Kolmogorov 記述量に基づく方言の自動分類

Automatic classification of Japanese dialects using similarity metric is based on Kolmogorov Complexity

http://www.tani.cs.chs.nihon-u.ac.jp/g-2007/yasuto

谷 研究室　関根 康人

Yasuto Sekine

概要

Kolmogorov 記述量に基づいたデータ間の類似度に関する距離が定義され，ＤＮＡの類似度や言語の類似度，音楽の類似度判定に有用だという実験結果が得られている．本研究では Kolmogorov 記述量が方言の類似度分析に対して有用かどうか実験を行う．

1　はじめに

常に全ての事柄は何かによって分類されている. 今日，コンピュータ技術の発達に伴い，文章や音楽も「データ」として扱われるようになってきた．Ray J.Solomonoff, Anderei Kolmogorov, Gregory J.Chaitin らは，Kolmogorov 記述量に基づくデータ間の類似度に関する距離を表す similarity metric を発案した．これを用いて，ＤＮＡの類似度や言語の類似度，音楽の類似度判定に有用だという実験結果が得られている．

これまで方言は，単語単位では類似度解析がされてきたが，文章を対象とした類似度解析はされたことがなかった．そこで本研究では,「方言ももたろう」（監修・著：杉藤美代子）というソフトのデータを対象として類似度解析を行っていく．

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 章は本研究において用いた系統樹作成法のでありますquartet method について説明をし，第 5 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，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 [?] 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．

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

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

$K(x)$ 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 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.

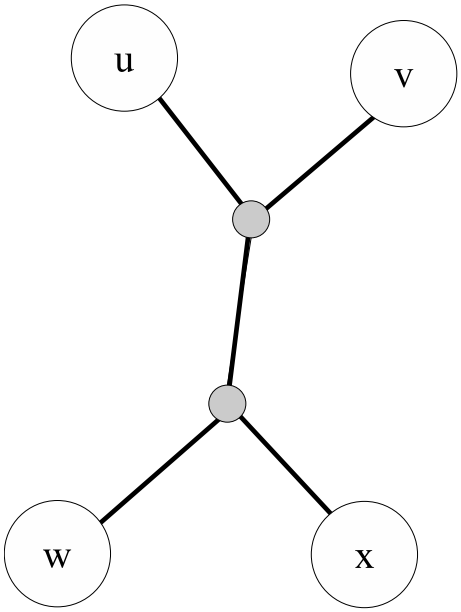

Figure1:quartet

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).

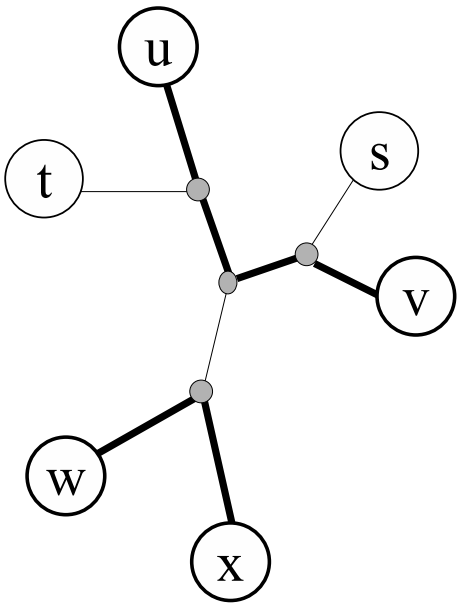

Figure2:consistent な組み合わせ

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 実験概要

実験の方法として，$S(T)$ の値がある一定以上になるまで繰り返す方法で行う．

実験データ

方言ももたろう（監修・著：杉藤美代子）というソフトに昔話「桃太郎」が日本各地５６箇所の方言で読まれた音声データがある。その音声データをテキスト化（ひらがな・。のみ使用）また「を」→「お」、「へ」→「え」、「は」→「わ」、伸ばす音は前の音の母音をもう一つ加えるという文章の書き方をした。

文字コード

一般に使われている EUC、JIS、SHIFT-JIS、UTF-8 を使用する。なお改行コードは、全て unix に固定させた。

前処理 1

50 音順で 1 から対応させて変換する。

前処理 2

使用頻度が高い順に 1 から対応させて変換する。

S(T) 値

quartet method における木の評価値である S(T) 値を用いた nj 法、upgma 法、で作成される木の評価。

6　実験結果

色分け

日本の 56 箇所のデータを 9 地方に分けて色分けをする。

北海道地方 → 赤
東北地方 → 緑
関東地方 → 青
中部地方 → 黄色
近畿地方 → ピンク
四国地方 → 紫
中国 → オレンジ
九州地方 → 水色
沖縄 → 白

結果

quartet-remake-preprocess2

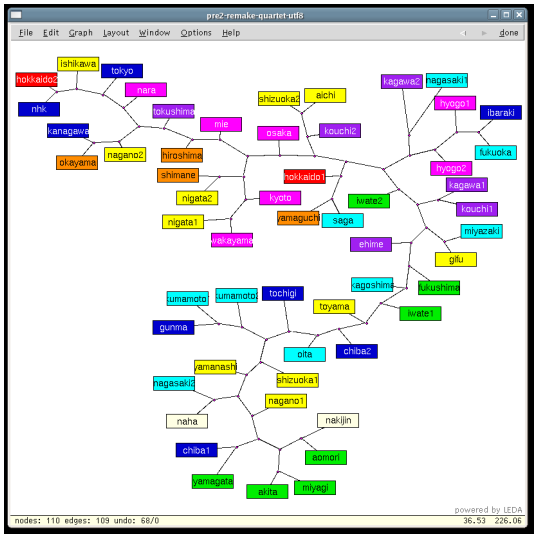

Figure1: $S(T) = 0.850286$
nj-remake-preprocess2

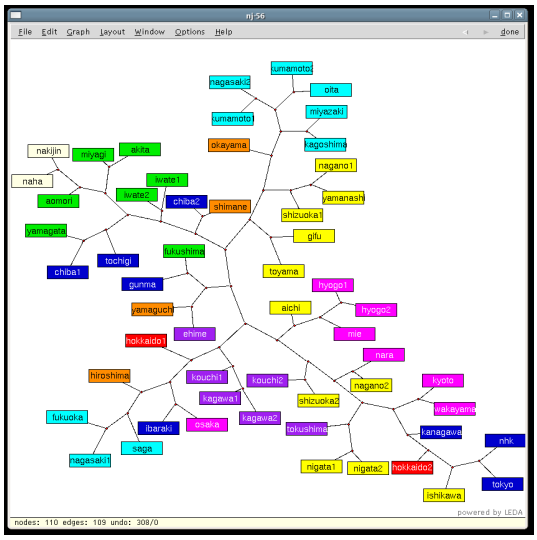

Figure2: $S(T) = 0.455565$
upgma-remake-preprocess2

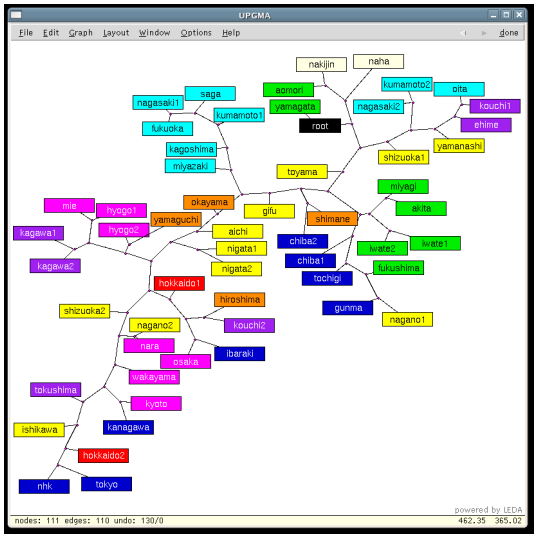

Figure3: $S(T) = 0.455565$

参考文献

[1] Rudi Cilibrasi and Paul M.B. Vitányi. A New Quartet Tree Heuristic for Hierarchical Clustering.

[2] 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.

[3] http://www.tani.cs.chs.nihon-u.ac.jp/taako/

[4] 徳川宗賢W. A. グロータース編. 方言地理学図集.