圧縮度にもとづいた汎用な類似度測定法

Paul Vitányi・（翻訳）渡辺 治

1. はじめに

文字列で表わされる物は少なくない[*1)]．たとえば，ネズミのゲノム情報を表わす A, C, G, T から成る文字列であるとか，トルストイの「戦争と平和」の本文となる文字列，等々，我々の周囲には文字列で表わされる物が様々にある．このように，文字列として与えられるデータ群に対し，それらの間の類似性を決定し，同類のもの同士をグループ化する問題を考える．なお，文字列は，ときには記号列といった方が適切な場合もあるが，本稿では「文字列」という呼び名を用いることにする．

ある種のデータ群が与えられたとする．たとえば，様々な音楽のファイル，といったデータ群である．あるいは，異なる ATM 端末の通信記録，様々な人のクレジットカードの申込書，各種遺伝子データ，等々，である．これらのデータ群の各データ間には，隠された関係があるはずで，我々はその関係を明らかにしたい．たとえば，遺伝子データからは，文字あるいは文字列の並びの頻度を見出せるかもしれない．音楽データからは，調子やリズム，あるいはハーモニー，など，様々な数値的な特徴を見出せるだろう．

我々の研究グループは，最近，圧縮度にもとづいて類似度を測定する非常に一般的な手法を提案した．つまり，類似度測定の「汎用な」手法である．これは，あらかじめ想定した特徴や事前知識を一切使わない方法である．別の言い方をすれば，この新しい方法はノンパラメトリックな統計手法のように，本質的に必要と思われる特徴を制限なく導入できる方法ともいえる．したがって，この類似度は非常に一般的で，音楽，文章，文学作品，プログラム，遺伝子情報，バイナリ，自然言語，等々，あらゆる分野のデータに対して，まったく同じに，必要であれば同時に適応できるのである．

2 つのデータに対し，もし，その 1 つを他の情報を元に十分なだけ「圧縮」できる場合，大ざっぱにいえば，その 2 つのデータは十分近いと考えてよい．これが我々の方法の基本的な考え方である．2 つの類似度が高ければ高いほど，1 つを知れば他を簡潔に表わすことができる，という考え方である．我々の方法は，文字列として表現されているデータであれば，どのような対象にも適用できる．これが，本稿で紹介する類似度測定法である．我々はまた，類似性を視覚的に見せるためのクラスタリング法も開発した．原理的には通常の進化系統樹を作る方法だが，従来法よりははるかに高速なヒューリスティックスを開発したのである．以上の手法は，ソフトウェアパッケージ CompLearn として公開されている[2)]．

*1) その他の場合として，対象物が，たとえば，「トルストイの戦争と平和」というように，その名前で表わされる場合がある．また，「家庭」とか「赤」のように，具体的な文字列が対応するわけではなく，名前と，その背後にある常識や知識によって表現されるものもある．

2. 類似距離と正規圧縮距離

与えられた 2 つの対象物に対し，両者の類似性を評価したい．そのために，我々は「類似距離」を導入する．正確には，非類似度を測るための距離なので「非類似距離」と呼ぶべきかもしれない．

具体的には，文字列として表現される 2 つの対象 x, y に対し，非類似度を表わす非負実数値を与える距離関数 $D(x, y)$ を定義する．ただし，通常の距離と同様，以下のような距離の公理を満たすものを考える：

- $D(x, y) = 0 \iff x = y$,
- $D(x, y) = D(y, x)$ （対称性）,
- $D(x, z) \leq D(x, y) + D(y, z)$ （三角不等式）.

たとえば，2 つのクラッシックの楽譜が対象 a, b の場合，$a = b$ ならば $D(a, b) = 0$，a と b が同一の作曲家によるものであれば $D(a, b) = 1$，そうでなければ $D(a, b) = 2$ と定義する．そうすれば，距離の公理を満たしている．これは楽譜の，ある 1 つの特徴，たぶん重要な特徴をとらえた距離である．

けれども，特徴は，これだけではない．一般のデータを考えた場合には，無限種類の特徴があり，その各々が独自の類似度を定義するだろう．データ解析における主要な問題は，問題領域に応じて，対象とするデータに対して適切な特徴を選ぶことである．

コルモゴロフ記述量の理論 (Kolmogorov complexity theory) に基づく理論的な考察から，万能な（汎用な）特徴，あるいは，万能類似度という概念を導くことができる．与えられた文字列 x に対し，コルモゴロフ記述量 $K(x)$ とは，大ざっぱにいえば，x を復号可能な範囲内で究極に圧縮したときに得られる二進列のビット長である．我々も，通常，`gzip` や `bzip2` など，データ圧縮の技術をよく利用する．たとえば，与えられた文字列（から成るファイル）x に対し（もし x に冗長性があれば），`gzip` は，それを，より短い文字列 x' に圧縮する．しかも，x' から x を復号するプログラムもある．このようなデータ圧縮プログラムの中で，究極のもの $\mathtt{c}_{\mathrm{opt}}$ があり，すべての圧縮プログラムよりも高い圧縮率を達成できたとしよう．ただし，圧縮された列を復元できなければならないので，この圧縮プログラム用の，とても強力な復号プログラムの存在も仮定する．直感的には，$K(x)$ は $\mathtt{c}_{\mathrm{opt}}$ により x を圧縮した結果の長さである．

さらに一般的には，y に対する x の相対コルモゴロフ記述量（条件付きコルモゴロフ記述量ともいう）を $K(x|y)$ と表わす．これは，補助データ y を用いて復元することを想定したときの，x の究極の圧縮列長である．

一般に，y に含まれる情報を用いてもよい場合には，x をさらに圧縮できる可能性がある．そのため，$K(x|y)$ は $K(x)$ よりはるかに小さくなる可能性がある．この差は，x を何も予備知識を仮定しないで圧縮する場合に対し，y を仮定して圧縮する場合の「圧縮効率の改善度」である．言い換えれば，この差 $K(x) - K(x|y)$ は，y 中に含まれる x の「情報量」と見ることができる．実際，この解釈については，理論的な妥当性も示されている．

さらに詳しく述べると，$K(x) - K(x|y)$ は，y 中における x の最大情報量であることが示せる．また，小さな誤差を無視すれば，$K(x) - K(x|y)$ と $K(y) - K(y|x)$ が，ほぼ等しいことも証明することができる．つまり，両者は x と y に共通する情報量の最大値であり，言い換えれば，両者の「類似部分」の量と見ることができる．この量が「圧縮に基づく万能類似度」の基本である．

ただし，実際的な類似距離の定義のためには，まだ，いくつか考えなければならない点がある．それらを以下で考えてみよう．

妥当な類似距離の定義に向けて

最初に考えなければならない点は，コルモゴロフ記述量の非計算可能性である．与えられた x に対し，$K(x)$ を計算することは一般には不可能である．非常に一般的で強力な復号プログラム $\mathtt{d}_{\mathrm{opt}}$ は構成可能である．しかし，その復号プログラム

$\mathtt{d}_{\rm opt}$ に対して，$\mathtt{d}_{\rm opt}(x') = x$ となるような最短の列，すなわち x の最短記述 x'（あるいは，その長さ）を計算することができないのである．

その対処法として，単純に，コルモゴロフ記述量を定義する究極の圧縮プログラムを，現実に使っている圧縮プログラムで近似しよう，というのが我々の提案である．たとえば，現実に使われていて，しかも性能の良い圧縮プログラムを c とする．そのプログラムにより得られる x の圧縮列のビット長を $C(x)$ とする．その $C(x)$ を $K(x)$ の近似として使おう，という考えである．

一般の圧縮プログラムには，相対コルモゴロフ記述量を定義するときに使った「補助情報を用いた圧縮」という概念がない．そこで，ここでは，相対記述量 $C(x|y)$ を，

$$C(x|y) \ = \ C(yx) - C(y)$$

と定義する．直感的には，yx を記述するには，y を記述し，それを用いて x を記述するのが順当な方法である．この前者の記述量が $C(y)$ であり，後者が $C(x|y)$ である．したがって，$C(yx) = C(y) + C(x|y)$ が成り立っていると考えてもよいだろう．上記の定義は，この式から導かれたものである．理論的にも，コルモゴロフ記述量においては，この関係式が小さい誤差を無視すると成り立つことが証明されている．

この $C(x)$ と $C(x|y)$ をコルモゴロフ記述量の代わりに用いると，$C(x) - C(x|y) = C(x) + C(y) - C(yx)$ が得られる．これを x と y が共有する情報量，言い換えれば，x と y の類似度の近似値として用いるのである．

二番目の点は正規化である．食い違いの大きさは，対象の大きさ（つまり，文字列の長さ）との比率で考えるべきである．たとえば，18,000 塩基から成るミトコンドリアの遺伝子において，9,000 箇所の違いは大きな違いである．けれども，3×10^9 塩基からなる核酸全体において，9,000 箇所の違いは微々たるものだろう．したがって，絶対的な差ではなく，正規化された差を考えるべきである．この点をもう少し慎重に考えると，文字列 x, y 自身の長さではなく，その記述量 $C(x), C(y)$ によって正規化した方がよいことがわかる．

以上の考察から，正規化された類似度を（近似的に）求める式として，次の式が導き出せる．

$$\frac{C(x) + C(y) - C(yx)}{C(x)} \qquad (1)$$

この式 (1) の値について，少し考えてみよう．一般に，$C(yx) \geq C(y)$ なので式の値は 1 以下だが，x と y がほとんど同じで，$C(yx) \approx C(y)$ である場合に，ほぼ 1 となる．一方，$C(yx) \geq C(x) + C(y)$ なので式の値は 0 以上だが，x と y が，まったく異なり，そのため $C(yx) \approx C(x) + C(y)$ の場合には，式 (1) ≈ 0 である．つまり，式 (1) は，0 から 1 までの範囲で x と y の類似度を表わしているのである．

我々が欲しいのは，非類似度を表わす距離であった．また，妥当な距離であるためには対称性も必要である．以上を考慮して最終的に到達したのが，以下に示す 正規圧縮距離 (normalized compression distance)，略して NCD である．

$$\begin{aligned} &\mathrm{NCD}(x,y) \\ &\quad = \ \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}} \\ &\quad = \ \frac{C(yx) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \end{aligned}$$

厳密には，最後の式は $C(xy) \approx C(yx)$ であると仮定したときの近似式だが，以下では，この最後の式を NCD の定義式と考えることにする．

上記の定義式の妥当性を見るために，$C(x) \geq C(y)$ の場合を考えてみよう．すると

$$\mathrm{NCD}(x,y) \ = \ \frac{C(yx) - C(y)}{C(x)}$$

となる．これは 1 から式 (1) を引いた値に他ならない．

3. 階層的クラスタリング

比較したい対象 n 個の集合が与えられたとしよう．NCD を用いれば，その要素同士の対ごとに類似距離を測ることができる．その距離を $n \times n$ の

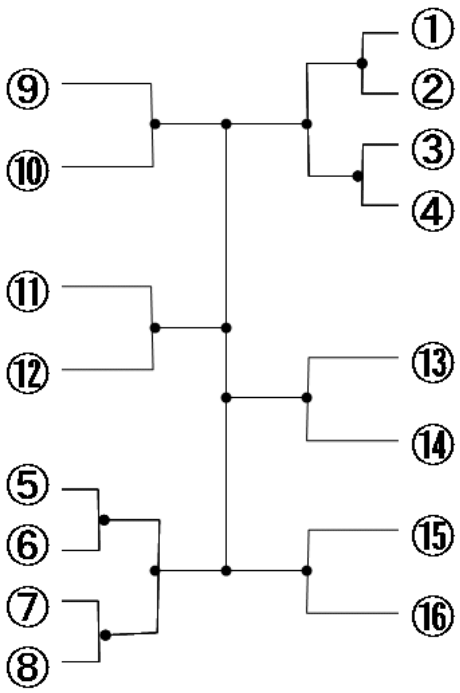
対称行列として表わしたのが距離行列である．この距離行列は，対象間の情報の近さ・遠さを数値的に表わしたものではあるが，その生の情報だけでは人間にとって理解しにくい．一般には，n が 3 より大きくなると，距離行列を見ただけで各データ間の関係の全体像をつかむのは難しい．我々にわかりやすいものにするためには，似たもの同士をグループ化するなど，さらにわかりやすいものに変形する必要がある．

データ解析では，似たもの同士のグループを クラスタ といい，グループ化することを クラスタリング という．我々の目標は，NCD による距離行列を用いて，対象となる n 個のデータをクラスタリングすることである．この場合，データが自然とクラスタを導出するのであり，クラスタ数などが前もってわからないのが普通である．この種のクラスタリングで最もよく使われているのが，階層型クラスタリング である．種の進化系統樹などに使われているクラスタリングのことで，計算論的生物学などで深く研究されている手法である．たとえば，図 1 のような木によるクラスタリングである．

階層型クラスタリングの中でも，もっとも精密といわれているのが「4 点法」と呼ばれる方法である．これは精密ではあるが，計算コストのかかる方法で，計算時間がデータ数 n の 4 乗もかかってしまう．もっと速い方法，たとえば計算時間が n の 2 乗ですむ方法なども提案されているが，それらは精密さに欠ける場合がある．一方，我々の解析では，僅かな距離の差も見逃さず，丁寧に解析するクラスタリングが望まれる．NCD の計算に使う圧縮プログラムの能力が非常に高いわけではないので，大きな差が出にくいからである．

そのような要請の中，我々は新しい 4 点法を開発した．これは，乱択型並列降下法と遺伝的プログラミングを併せたようなアルゴリズム（正確にはヒューリスティックス）である．このアルゴリズムは CompLearn のパッケージ[2] に含まれている．

アルゴリズムの方針を簡単に述べよう．目標とするのは，すべての対象データを葉に持つ 3 分木



次節で説明する**例 1** のデータ群に対して NCD を求め，その距離行列を元に，3 分木の形にクラスタリングした例．辺の長さなどは無関係．葉と葉を結ぶ道上の内部頂点（●の点）の数が，葉同士の「近さ」を表わしている．たとえば，　と　，　と　は共に 4 点なので，同程度の近さ．それに対し，　と　は少し遠い．

図 1　階層型クラスタリングの例

である．距離の近いものは近くに，遠いものは遠くに置くように木を構成するのである．与えられた n 個のデータ間の距離行列に対し，アルゴリズムは，まず，ランダムに n 個の葉と $n-2$ 個の内部頂点を持つ 3 分木を構成する．そのランダム木に対し，葉や部分木を交換しながら，実際の距離行列に即した距離関係になるように変形していくのである．より詳しい説明は文献[3] を参照されたい．

他の方法と違い，我々のアルゴリズムでは，少なくとも十分長い間実行していれば，いつかは最適解に到達することが保証されている．ただし，最適解を求める問題はＮＰ困難であることが知られているので，つねに，妥当な時間で最適解に到達することは望めない．しかしながら，実際のデー

タで実験すると，大抵は良い解に到達するようである．さらに計算の並列化の工夫を加え，我々は，対象データ数 n が 70 程度までならば，十分妥当な時間で処理するプログラム得ている．これは従来法では $n = 15$ 程度までだったのに比べ大幅な改良といえよう．

アルゴリズムは，木 T の他に，T が距離行列での遠近関係にどれくらい忠実かを表わす指標 $S(T)$ も求める．この $S(T)$ は 0（最悪）から 1（最良）までの値をとる．一般に，n 個の要素間の距離行列に対し，それを忠実に表現するには $n-1$ 次元の空間が必要である．それを単純な 3 分木で表現しようというのだから，忠実に表現できない部分が当然出てくる．たとえば，5 頂点に対する距離行列で，最悪の場合には，どのような木 T でも $S(T) < 0.8$ となってしまう場合もある．したがって，30 個程度のデータ数で，しかも $S(T) \geq 0.95$ の木が得られる場合には，類似距離と階層型クラスタリングによるグループ化が，かなりうまく行われたと見なしてよいだろう．

4. NCD の応用

正規圧縮距離 NCD と我々の階層型クラスタリング法を用いたデータの解析は，様々なデータ例に対して行われてきた．ここでは，そのうちの 3 例を紹介しよう．

例 1 種類の異なるデータ

最初の例は，まったく異なる 4 種類のデータに対し，何ら区別せずに我々の手法を適用した例である．対象としたのは，以下のような 4 種類，合計 16 個のファイルである．

～ ：4 種の哺乳類のミトコンドリア遺伝子
, は熊, はネズミ, は狐
～ ：文学作品からの抜粋
～ ：音楽 MIDI ファイル
, ：Linux x86 ELF 実行ファイル
, ：Java オブジェクトファイル

NCD の計算には `bzip2` を圧縮プログラムとして用いた．この結果，図 1 のような木 T が得られた．この木の忠実度を表わす尺度 $S(T)$ は 0.984 で，非常に高い．

見てわかるように，種類ごとにうまく分類されている．音楽データのうち， はロック (Hendrix)， はクラシック (Debussy) だったのだが，この分類もできている．

この例でもわかるように，正規圧縮距離を元にしたクラスタリングは，異なる種類のデータに対し，まったく同じに適用でき，しかも，各々でそれなりの効果を得ることができる方法なのである．この汎用性については，最近，他の研究者が，かなり大規模な実験が行ったが，やはり同様の結果が報告されている．

例 2 文学作品

図 2 に示したのは，ロシアの文学作品に対して適用した結果である．データは，WWW 上のサイトからダウンロードしてきたキリール文字で書かれたロシア語テキストである．

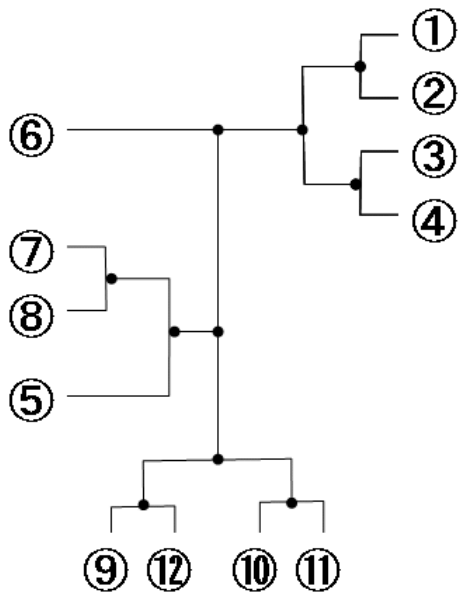
この場合も，NCD の計算には `bzip2` を用いた．得られた木 T に対する $S(T)$ 値は，この場合も $S(T) = 0.949$ とかなり高かった．ほとんどが，著者ごとに分類されていることがわかる．

なお，同じ作品で英語訳のテキストを分類する実験も行ったが，この場合には，著者できれいに分かれない場合もあった．翻訳者の同異による影響が混在していたためである．その他に，著者の性別や年代などによる違いを見出す実験等もある．

例 3 音楽

最後の例は音楽の分類である．インターネット上で配信される音楽は，近年，非常に多くなってきている．これらの音楽ソフトを提供するウェブサイトでは，何らかの方法によって音楽を分類する必要がある．適切な分類は，ユーザが好きな音楽を探すのにも役立つし，それを元に，ユーザへ適切な音楽ソフトを推薦することもできる．

これまで，そのような分類作業のほとんどは人手で行われてきた．しかしながら，最近では音楽



上図は，以下の 12 曲の MIDI ファイルに対して分類を行った結果である．忠実度は $S(T) = 0.968$ だった．

～　：平均律クラヴィーア曲集 2 巻
　　前奏曲集 1, 2，フーガ 1, 2　（バッハ）
～　：２４の前奏曲集，作品 28
　　#1, #15, #22, #24　（ショパン）
～　：ベルガマスク組曲，4 楽章　（ドビュッシー）

図 2　音楽の分類

ソフトの自動化についても，いろいろな研究が進んできている．我々の手法も，その有力な 1 つといえるだろう．図 3 に示したのは，12 曲の分類という小さな例だが，多数の多ジャンルの音楽の分類実験でも，よい成果が得られている[3]．

以上，3 つの例を紹介したが，我々の手法は，様々なデータで試されている．また，最近では，コンピュータウィルスの発見への応用など，実際の問題への応用も始まっている．紙面の都合上，関連する文献等を掲載できないが，similarity metric とか clustering by compression というキーワードで検索すれば，主要な結果や最新の情報を手に入れられるだろう．

ここで，実際の問題への適用に関して，1 つ重要な注意をしておく．我々の手法を適用して，よい成果を得るには，比較対象の符号化にも十分注意を払う必要がある．たとえば，テキストファイルを比較する際，Unicode で符号化いるものと，8 ビットのコードで符号化されているものを混ぜて比較しても，望ましい結果は得られない．コルモゴロフ記述量に基づく類似距離であれば問題はないのだが，我々が使える NCD は，あくまで，実際の圧縮プログラムを用いたものに過ぎないからである．

5. これまでの研究，そして，これからの研究

NCD に似た概念は，これまでにいろいろと提案され，現在の定義に落ち着いた．その足跡を追ってみよう．

情報距離としては，1992 年頃[5]，まず和 $K(x|y)+K(y|x)$ の形が考えられた．そのすぐ後に，max 形，$\max\{K(x|y), K(y|x)\}$　が導入され，距離の公理を満たすことや，その最適性が証明された．けれども，コルモゴロフ記述量の計算不可能性から，あくまでも理論的な研究にとどまっていた．

しかしながら，1999 年に，正規化版 $(K(x|y)+K(y|x))/K(xy)$ の近似が類似距離として導入され，バクテリアや哺乳類の進化系統樹を作成するのに用いられた（たとえば文献[6] 参照）．ここで，計算不可能なコルモゴロフ記述量に代わり，実世界にある圧縮プログラムを用いてそれを近似する，という大胆な発想が用いられたのである．この応用では，特殊な遺伝子配列のための圧縮プログラム `GenCompress` が用いられた．

最終的には 2001 年頃に，和ではなく max を取るものを正規化したものが導入され（文献[7] 参照），それが，距離の公理を満たし，しかも すべての 計算可能な正規距離を包括する距離であることが示された．これにより，本稿で紹介した NCD（コルモゴロフ記述量版）が “the” similarity metric として完成したのである．一方，階層型クラスタリング法を用いて，NCD（近似版）を元に多様なデータに対し統一的にクラスタリングを行う技術が提案され，2003 年頃より，様々なデータ群に対

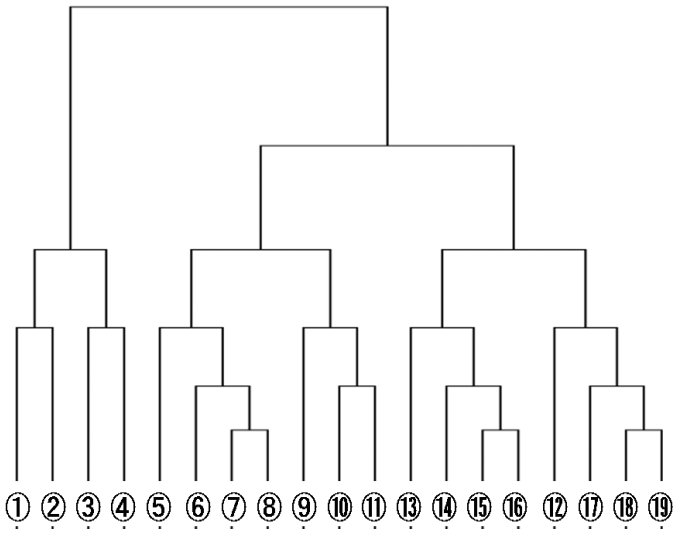
して応用されはじめた（文献[3] 参照）.

なお，同様の近似距離は，我々のグループとは別の研究グループでも提案されているが，ここでは紙面の都合上，紹介は割愛させて頂く（たとえば，文献[1] 参照）.

最後に，類似距離の新しい展開について述べよう．NCD は，文字列として定義されているデータ間の類似性を測る方法だった．一方，世の中には，ある固定した文字列として定義されていない対象がある．たとえば,「赤」のように，我々の常識から規定される概念などである．最近，このような概念に対しても，NCD を用いる試みが始められている．アイデアは簡単である．そうした概念に対する Google のような検索エンジンの結果に対し，NCD を用いるのである（詳しくは文献[4] 参照）．この「正規化グーグル距離」に対しては，新たな応用の発見など，今後の発展に期待したい.

参考文献

1) D. Benedetto, E. Caglioti, and V. Loreto, Language trees and zipping, *Phys. Review Lett.*, 88:4, 2002, 048702.
2) R. Cilibrasi, The CompLearn Toolkit, CWI, 2003–, `http://www.complearn.org/`
3) R. Cilibrasi and P.M.B. Vitanyi, Clustering by compression, *IEEE Trans. Information Theory*, 51:4, 2005, 1523–1545.
4) R. Cilibrasi and P. Vitanyi, Automatic meaning discovery using Google, Manuscript, CWI, 2004; `http://arxiv.org/abs/cs.CL/0412098`
5) M. Li and P.M.B. Vitányi, Reversibility and adiabatic computation: trading time and space for energy, *Proc. Royal Society of London*, Series A 452, 1996, 769–789.
6) M. Li, J.H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics*, 17:2, 2001, 149–154.
7) M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, The similarity metric, *IEEE Trans. Information Theory*, 50:12, 2004, 3250–3264.



作家と作品（番号は左図の葉番号）

F. ドストエフスキー
- 罪と罰
- 貧しき人びと
- 賭博者
- 白痴

I.S. ツルゲーネフ
- ルーヂン
- 貴族の巣
- その前夜
- 父と子

L.N. トルストイ
- 青春時代
- アンナ・カレーニナ
- 戦争と平和
- コサック

N.V. ゴーゴリ
- 肖像画
- 二人のイワンが喧嘩した話
- 死せる魂
- 隊長ブーリバ

M. ブルガーコフ
- 運命の卵
- 犬の心臓
- 巨匠とマルガリータ

キリール文字で書かれたロシア語テキストに対して `bzip2` を用いて NCD を計算し，それを元に 3 分木の形にしたのが左図である．図を見やすくするために内部点は省略した．

図 3　ロシア文学作品の分類

（ヴィタニ・ポール，CWI & U. Amsterdam）
（わたなべ・おさむ，東京工業大学）