圧縮による類似度を用いた方言の自動分類

Automatic classification of Japanese dialects using based Compression similarity distance

谷　研究室　　齋藤　岳丸
Takemaru Saito

概要

圧縮による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。昨年、荒堀、福田氏がが Kolmogorv 記述量を用いた方言の類似度分析に対して、音声入力データのの自動テキスト化や木の評価方法、圧縮方法の違いによる研究を行ったが、本研究では音声入力データのテキスト化にピッチデータを含むことについての追加研究を行う。

1　はじめに

現在様々な事柄がデータ化されている。もちろん、音楽や文章も「データ」として扱われるようになってきている。近年 Ming Li らが 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（現在 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ことが分かっている。）した。

昨年、新堀、福田氏が Kolmogorv 記述量を用いた方言の類似度分析に対して、音声入力データをドラゴンスピーチという音声認識ソフトで音声データをテキスト化したものを用いて実験を行った。本研究では音声入力データを音声録聞見というソフトウェアによりピッチ情報を抽出し、その情報をテキストに付加したデータを用い、去年のデータとの比較を行う追加研究とする。

2　Kolmogorov 記述量の定義

Kolmogorov 記述量の基本的な定義を述べておく。

2.1　Kolmogorov 記述量

あるデータ x が存在し、データ x の Kolmogorov 記述量とはあるプログラム言語で x を生成する最小のプログラムのサイズである。どんなプログラム言語を選んでも、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。S をプログラム言語、$|p|$ をプログラムサイズとすると、データ x の Kolmogorov 記述量 $K_s(x)$ は、以下のように定義される。

$$K_s(x) = min|p| : S(p) = x$$

すなわち $K_s(x)$ は、”プログラム言語 S において x を生成する最小のプログラムの長さ ”と考えていいだろう。

次に対象 y が与えられているときの対象 x についての記述量（相対記述量）について考える。文字列 x を、計算可能関数（インタプリタ関数）f、それに文字列 p と y により $f(p, x) = x$ と記述することを考える。インタプリタ関数 f のもとで、補助関数 y に対する x の記述量 K_f を次のように定義する。

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

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の長さ $K(xy)$ とあらわす。$O(logK(xy))$ 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、どのような機能的部分関数 $\phi(x)$ を考えても、もしその値が無限個の点で定義されているのならば、その定義域上のどこかの点で $\phi(x) \neq K(x)$ となる。

(b) 引数 t に対しては単調減少で（全域的な）機能的関数 $H(t, x)$ が存在し,$\lim_{t\to\infty} H(t, x) = K(x)$。すなわち、$K(x)$ の良い近似（ただし一様近似ではない）は計算可能。

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 についての情報をたぶんに含んでいる」と考えることが出来る。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

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

とみなすことが出来る。また、$K(x|x)$ となる x を選べば、

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

となる。ここで情報の対象性ということを考える。$OlogK(xy)$ の誤差範囲では、

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

である。したがって

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

が成立する。

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 音声のピッチ情報

人間の発音する音声は声帯が振動することにより発生する音が舌や骨格などで○○することで作り出されるが、その音声は様々な周波数成分を含む。特に母音を発音する時に声帯が振動し、この周波数をピッチ周波数という。方言によりこのピッチ周波数の分布に特徴がでる。

5 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．

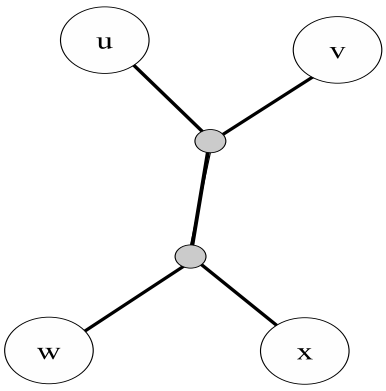

Figure1:quartet

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).

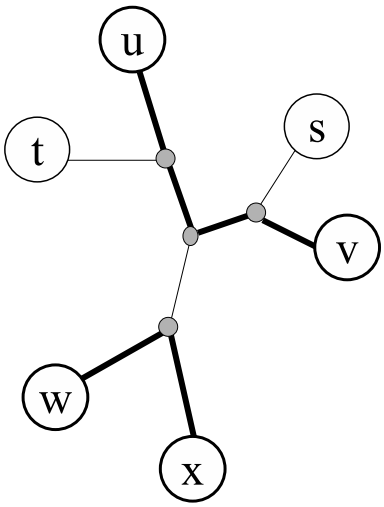

Figure2:consistent な組み合わせ

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 minimum cost と maximum cost を計算し，比較することにする．木の minimum cost とは，それぞれの quartet に対する minimum cost の和とし，同様に quartet の maximum cost の和をその木の maximum 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 maximum cost のとき S(T)=0，もっとも良い木，つまり minimum 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 inner 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 inner 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 inner 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．

6 木の評価

クラスタリングをして出来た木が similarity metric で求めた類似度の距離に沿って出来ているかを検証するための評価値を定義する。
similarity metric で求めた類似度の距離を $sd(u,v)$ とし、最小値を $min(sd)$、最大値を $max(sd)$ とする。またグラフとしてみた場合の頂点同士の距離（2 頂点間の辺数）を $td(u,v)$ とし、$td(u,v)$ のうちの最小値を $min(td)$、最大値を $max(td)$ とする。

$td(u,v)$ を類似度の距離に対応させた距離を $ntd(u,v)$ と定義する。そして変換定数を S とする。

$$S = \frac{max(sd) - min(sd)}{max(td) - min(td)}$$

$$ntd(u,v) = min(sd) + S(td(u,v) - min(td))$$

これらから木の評価値を TV_1、TV_2 と 2 種類定義する。L は末端ノードである。

$$TV_1 = \sqrt{\sum_{u,v \in L} (ntd(u,v) - st(u,v))^2}$$

$$TV_2 = \sqrt{\frac{\sum_{u,v \in L} (ntd(u,v) - st(u,v))^2}{\sum_{u,v \in L} ntd(u,v)^2}}$$

7　実験概要・実験データ

実験はこれからでございます。

8　実験結果

実験はこれからでございます。

9　今後の課題

そりゃいろいろありまんがな。

10　参考文献

(1)　Rudi Cilibrasi and Paul Vitanyi and Ronald de Wolf　Algorithmic Clustering of Music (2003)

(2)　Ming Li and Paul M.B.Vitanyi. 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)

(3)　谷 聖一　データの複雑さ・データ表現の複雑さ

(4)　堀中 幸司　Kolmogorov 記述量に基づく類似度を用いた方言の自動分類　(2007)