Kolmogorov Complexity を用いた民族音楽の分類

Clustering of Folkmusic by the Similarity Metric

http://www.tani.cs.chs.nihon-u.ac.jp/g-2005/taako/

谷 研究室　三井 貴子

Takako MITSUI

概要

これまでに音楽を自動的に分類する様々な手法が提案されてきた．Ming Li らは Kolmogorov Complexity に基づいた similarity metric を提案し，Rudi Cilibrasi らは similarity metric を用いて音楽を自動分類する手法を提案した．本研究では Similarity Metric を用いて，民俗音楽を分類する．

1　はじめに

全ての事柄は常に何かに分類されることによって，人々にその存在を認識されている．色や形という外観やその時代背景，目に見える情報やそれらから得られた情報をもとに推測するなど方法は様々である．また，コンピュータ技術の発達に伴い，今日，様々な事柄が電子的に保存されるようになり，文章や音楽も「データ」として扱われるようになってきた．Ming Li，Xin Chen，Bin Ma，Paul M.B. Vitányi らは，Kolmogorov Complexity に基づいてデータ間の距離を表す similarity metric を提案した [1]．これにより「データ」という観点から分類することが可能になり，今まで無関係だとされていた事柄に対して新しい関係の発見が期待される．Rudi Cilibrasi，Paul Vitányi，Ronald de Wolf らの先行研究 [2] では，similarity metric を用いて，Classic や Rock など幅広いジャンルの音楽を取り扱い，それらの情報としての近さをグラフで表した．本研究では，地域の特徴が表れていると予測される民俗音楽 (民謡) を扱うことで，地理的関係の有無やその信頼性を similarity metric を用いて検証する．

第 2 章では，Kolmogorov Complexity について，第 3 章では similarity metric について述べ，第 4 章では実験の概要として前処理の方法や結果の表示方法について解説する．第 5 章では実験の結果について述べる．

2　Kolmogorov Complexity の定義

この章では，Kolomogorov Complexity の基本的な定義 [3] について述べておく．

2.1　Kolmogorov Complexity

ある事柄に含まれている情報について考える．与えられた有限の対象 (たとえば文字列) に含まれる情報量を，その対象を生成するプログラムのサイズ (すなわちビット数) と定義する．ただし，この場合プログラムとは，空の状態からスタートし，すべてを出力するものとする．どんなプログラミング言語を選んでも，それが妥当であれば，情報量は定数分の差しかないことが証明されている．

対象の集合からなる定義域 $D \ni x$ について，s をプログラム言語，$|p|$ をプログラムのサイズ，$S(p)$ を $n = S(p)$ となる関数，$n(x)$ を列挙する方法を仮定したときの x の順位とすると，記述量は以下のように定義される．

$$K_s(x) = min\{|p| : S(p) = n(x)\}$$

ただし，x を生成する p が存在しないときは，

$$K_s(x) = \infty$$

となる．

また，ある対象 y が与えられているときの対象 x の記述量を定義する．文字列 x を，計算可能関数 f，それに文字列 p と y により $f(p, y) = x$ と記述することを考える．そこで，任意の計算可能関数 f をインタプリンタ関数あるいは複合関数と，任意の $p \in \{0, 1\}^*$ をプログラムと呼ぶことにする．インタプリンタ関数 f のもとで，補助情報 y に対する x の記述量 K_f を以下のように定義する．

$$K_f(x|y) = min\{|p| : p \in \{0, 1\}^* \& f(p, y) = x\}$$

ただし，条件を満たす p がないときは，

$$K_f(x) = \infty$$

となる．

また，x と y を区別できる形で出力させる最短のプログラム長を，$K(xy)$ と表す．$O(\log K(xy))$ の誤差範囲では，

$$K(xy) = K(x) + K(y|x)$$

となることが証明されている．

2.2 K のアルゴリズム的性質 [3]

$K(x)$ を正の整数から正の整数への関数と考える.

定理

(a) 関数 $K(x)$ は帰納的ではない. しかも, どのような帰納的部分関数 $\phi(x)$ を考えても, もしその値が無限個の点で定義されているならば, その定義上のどこかの点で $\phi(x) \neq K(x)$ となる.

(b) 引数 t に対して単調減少で (全域的な) 帰納的関数 $H(t,x)$ が存在し, $\lim_{t\to\infty} H(t,x) = K(x)$. すなわち, $K(x)$ のよい近似は計算可能. ただしこれは一様近似ではない.

2.3 情報量

相対記述量 $K(x|y)$ が絶対記述量 $K(x)$ よりかなり小さい場合,「y が x についての情報を多分に含んでいる」と考えることができる. したがって, 定数分の差異を無視すれば,“ y に含まれる x の情報量 ”は

$$I(x:y) = K(y) - K(y|x)$$

とみなすことができる. また, $K(x|x)$ となる x を選べば,

$$I(x:x) = K(x)$$

となる. ここで考えるべきことは, 情報の対称性である. $O\log K(xy)$ の範囲では,

$$K(xy) = K(x) + K(y|x)$$

が成り立つので, これを用いると,

$$I(x:y) = I(y:x)$$

が成り立つ.

3 similarity metric

数学において距離空間とは, 任意の 2 点間で距離が定められた空間のことをいう.

定義

ある集合 X 上の距離とは, 実数値関数 $d : X \times X \to R$ で任意の $x, y, z \in X$ に対して次のような性質を満たす.

$$d(x,y) \geq 0$$
$$d(x,y) = 0 \Leftrightarrow x = y$$
$$d(x,y) = d(y,x)$$
$$d(x,y) \leq d(x,z) + d(z,y) : \text{三角不等式}$$

これをもとに, 情報距離について考える.

Ming Li らの研究では, 情報に関する距離を標準化している. 任意の文字列 x, y について, 以下のように決める.

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

また, $K(y) \geq K(x)$ としたとき,

$$d(x,y) \quad = \quad \frac{K(y|x)}{K(y)}$$

これに対して先に述べた情報量の公式を用いると,

$$d(x,y) \quad = \quad \frac{K(y) - I(x:y)}{K(y)}$$

となる. また 2 節で示した通り, $O(\log K(xy))$ の範囲では $K(xy) = K(x) + K(y|x)$ が成り立つので,

$$d(x,y) = \frac{K(xy) - K(x)}{K(y)}$$

と表すことができる.

実際の実験では, この式とともに以下の 3 つの理想理論のもとに行われた.

(1) 要求された情報距離 $d(x,y)$ は漠然と長い文字列 x, y によって得られる.

(2) Kolmogorov Complexity は帰納的でないため, 計算不可能である.

(3) 実用的な方法で情報距離を近似する際, 圧縮方法の一つである“ bzip2 ”を用いる.

以上より、$bzip(x)$ を文字列 x を bzip2 で圧縮したときのファイルサイズとすると情報距離 $d(x,y)$ は以下のように近似される.

$$d(x,y) \quad \frac{bzip(xy) - bzip(x)}{bzip(y)}$$

この距離関数をもとに, データの分類を進めていく.

4 実験概要

4.1 前処理

今回の実験で取り扱ったデータは MIDI と呼ばれる音楽ファイルである. MIDI とは Musical Instrument Digital Interface の略で, 楽器間のデータ通信方法の規格を指す [4]. 音をデジタル化することによって, 音色というものは音程 (ピッチ)・波形・音量という 3 つのデータとして記憶される. この記憶されたデータを MIDI メッセージと呼び, 今後は１６進数で表現する. メッセージ

の内容によっては複数のバイトにまたがるため，最上位ビット (Most Significant Bit，以後 MSB) がフラグのような機能を備えている．MSB が 1 の場合，そのバイトをステータス・バイト，0 の場合データ・バイトと呼ぶ．ステータス・バイトにはメッセージの種類が示されており，データ・バイトにはデータの情報が書かれている．以下に MIDI メッセージの中から一例を紹介しておく．

ノート・オン／ノート・オフ　ノート・オン／ノート・オフとは音を出す MIDI メッセージである．これらは必ずペアで存在し，共に 3 バイト必要である．以下にメッセージ内容を示す．

ノート・オン

9n	kk	vv
メッセージ種類	ノート番号	ベロシティ

ノート・オフ

9n(8n)	kk	00
メッセージ種類	ノート番号	ベロシティ

n:MIDI チャンネル　楽器の設定を行う．ノート・オン／ノート・オフにおける，ステータス・バイトの下位 4 ビット (0〜F) で表される．

kk:ノート番号　音程を表している．データ・バイトにあたるため，MSB は 0 で設定されている．よって，00〜7F までで表される．ノート番号 60 とはドの音 ($261.6Hz$) に当たる．

vv:ベロシティ　音の強弱を伝えるパラメータのこと．ノート番号と同様にデータ・バイトあるため，00〜7F で表される．00 とは，無音つまりその音を止めることである．ノート・オフメッセージには 9n〜と 8n〜の 2 種類があるが，一般的に 9n〜が使われている．

実際の MIDI ファイルは，この演奏データに時間情報を加えた SMF(Standard MIDI File) と呼ばれる標準ファイルフォーマットとなっている．１４バイトで表現された「ヘッドチャンク」から始まり,「トラックチャンク」と呼ばれるトラックがいくつか連なって一つのファイルを構成している．それぞれのファイルで単位時間が定義されており，SMF はその時間を基準に MIDI メッセージを送信する順番を記憶しているだけである．つまり，プレイヤーでは「待つ」→「送信」が繰り返し行われているのである．

実際の Rudi らの実験では，MIDI チャンネルをピアノに設定し，MIDI データを 0.05 秒ごとに細分化する．その後それぞれの平均ベロシティを計算し昇順にソートし，もう一度一つのファイルにまとめる，という操作を行っている．今回の実験では，MIDI チャンネルをピアノに設定し，トラックチャンク内の音楽データにおけるノートオン／ノートオフ以外の MIDI メッセージをすべて取り除いて行っている．

4.2 quartet method

次に，結果の表示方法について述べる．この実験では，“ quartet method ”という系統樹の一つを用いている．“ quartet ”とは，２つの葉をもつ２つの subtree が連結したグラフを指す (Figure1)．ある n 個のデータ集合を S としたとき，$S \ni u, v, w, x$ に対して quartet は $uv|wx$ と表現する．“ | ”は２つの subtree に分解されることを表している．

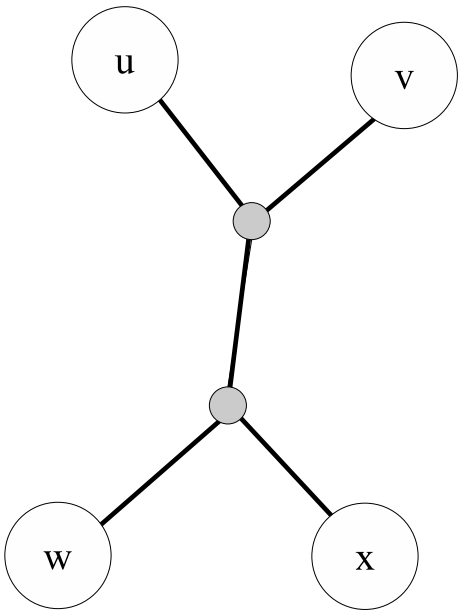

Figure1:quartet

quartet に対する cost を次のように定義する．

$$C_{uv|wx} = d(u, v) + d(w, x)$$

また，ある木 T が与えられたとき，u から v までの辺と w から x までの辺が交わらないような $uv|wx$ のことを“ *consistent* ”と呼ぶ (Figure2)．

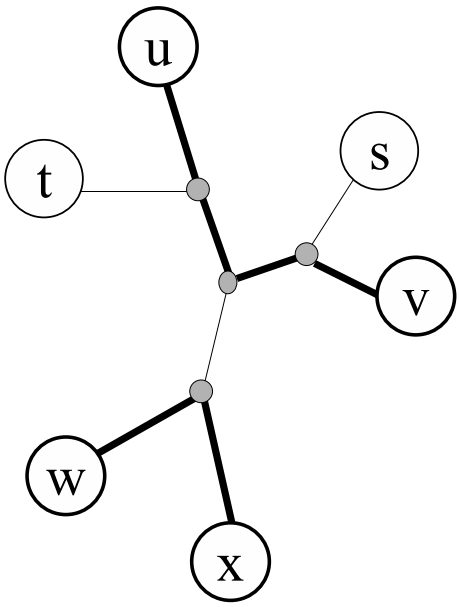

Figure2:consistent な組み合わせ

T から４つの葉を選べば３つの quartet となる組合せが考えられ，そのうちの１つが必ず *consistent* となる．この *consistent* となるすべての quartet の cost の和を，その木の total cost とする．これを計算するには n^4 かかるため，データ数が多くなるほど時間がかかってしまうことが難点である．この total cost が小さいほど，良い木である．そこで，その木に対する best(minimal)cost と worst(maxmal)cost を計算し，比較することにする．木の best(minimal)cost とは，それぞれの quartet に対する best(minimal)cost の和とし，同様に quartet の worst(maxmal)cost の和をその木の worst(maxmal)cost とする．ここで大切なことは，その組合せで木が作れなくてもよいということだ．total cost は必ず，この２つの値の間に存在することになる．ここで，もっとも悪い木のとき，つまり worst cost のとき $S(T) = 0$，もっとも良い木，つまり best cost のとき $S(T) = 1$ となる評価関数を $S(T)$ とする．問題の目標としては，$S(T) = 1$ となる木を得ることである．しかし，この問題は NP-困難であることが知られている．

そこで，先行研究に基づいて本実験は以下のようなヒューリスティックを用いて，$S(T)$ の値をある一定値以上となるような木を得ることにする．

(1) n 個の leaf，$n-2$ 個の internal node(次数:3) をもつ木をランダムに生成

(2) $S(T)$ の値を計算

(3) 以下の３つのうち一つをランダムに選び，操作を行う (tranceform)

(a) $leafswap$
２つの leaf をランダムに選び，交換する

(b) $subtreeswap$
２つの internal node をランダムに選び，subtree ごと交換する

(c) $subtreetransfer$
ランダムに選んだ leaf，または internal node を切り離し，他の場所へ移動させる

(4) (2)，(3) を繰り返し，$S(T)$ の値が大きくなるように T を更新

これらの操作をランダムに繰り返しているため，$S(T)$ の値はプログラムを実行するたびに変化する．tranceform の回数や $S(T)$ の値を基準とし，複数回実験を行うことにする．

5 実験結果

実験の方法として，tranceform の回数を固定する方法と，$S(T)$ の値がある一定以上になるまで繰り返す方法の二通りで行う．また，参考として前処理をしない場合も行った．

5.1 実験１

まず，今回の実験では先行研究とは前処理方法が違うため，同じデータを用いて比較実験を行う．Rudi らの研究では，$S(T) = 0.958$ であったので (Figure3)，tranceform の回数を 100,000 回，$S(T) \geq 0.9$ にそれぞれ固定して行った．複数回実験を行うと，似たような結果が得られた (Figure4)．これは tranceform を 100,000 回，$S(T) = 0.95$ の図である．

扱ったデータは，J.S.Bach の平均律クラーヴィア 2 のプレリュード 1，2 とフーガ 1，2，Chopin のプレリュード作品 28 のうち 1，15，22，24，Debussy のベルガマスク組曲より 4 曲すべてである．それぞれ作曲家ごとにまとまっていることに注目されたい．

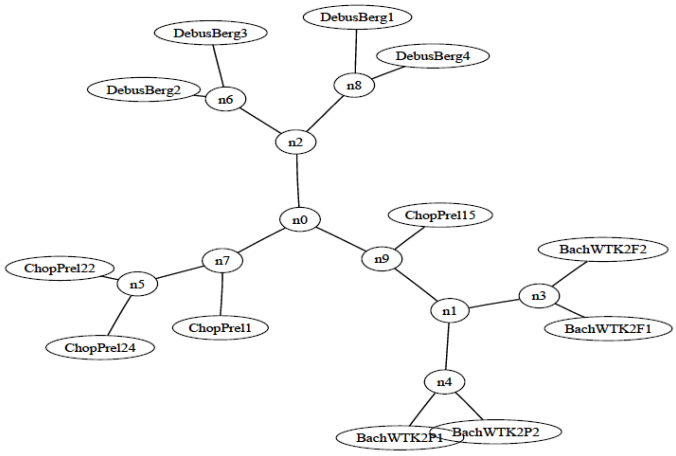

Figure3:Rudi らの実験結果

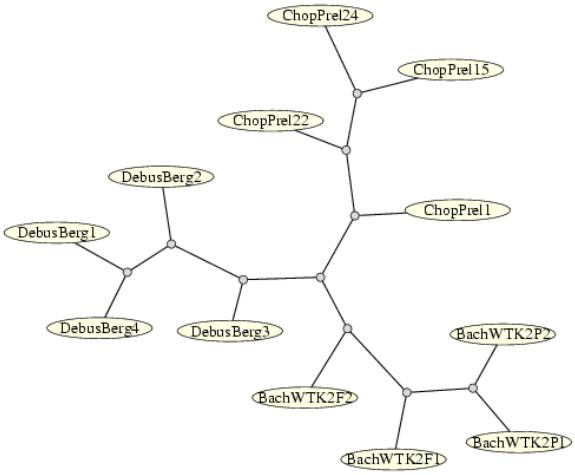

Figure4:tranceform:100,000 $S(T) = 0.9488$

これにより，あまり複雑な音楽データでなければ簡単な前処理をするだけでもよい結果が得られると考えられる．

5.2 実験２

世界各地の民族音楽データを各国 1 曲ずつ 27 個集め [6]，同様の実験を行う．データ数が多いと $S(T)$ の値は上がり難くなってしまう．よって，今回では tranceform を 50,000 回，$S(T) \geq 0.75$ とする．

5.3 実験３

日本各地の民謡を集め [7]，同様の実験を行う．地方民謡や県民謡などを含む 43 曲で行う．
tranceform を 1,000 回，$S(T) \geq 0.7$ とする．

現段階では例として tranceform を 1,000 回行った場合を最後に載せておく (Figure5)．

参考文献

[1] Ming Li, Xin Chen, Bin Ma and Paul M.B. Vitányi. The Similarity Metric.In Proceedings of the 14th annual ACM-SIAM symposium on Discreat algorithms(SODA),pp.863-872,Baltimore,Maryland,2003.

[2] Rudi Cilibrasi, Paul Vitányi and Ronald de Wolf. Algorithmic Clusteringof Music(2003)

[3] Ming Li and Paul M.B. Vitányi. 渡辺 治 翻訳 Kolmogorov Complexity and its Applications. コンピュータ基礎理論ハンドブック (1994)

[4] リットーミュージック出版編集部【編】 MIDI バイブル I MIDI1.0 規格 基礎編 (1997)

[5] 浅野哲夫小保方幸次 共著. LEDA で始める C/C++ プログラミング

[6] http://www.momo-mid.com/

[7] http://www5b.biglobe.ne.jp/pst/nihon-minyou/

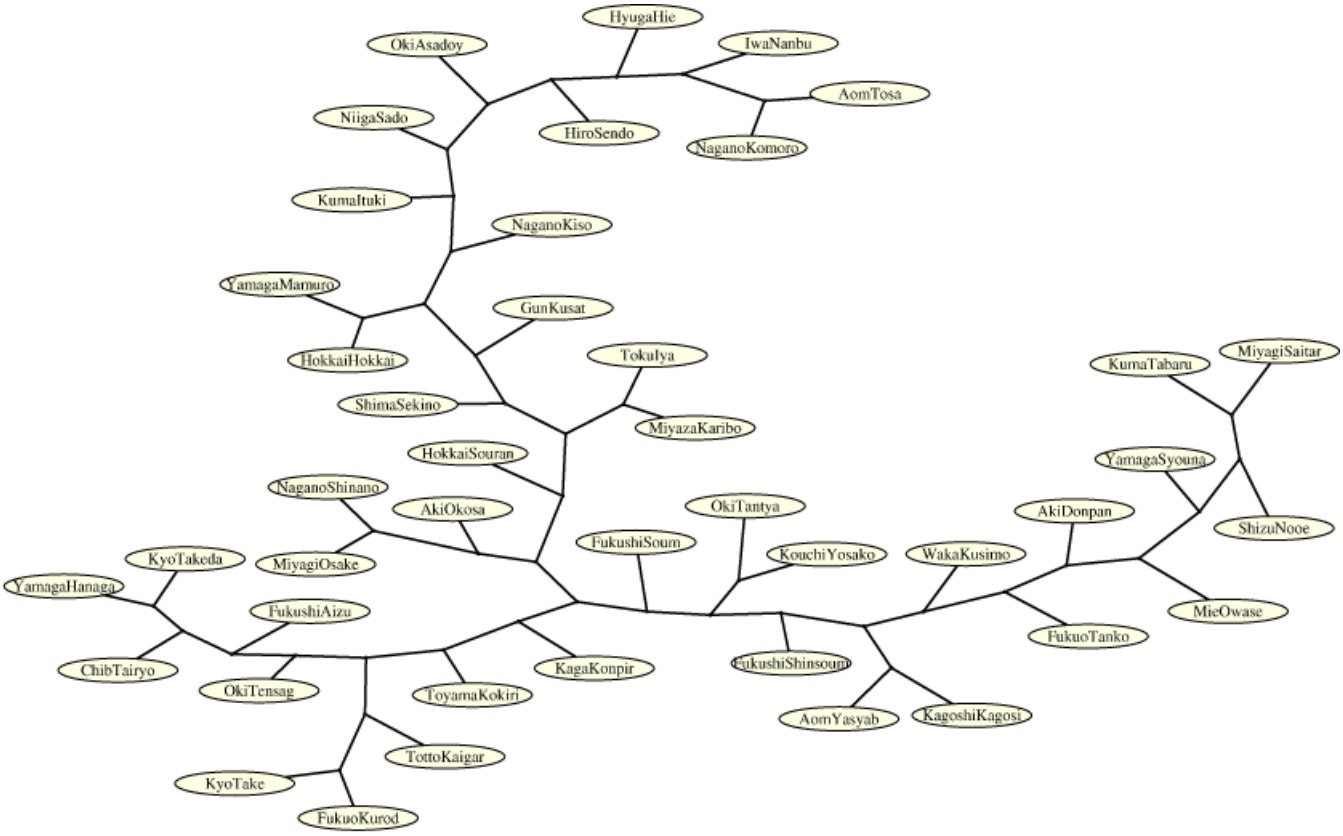

Figure5:tranceform:1,000 $S(T) = 0.4491$