : 問い合わせと非常に良く似ている. : まあ似ている. ×: 似ていない.

実際に, 各問い合わせに対する検索結果の 2 位から 10 位までを試聴し, × で評価する. 次に × が割り振られた曲数を求める. ここで, も推薦曲の候補として考えられるため, または となったものを + として集計する. 最後に同じジャンルに属する曲全体の +, × に対しての平均を求め, 表 1 を作成した.

6 考察

表 1 を用いて考察する. 非類似度 dis_sim_1 では, + の最小値は Tango 行の 1.1 である. 最小となった Tango では 9 曲中 1 曲程度しか妥当な楽曲が推薦されないことになるが, 3059 曲のデータ集合の中に Tango は高々 100 曲しか含まれていないことから, 決して悪い結果ではないと考えられる. + の最大値は, Dance 行の 4.9 である. これは, Dance に分類されている楽曲が, 検索結果の 9 曲中に 約 5 曲問い合わせ曲のリズムに似ている楽曲として検索されていたことを意味する. また, 検索結果の上位 9 曲中 4 曲以上に + が付加されていたジャンルは Dance, HipHop, Reggae であった. これらに共通する点はドラムやベースなどのビートが強い楽曲が多いことである. すなわち, 自己相関係数列を用いたリズム検索が有効であったことを示している. + の列平均は, 3.0 以上であった. このことは, 9 曲中 3 曲程度は, 推薦されても違和感の少ない曲であったことを意味している.

非類似度 dis_sim_2 では, Dance, Reggae に対しては dis_sim_1 と同等の結果が得られた. さらに, Chanson, Tango, TUBE などのジャンルでの成績向上が見られた. 平均の行を見ても dis_sim_2 の方が, dis_sim_1 より良い成績となっている. このことは, dis_sim_2 もまたリズム検索に対して有効であったことを示している.

7 まとめ

本稿では, 楽曲内容に基づく楽曲推薦システムの枠組を示した. 本枠組の特徴は, TwinVQ のエンコーダを特徴量抽出器として用い, TwinVQ 圧縮, 音楽特徴量抽出の双方を楽曲圧縮の枠組の中で行うというものであった. さらに, 楽曲推薦システムで用いるためのリズム検索手法について述べ, リズムによる検索実験を実行した. 検索結果, ビートが強い楽曲に対しては有効であることを確認した. 本稿では, その他の検索の枠組, 検索結果については記載しなかったが, 発表では, その他の検索についても紹介をしたい.

参考文献

- [1] K. Onishi, M. Kobayakawa, M. Hoshi, and T. Ohmori. A feature independent of bit rate for twinVQ audio retrieval. In *Proc. ICME2001*, pp. 293–296, 2001.
- [2] M. Kobayakawa, M. Hoshi, and K. Onishi. A method for retrieving music data with different bit rates using mpeg-4 twinVQ audio compression. In *Proc. ACM MM*, 2005.
- [3] M. Kobayakawa and M. Hoshi. A partial retrieval of music data with different bit rate using mpeg-4 twinVQ audio compression. In *Proc. MDDM2007*, 2007 (to appear).
- [4] Simon Dixon, Fabien Gouyon, and Gerhard Widmer. Towards characterisation of music via rhythmic patterns. In *Proc. ISMIR2004*, 2004.
- [5] K. Hoashi, K. Matsumoto, and N. Inoue. Personalization of user profiles for content-based music retrieval based on relevance feedback. In *Proc. ACM MM*, pp. 110–119, 2003.
- [6] Daniel P.W. Ellis. Extracting information from music audio. *Comm. of the ACM*, Vol. 49, No. 8, pp. 32–37, 2006.