Vol.2012-MUS-94 No.31
Vol.2012-SLP-90 No.31
2012/2/4

類似度に基づくポリフォニックな楽曲の分類

阿南陽子[†1]　畑埜晃平[†1]
坂内英夫[†1]　竹田正幸[†1]

本稿ではポリフォニックな楽曲の分類問題に取り組む．特にデータ間の非類似度に基づいた分類手法 [1] を用いる．ポリフォニックな楽曲に文字列間の非類似度関数を適用するため，楽曲データをベクトル列に変換する前処理を行う．前処理と非類似度関数を変えながら分類実験を行ったところ，どの場合でも高い分類精度が得られた．

Polyphonic Music Classification Based on Similarity

Yoko Anan,[†1] Kohei Hatano,[†1] Hideo Bannai[†1]
and Masayuki Takeda[†1]

In this paper we address the problem of classifying polyphonic music. Especially we use the classification approach based on dissimilarity [1]. We convert polyphonic music into vector sequence in order to apply the dissimilarity function used for string to music. When classified by changing vector sequence and dissimilarity function, classification accuracy was good at any case.

1. はじめに

楽曲の分類には多くの既存研究がある（例えば [2,3] を参照）．その 1 つは特徴ベクトルを用いる方法である．特徴ベクトルを用いた分類は機械学習の分野でよく使われており，楽曲の分類に関して，performance alphabet [4] 等，多くの特徴ベクトルが考案されている [2,5–7]．しかし楽曲の分類に有効な特徴を見つけることは簡単ではなく，精度の高い分類を行うことは難しい．

特徴ベクトルの他に楽曲の分類で有名な手法はカーネルと組み合わせたサポートベクターマシン (Support Vector Machines; SVMs) を使う方法である [8,9]．この手法では，楽曲データの分類に有効なカーネルが設計できれば，精度の高い分類が行える．しかしカーネルを設計する際，類似度関数が非負の半正定値という条件を満たさなければならないという制約がある．

著者ら [1] は，Wang らの方式に基づいて新たな分類手法を提案した．この手法では，正例と負例の対を弱分類器とし，各弱分類器は，未知の例と正例・負例との非類似度を求めて，正例に近ければ正，負例に近ければ負と判定する．弱分類器が判定した結果の重み付き多数決によって強分類器を構成する．この手法の利点は，任意の非類似度関数を使用できるという点である．すなわち，非負の半正定値といった制約を気にせずに非類似度関数を設計／選択できる．

同時に鳴る音が高々1 個であるような楽曲をモノフォニックといい，そうでない楽曲をポリフォニックという．文献 [1] では，モノフォニックな楽曲の分類問題にこの手法を適用して高い精度が得られることを示した．そこで本稿では，ポリフォニックな楽曲の分類問題に取り組む．ポリフォニックな楽曲間の非類似度を求める方法としては，楽曲をクロマベクトルの系列とみなして文字列間の非類似度関数を用いる方法，クロマベクトルを 6 次元の実数値ベクトルの系列 [10] に変換して非類似度を求める方法，あるいは，さらに実数値ベクトルを 2 値化してから非類似度を求める手法 [11] などが提案されている．本稿では，楽曲データの前処理と文字列間の非類似度関数の組合せを変えながら，分類精度を比較する．変奏曲の分類や JPOP と演歌のジャンル分類の問題に適用したところ，高い分類精度が得られた．

本稿の構成は以下のとおりである．2 節では非類似度関数を用いた分類手法について説明する．3 節では楽曲に対する前処理と非類似度関数に関する既存研究を概観する．4 節ではベクトル列に非類似度関数を適用する方法を説明する．5 節では変奏曲の分類および JPOP と演歌のジャンル分類の実験結果について述べる．6 節ではまとめと今後の課題を述べる．

2. 非類似度関数を用いた分類

この節では，非類似度関数を用いた分類手法 [1] の概略を説明する．この手法は，Wang ら [12] の提案したスキームに沿ったもので，正例と負例の対を弱分類器とした重み付き多数決によって強分類器を構成する．各弱分類器は，未知の例に対し正例および負例との非類

†1 九州大学大学院システム情報科学府
Department of Informatics, Kyushu University

Vol.2012-MUS-94 No.31
Vol.2012-SLP-90 No.31
2012/2/4

似度をそれぞれ求め，正例に近ければ正，負例に近ければ負と判定する．したがって，この手法の成否は非類似度関数に強く依存する．

2.1 分類精度に関する理論的保証

本手法は，Wang ら [12] が示したように,「良い」非類似度関数を使うと高い確率で十分な分類精度が得られるという理論的保証をもつ.「良い」非類似度関数の定義を以下に示す．X を事例空間とし，d を $\mathrm{X} \times \mathrm{X}$ から $\mathbb{R}^+$ への非類似度関数とする．事例 $x \in \mathrm{X}$ とラベル $y \in \{-1, 1\}$ の組 (x, y) を例と呼ぶ．各例は確率分布 P に従ってランダムかつ（条件付きで）独立に現れるものとする．

定義 1 ([12]) 非類似度関数 d が P **に関して** (ϵ, η)**-適合** であるとは，ラベル付き例の集合 $\mathrm{X} \times \{+1, -1\}$ から P に従って $z = (x, y)$ を選ぶと $1 - \epsilon$ 以上の確率で次が成り立つときをいう．

$$\Pr_{z', z'' \sim \mathrm{P}}(d(x, x') < d(x, x'') | y' = y, y'' = -y) \geq 1/2 + \eta/2 \quad (1)$$

ここで，$z' = (x', y')$, $z'' = (x'', y'')$ は P に従って選ぶものとする．

定義から，良い非類似度関数 d の下では，ほとんどの例 $z = (x, y)$ に対し，事例 x と他の事例との非類似度は，ラベルが同じ場合ラベルが異なる場合に比べ高い確率で小さくなる．

P に関して (ϵ, η)-適合な関数については，以下の定理が成り立つ．

定理 1 ([12]) 非類似度関数 d が P に関して (ϵ, η)-適合であるならば，例の組 (z'_i, z''_i) を $y'_i = 1$，$y''_i = -1$ という条件付きで確率分布 P からランダムに $n = (4/\eta^2) ln(1/\delta)$ 組選ぶとき，$1 - \delta$ 以上の確率で次が成り立つ．

$$\Pr_{z', z'' \sim \mathrm{P}}(yf(x) \leq 0) \leq \epsilon + \delta.$$

ここに，

$$f(x) = \frac{1}{m} \sum_{i=1}^{n} \mathrm{sgn}[d(x, x''_i) - d(x, x'_i)]$$

とし，sgn は，$a > 0$ のとき $\mathrm{sgn}[a] = 1$，そうでないとき $\mathrm{sgn}[a] = -1$ とする．

この定理から，ランダムに選択された例が十分存在すれば，(重み無し) 多数決によって高い確率で精度の高い分類が得られることがわかる．さらに， Wang ら [12] は「良い」非類似性関数の定義を緩和したものを示し，その下でも重み付き多数決によって同様の結果が得られることを示している．

2.2 1-ノルムソフトマージン最適化

上に述べたように，確率分布 P に関して「良い」非類似度関数 d が与えられたと仮定すると，十分な数の正例と負例の対をランダムに選べば，高い確率で高い精度が得られることが保証される．

次に，弱分類器を結合する際の重みをどう求めるかが問題となる．論文 [1] では，この問題を 1-ノルムソフトマージン最適化と捉え，LPBoost によって重みを求める手法を提案している．1-ノルムソフトマージンの最適化は，汎化誤差の理論的保証をもつばかりでなく，重みベクトルが疎になりやすいことが知られており，簡潔な仮説が得られることが期待できる．

3. 楽曲間の非類似性指標 (既存研究)

上述の分類手法を用いて楽曲の分類を行う際には，楽曲間の非類似性をいかに与えるかが成功の鍵となる．文字列処理の分野では，編集距離をはじめとする様々な文字列間の非類似度関数が提案されており，遺伝子情報の検索や解析などに応用されている．こうした文字列間の非類似度関数を楽曲に適用する際には，前処理として，楽曲データ (楽譜データ，演奏データ等) を何らかの系列データに変換することになる．すなわち，前処理を行う関数を p，文字列間の非類似度関数を δ とするとき，楽曲 x, y 間の非類似度 $d(x, y)$ は，$d(x, y) = \delta(p(x), p(y))$ の形で与えられる．

以下では，前処理 p と文字列間非類似度関数 δ の組合せについて，既存の研究を概観する．

3.1 前 処 理

3.1.1 音符休符列・音高コンツァ

Mongeau と Sankoff [13] は，楽譜データにおいて，音符および休符を 1 つの文字とみなし，モノフォニックな楽曲を音符と休符から成る 1 本の文字列と扱っている．また音高コンツァとは，音高の変化のパターンを表す文字列で，変化のタイミングは無視している．

3.1.2 音高文字列

時間軸を一定幅の区間に区切り，各区間で鳴っている音に着目すると，楽曲データから音高の集合の列が得られる．Kadota ら [14] は，時間軸を 16 分音符単位に区切り，各音高集合中で最も高い音を選ぶこと (スカイライン方式) により，楽曲を音高の文字列に変換している．オクターブの違いは無視しているため音高は 12 種類となり，これに無音を表す記号を加えた合計 13 個の記号の集合をアルファベットとする**音高文字列**が得られる．

3.1.3 リズム文字列

Kadota ら [14] は，上述の音高文字列に加え，音の鳴っている区間については先頭 (N) かそれ以降 (n) かを区別し，音の鳴っていない区間については先頭 (R) かそれ以外 (r) かを区別することによって，楽曲をサイズ 4 のアルファベット上の文字列に変換し，これを**リズム**

Vol.2012-MUS-94 No.31
Vol.2012-SLP-90 No.31
2012/2/4

文字列と名付けている．

3.1.4 クロマベクトル列

オクターブの違いを無視した 12 の音高を次元とし，次元ごとの「強さ」を実数値で表現した 12 次元の実数値ベクトルを**クロマベクトル**と呼ぶ．楽曲はクロマベクトルの系列とみなすことができる．特に，各次元の値を 0,1 に限定したクロマベクトル列を **2 値クロマベクトル列**とよぶ．

3.1.5 TC ベクトル列

Harte ら [10] は，クロマベクトルを 6 次元の実数値ベクトルに変換する調性重心 (tonal centroid; TC) 法を提案している．任意のクロマベクトル $\mathbf{c} = {}^T[c_0, \ldots, c_{11}]$ に対して，対応する TC ベクトル $tc(\mathbf{c})$ を以下で定める．

$$tc(\mathbf{c}) = \frac{1}{\|\mathbf{c}\|_1}[\phi_0, \ldots, \phi_{11}]\mathbf{c},$$

ここで，$\|\cdot\|$ は L1-ノルムを表し，また，$p = 0, \ldots, 11$ に対し，

$$\phi_p = {}^T\left[\begin{array}{cccccc} \sin\frac{7\pi}{6}p & \cos\frac{7\pi}{6}p & \sin\frac{3\pi}{2}p & \cos\frac{3\pi}{2}p & \frac{1}{2}\sin\frac{2\pi}{3}p & \frac{1}{2}\cos\frac{2\pi}{3}p \end{array}\right]$$

とする．こうして得られるベクトルを **(実数値)TC ベクトル**とよぶ．TC ベクトルへ変換する利点は，3 度や 5 度などの協和音に対して，ユークリッド距離が小さくなる点にある．

3.1.6 2 値 TC ベクトル

Ahonen ら [11] は，2 値クロマベクトルから得た TC ベクトルに対し，各次元ごとに可能なすべての値の中央値を閾値として 2 値化するアイデアにより，サイズ $2^{12} = 4096$ のアルファベット上の文字列である 2 値クロマベクトルをサイズ $2^6 = 64$ のアルファベット上の文字列へ変換している．

3.2 文字列間の非類似性指標

文字列間の非類似性指標としては，**編集距離 (edit distance)** がよく知られている．これは，一方の文字列を他方の文字列へ変換するのに必要な編集操作の適用回数の最小値として定義され，編集操作としては，文字の置換・削除・挿入の 3 種類が用いられる．**重み付き編集距離 (weighted edit distance)** は，編集操作に重みとして実数値を関連づけ，適用した編集操作の重みの和の最小値として定義される．重みの値は操作に関わる文字に依存して決まる．

編集距離と関係の深い類似性指標として，**最長共通部分列**(Longest common subsequence; LCS) の長さがよく用いられる．その拡張として，文字の一致の具合を実数値重みで与えたものを，本論文では便宜上**重み付き LCS** とよぶことにする．

文字列カーネルは，文字列間の類似性指標である．主なものとして，n-グラムカーネル [15]，ミスマッチカーネル [9] などが知られている．

文字列 x, y 間の正規化情報距離 (Normalized Information Distance; NID) は，次式で与えられる．

$$NID(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}.$$

ここに，$K(\cdot)$ は Kolmogorov 記述量，$K(\cdot|\cdot)$ は条件付き Kolmogorov 記述量である．この式において，$K(x|y) \approx K(yx) - K(y)$ とし，さらに，Kolmogorov 記述量を適当な圧縮プログラムによる圧縮データサイズで代用したものが，**正規化圧縮距離 (Normalized Compression Distance; NCD** である．圧縮プログラム C による文字列 z の圧縮サイズを $C(z)$ で表すとき，文字列 x, y 間の NCD は，以下で定義される．

$$NCD_C(x, y) = \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}}.$$

通常 $C(yx)$ を $C(xy)$ で置き換えた式が用いられるが，本論文では，$NCD_C(x, y) = NCD_C(y, x)$ とするために上式を用いることにする．

3.3 楽曲間に特化した非類似度関数

Mongeau と Sankoff [13] は，音符休符列に重み付き編集距離を適用した．その際，新たな編集操作として，1 つの音符を連続した複数音符に対応させる「分割」操作，およびその逆の「統合」操作を加えている．また，音符の「置換」操作にかかわる重みは，音高の協和／不協和の度合と音長に基づいて与えている．この手法は，同時に複数の音符が存在しないことを前提としている．そして，モーツァルトのキラキラ星変奏曲を対象とし，各変奏の主題からの距離を定量化する問題へ適用している．

[14] では，音高文字列やリズム文字列に対し，SRS (String Resemblance Systems) [16,17] と名付けたフレームワークを用いて，直感的な類似性指標を三つ提案しその有効性を示している．

Vitanyl ら [3] は，余計な情報を削ぎ落とした MIDI ファイルを対象に，NCD を用いた楽曲のクラスタリングを行った．圧縮プログラムとしては，gzip と bzip2 を用いている．

Li と Sleep [18] は，音高コンツァーを対象に，NCD を用いて楽曲間の距離を求め，k-NN を用いて楽曲の分類を行っている．実際の圧縮プログラムによる圧縮ファイルサイズを用いる代わりに，LZ78 分解や LZ77 分解における分解されたブロック数を用いているのは特徴的である．

Ahonen ら [11] は，2 値 TC ベクトルの系列に対し，NCD を用いて楽曲間の距離を求め，

変奏曲の分類やクラスタリングの問題へ適用している．圧縮プログラムとしては，bzip2, PMMZ を用いている．

4. ポリフォニックな楽曲間の非類似度関数

本稿では，楽曲データから得た 2 値クロマベクトル系列に対して非類似度関数を設計する問題を考える．次の 3 つのアプローチが考えられる．

(1) 各 2 値クロマベクトルを 1 つの文字とみなすアプローチ．
(2) 各 2 値クロマベクトルを実数値 TC ベクトルに変換し，TC ベクトル系列間の非類似度を測るアプローチ．
(3) 各 2 値クロマベクトルから得た TC ベクトルを 2 値化し，得られた 2 値 TC ベクトルを 1 つの文字とみなすアプローチ [11].

(2)(3) は，(1) において，文字 (2 値クロマベクトル) の一致／不一致を，実数値 TC ベクトルあるいは 2 値 TC ベクトルの一致／不一致に置き換えたものとみなすこともできる．いずれにせよ，(1)(2)(3) のすべてに対し，任意の文字列間の非類似度関数が適用可能である．

(1)(2) は文字 (ベクトル) のとりうる値の種類が比較的多いため，比較する文字列間において文字や部分文字列の完全な一致が生じる確率が下がり，その結果，文字や部分文字列の完全一致に基づいた非類似度関数はあまり有効に働かないと予想される．おそらくはこの理由により，Ahonen [11] は，(3) のアプローチによって文字 (ベクトル) のとりうる値を 64 に減らした上で，NCD により非類似度を求めている．しかし，別な方法としては，ベクトル間の非類似度を考慮し，これを文字間の非類似度と考えて，重み付き編集距離や重み付き LCS を用いる方法もあろう．

ベクトル間の類似度としては，ベクトルの成す角の余弦を考え，これを重み付き LCS における重みとする．また，重み付き編集距離においては，この余弦を 1 から引いた値を重みとして用いる．

5. 分 類 実 験

前処理と用いた非類似度関数によって分類精度がどのように変化するかを見るために，ピアノ変奏曲の分類，および JPOP と演歌のジャンル分類をそれぞれ行った．

前処理については，楽曲から得た (1)2 値クロマベクトル列，(2)TC ベクトル列，(3)2 値 TC ベクトル列を用いた．非類似度関数については，以下のものを用いた．

- 編集距離 (EDIT)， 重み付き編集距離 (wEDIT).
- 最長共通部分列長 (LCS)，重み付き LCS (wLCS).
- NCD (gzip, bzip2).
- n グラムカーネル (文字列長 $n = 2, 5, 10$).
- ミスマッチカーネル (文字列長 $n = 5, 10$; ミスマッチの上限 $k = 1, 2$).

分類精度の計算には，5-クロスバリデーションを用いた．1-ノルムソフトマージンのパラメータ ν は 0.05 とした．

5.1 ピアノ変奏曲の分類

1 つの変奏曲の主題と変奏を正例とし，他の変奏曲の主題と変奏を負例として，2 クラス分類を行った．変奏曲データには，モーツァルトのピアノ変奏曲 K.25，K.265，K.354，K.398，K.460，K.501 の MIDI データを使用した．各変奏曲は，1 つの主題と複数の変奏から成っている．そこで，まず，各々の変奏曲から主題と各変奏を切り出し，合計 51 個の MIDI ファイルを得た．次に，これらの MIDI ファイルに対して，16 分音符の長さを単位とし，左手と右手に対応するトラックからそれぞれ 2 値クロマベクトル列を抽出し，これらをマージして 2 値クロマベクトル列を得た．

NCD の計算には圧縮プログラムとして gzip および bzip2 を用いたが，これらは基本的に入力を 1 バイト単位で読み込んで圧縮を行う．(3) の 2 値 TC ベクトル列は各々のベクトルが 6 ビットで表現されるため，これらを 1 バイト整数の系列として書き出したデータファイルにそのまま gzip や bzip2 を適用した．一方，(1) の 2 値クロマベクトル列は各々のベクトルが 12 ビットで表現されるため，1 バイト整数では表現できない．そこで，これらを 4 バイト整数の系列としてファイルに書き出し gzip や bzip2 を適用した．(2) の TC ベクトル列については NCD による非類似度の計算を行っていない．

2 値クロマベクトル列に対する分類精度を表 1 に示す．

また，2 値クロマベクトル及び TC ベクトルに対し，重み付き編集距離 (wEDIT)，重み付き LCS(wLCS) を用いた際の分類精度を表 2 に，2 値 TC ベクトル列に対する分類精度を表 3 にそれぞれ示す．

2 値 TC ベクトル列だとクロマベクトル列・TC ベクトル列に比べて情報が減るため，分類精度も下がると考えていた．しかし分類精度に大きな差は見られない．

全体的に分類精度が高くなる傾向にあるのは，例の数が少ないためだと推測される．上述のように，正例が負例に対して極端に少ない．このような場合は，全て負例と分類する分類器を用いても高い分類精度が得られる．そこで，全て負例として分類した場合の分類精度の平均値を求めたところ，83.33%となった．全て負と分類した場合と比べて多くの場合で高

Vol.2012-MUS-94 No.31
Vol.2012-SLP-90 No.31
2012/2/4

表 1 モーツァルトの変奏曲の分類 (2 値クロマベクトル列)

	EDIT	LCS	NCD(bzip2)	NCD(gzip)
K.25	92.73%	87.27%	94.55%	92.73%
K.265	92.73%	87.27%	83.64%	90.91%
K.354	89.09%	96.36%	94.55%	94.55%
K.398	98.18%	96.36%	90.91%	96.36%
K.460	94.55%	98.18%	100.00%	89.09%
K.501	90.91%	94.55%	96.36%	94.55%
平均	93.03%	**93.33%**	**93.33%**	93.03%

	n グラムカーネル			ミスマッチカーネル			
				$n=5$		$n=10$	
	$n=2$	$n=5$	$n=10$	$k=1$	$k=2$	$k=1$	$k=2$
K.25	90.91%	87.27%	87.27%	89.09%	54.55%	81.82%	58.18%
K.265	80.00%	87.27%	89.09%	81.82%	54.55%	87.27%	65.45%
K.354	96.36%	98.18%	83.64%	89.09%	50.91%	87.27%	76.36%
K.398	89.09%	92.73%	100.00%	98.18%	50.91%	94.55%	74.55%
K.460	94.55%	87.27%	90.91%	83.64%	45.45%	94.55%	87.27%
K.501	81.82%	89.09%	92.73%	56.36%	50.91%	90.91%	70.91%
平均	88.79%	90.30%	90.61%	83.03%	51.21%	89.39%	72.12%

表 2 モーツァルトの変奏曲の分類 (ベクトルのなす角の余弦を利用)

	2 値クロマベクトル列		TC ベクトル列	
	wEDIT	wLCS	wEDIT	wLCS
K.25	92.73%	90.91%	94.55%	85.45%
K.265	94.55%	90.91%	96.36%	92.73%
K.354	94.55%	89.09%	94.55%	100.00%
K.398	29.09%	90.91%	30.91%	100.00%
K.460	92.73%	98.18%	90.91%	98.18%
K.501	90.91%	87.27%	76.36%	96.36%
平均	82.42%	91.21%	80.61%	**95.45%**

い分類精度が得られたことがわかる.

5.2 JPOP と演歌のジャンル分類

JPOP と演歌のジャンルの分類実験では，JPOP と演歌の MIDI データ 119 曲ずつを使用した．MIDI データ中のコード進行の情報から 4 分音符の長さを単位としてコード文字列を抽出した．各々のコードを構成する音高に着目し，2 値クロマベクトル列へ変換した．さらに，これから TC ベクトル列，および，2 値クロマベクトル列を得た.

2 値クロマベクトル列に対する分類精度を表 4 に，2 値クロマベクトル列および TC ベクトル列に対してベクトル間のなす角の余弦を文字の類似度として用いた場合の分類精度を

表 3 モーツァルトの変奏曲の分類 (2 値 TC ベクトル列)

	EDIT	LCS	NCD(bzip2)	NCD(gzip)
K.25	90.91%	92.73%	92.73%	89.09%
K.265	90.91%	85.45%	89.09%	85.45%
K.354	96.36%	100.00%	94.55%	98.18%
K.398	49.09%	94.55%	90.91%	94.55%
K.460	92.73%	92.73%	89.09%	85.45%
K.501	87.27%	90.91%	90.91%	94.55%
平均	84.55%	92.73%	91.21%	91.21%

	n グラムカーネル			ミスマッチカーネル			
				$n=5$		$n=10$	
	$n=2$	$n=5$	$n=10$	$k=1$	$k=2$	$k=1$	$k=2$
K.25	83.64%	85.45%	94.55%	85.45%	90.91%	92.73%	89.09%
K.265	81.82%	89.09%	94.55%	78.18%	76.36%	92.73%	90.91%
K.354	92.73%	87.27%	87.27%	89.09%	90.91%	96.36%	87.27%
K.398	85.45%	90.91%	98.18%	94.55%	85.45%	94.55%	98.18%
K.460	92.73%	85.45%	98.18%	92.73%	98.18%	87.27%	100.00%
K.501	83.64%	87.27%	92.73%	85.45%	92.73%	83.64%	50.91%
平均	86.67%	87.57%	**94.24%**	87.58%	89.09%	91.21%	86.06%

表 5 に，2 値 TC ベクトル列に対する分類精度を表 6 に，それぞれ示す．なお，全て負例として分類した場合の分類精度の平均値は 49.58%であった.

表 4 JPOP と演歌の分類 (2 値クロマベクトル列)

	EDIT	LCS	NCD(bzip2)	NCD(gzip)
分類精度	82.92%	84.58%	82.92%	**88.33%**

	n グラムカーネル			ミスマッチカーネル			
				$n=5$		$n=10$	
	$n=2$	$n=5$	$n=10$	$k=1$	$k=2$	$k=1$	$k=2$
分類精度	82.50%	75.83%	69.17%	82.92%	82.08%	76.25%	75.42%

表 5 JPOP と演歌の分類 (ベクトルのなす角の余弦を利用)

	2 値クロマベクトル列		TC ベクトル列	
	wEDIT	wLCS	wEDIT	wLCS
分類精度	81.25%	82.92%	**83.75%**	82.50%

Vol.2012-MUS-94 No.31
Vol.2012-SLP-90 No.31
2012/2/4

表 6 JPOP と演歌の分類 (2 値 TC ベクトル列)

	EDIT	LCS	NCD(bzip2)	NCD(gzip)				
分類精度	83.33%	79.58%	**85.83%**	85.00%				
	n グラム			ミスマッチ				
				n=5		n=10		
	n=2	n=5	n=10	k=1	k=2	k=1	k=2	
分類精度	84.58%	72.92%	71.25%	78.33%	74.17%	71.67%	78.33%	

6. おわりに

本研究では非類似度に基づいてポリフォニックな楽曲の分類を行った．この分類手法はデータ間の非類似度を計算できれば分類が可能となる．文字列間の非類似度関数をポリフォニックな楽曲に適用する際に，前処理として楽曲データをベクトル列に変換した．今回変奏曲の分類と JPOP と演歌のジャンル分類を行った．その際，楽曲データの前処理と非類似度関数の組み合わせによる分類精度の違いを見た．実験の結果，どの場合でも高い分類精度が得られた．今後の課題として，今回用いた非類似度に基づいた分類手法だけでなく，k-NN 等の他の分類手法でも実験を行う．また楽曲数が少ないため，他のクラシック曲や違うジャンルの楽曲も実験の対象とする．

参考文献

1) Anan, Y., Hatano, K., Bannai, H. and Takeda, M.: Music Genre Classification using Similarity Functions, *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR'11)* (2011).
2) Tzanetakis, G., Essl, G. and Cook, P.: Automatic Musical Genre Classification of Audio Signals, *Proceedings of the 2nd International Symposium on Music Information Retrieval (ISMIR'01)* (2001).
3) Cilibrasi, R., Vitányi, P. and de Wolf, R.: Algorithmic Clustering of Music Based on String Compression, *Computer Music Journal*, Vol.28, No.4, pp.49–67 (2004).
4) Saunders, C., Hardoon, D.R., Shawe-taylor, J. and Widmer, G.: Using string kernels to identify famous performers from their playing style, *Proceedings of the 15th European Conference on Machine Learning (ECML'04)*, pp.384–395 (2004).
5) Lidy, T. and Rauber, A.: Evaluation of Feature Extractors and Psycho-Acoustic Transformations for Music Genre Classification, *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pp.34–41 (2005).
6) Bergstra, J., Casagrande, N., Erhan, D., Eck, D. and Kégl, B.: Aggregate features and AdaBoost for music classification, *Machine Learning*, Vol.65, pp.473–484 (2006).
7) Lidy, T., Rauber, A., Pertusa, A. and Iñesta, J.M.: Improving genre classification by combination of audio and symbolic descriptors using a transcription system, *Proceedings of 8th International Conference on Music Information Retrieval (ISMIR'07)* (2007).
8) Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C. and Scholkopf, B.: Text Classification using String Kernels, *Journal of Machine Learning Research*, Vol.2, pp.563–569 (2002).
9) Leslie, C.S., Eskin, E., Cohen, A., Weston, J. and Noble, W.S.: Mismatch string kernels for discriminative protein classification, *Bioinformatics*, Vol.20, No.4, pp. 467–476 (2004).
10) Harte, C., Sandler, M. and Gasse, M.: Detecting Harmonic Change in Musical Audio, *Proceedings of 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM'06)*, pp.21–26 (2006).
11) Ahonen, T.E., Lemström, K. and Linkola, S.: Compression-based Similarity Measures in Symbolic, Polyphonic Music, *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR'11)* (2011).
12) Wang, L., Sugiyama, M., Yang, C., Hatano, K. and Feng, J.: Theory and Algorithm for Learning with Dissimilarity Functions, *Neural Computation*, Vol.21, pp. 1459–1484 (2009).
13) Mongeau, M. and Sankoff, D.: Comparison of Musical Sequences, *Computers and the Humanities*, Vol.24, pp.161–175 (1990).
14) Kadota, T., Hirao, M., Ishino, A., Takeda, M., Shinohara, A. and Matsuo, F.: Musical Sequence Comparison for Melodic and Rhythmic Similarities, *Proceedings of the 8th International Conference on String Processing and Information Retrieval (SPIRE'01)*, pp.111–122 (2001).
15) Leslie, C.S., Eskin, E. and Noble, W.S.: The spectrum kernel: a string kernel for SVM protein classification, *Proceedings of the Pacific Symposium on Biocomputing (PSB'02)*, pp.566–575 (2002).
16) Takeda, M.: String resemblance system: A unifying framework for string similarity with applications to literature and music, *Proceedings of 12th Annual Symposium on Combinatorial Pattern Matching (CPM'01)*, pp.147–151 (2001).
17) Takeda, M., Fukuda, T., Nanri, I., Yamasaki, M. and Tamari, K.: Discovering instances of poetic allusion from anthologies of classical Japanese poems, *Theoretical Computer Science*, Vol.292, No.2, pp.497–524 (2003).
18) Li, M. and Sleep, R.: Melody classification using a similarity metric based on Kolmogorov complexity, *Sound and Music Computing* (2004).