

深層学習を用いたアルゴリズム的情報理論に基づく類似度推定

佐藤 哲†

パーソルキャリア株式会社†

1. はじめに

非構造化データの中で、自然言語や画像・音声など多くのデータは、文法や信号の特性や滑らかさなど自然に定まる規則を持っている。しかし人間社会の中で生成されるデータの中には、自然には定まらない外部要因的な規則が混入しているものが多い。例えば履歴書や職務経歴書のようなデータは、ある程度の形式があるため全体として自然言語による一つの文章にはならない、また、ある程度の形式の中の各項目は記述の自由度があり、生成者によって多様なデータとなる。結果として、このようなデータ同士を機械的に比較することは難しい。そこで本研究では、非構造化データ同士を比較することを可能とするために、データに含まれる情報の量を利用することでデータ同士の類似度推定手法を提案する。

2. アルゴリズム的情報理論に基づく類似度推定

アルゴリズム的情報理論では、Kolmogorov 複雑性によってデータの持つ情報の量を定義することが重要であり、またデータ圧縮によって Kolmogorov 複雑性は近似可能であることが示されている [1]。ここではデータ x に対する Kolmogorov 複雑性を $K(x)$ と表す。また、データ x とデータ y を結合したデータを xy と書き、データの生起確率を $p(x)$ 、データ長にたいする $K(x)$ の長さの割合を $c(x) = K(x)/|x|$ と表す。データはあるビットにより構成される文字 X_i から成るとする：

$$x = X_1 X_2 \cdots X_m$$

$K(x)$ は、データ x を出力する最小のプログラム長と定義され、単純なデータであれば小さくなり、規則性が無い複雑なデータであれば大きくなる。従ってデータを生成する確率分布には依存するが、データに含まれている情報の量を表すと考えられる。一般には $K(x)$ は計算不可能であるが、現実世界の問題に対してはデータ圧縮により近似計算が可能である。この情報の量から、あるデータ x に対し、データ y とデータ z のうちどちらが x と類似度が高いかを判定する問題を考える。データ x が持つ情報量は $K(x)$ である。ここにさらにデータ y を得た場合、情報量は $K(xy)$ となり、一般に情報量は増えるため $K(xy) > K(x)$ となる。この場合の増えた情報量 $K(xy) - K(x)$ は、データ x のもとでデータ y が持つ

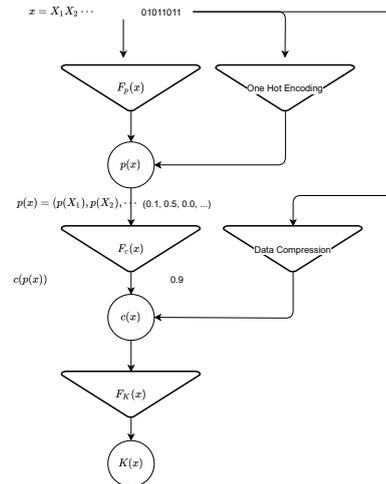


図 1: ニューラルネットワークのブロック図

ている情報量であり、条件付き Kolmogorov 複雑性 $K(y|x)$ と呼ばれる。つまり

$$K(y|x) = K(xy) - K(x) \quad (1)$$

である。この量は、 x のために y がもたらす情報量であり、 x と y が異なるほど大きくなり、逆に x と y が似ている場合は y が x にもたらす情報は少ないため小さくなる。つまり $K(y|x)$ は x と y の非類似度と言える。従って $K(y|x)$ の値が小さいほど x と y は類似度が高い。同様に、 y と異なるデータ z がデータ x のもとで持っている情報量は以下となる：

$$K(z|x) = K(xz) - K(x) \quad (2)$$

x を固定した時、 x に対して y と x のどちらが似ているかを調べるには $K(y|x)$ と $K(z|x)$ の大きさを調べれば良く、従って式 (1) 及び式 (2) より、 $K(xy)$ と $K(xz)$ の大きさを比べれば良い。

$K(x)$ はデータ圧縮により近似的に計算可能であるが、全ての入力データに対し圧縮処理を行うことはコストが高い。そこで、限られた教師データよりニューラルネットワークを用いて推定器を作り、評価したいデータを入力として推定する。学習及び推定に関するデータダイアグラムを図 1 に示す。推定機は 2 段階のニューラルネットワークで構成され、最初の出現頻度推定ネットワークでデータの出現頻度を推定し、次のデータ圧縮推定ネットワークで出現頻度列に基づき圧縮率を推定する。出現頻度推定ネットワークは、入力データを One hot encoding するための入力層、LSTM 層、出力層の 3 層で構成されて

A Similarity Measure based on Algorithmic Information Theory using Deep Learning

†Tetsu R. Satoh, PERSOL CAREER CO., LTD.



図 2: MNIST データの例

いる。データ圧縮推定ネットワークはシンプルな3層全結合ネットワークである。そして、推定された圧縮率により Kolmogorov 複雑性が推定され、その値によりデータ同士の相対的な類似度が推定できる。

3. 類似度推定実験

MNIST データを例に、データ同士の類似度を推定する実験を行う。データセットから、64 枚の画像データを用いて訓練し、64 枚のデータにより類似度推定のテストをする。簡単のため、画像データの濃淡は2ビット量子化の処理をし、ターゲットの文字種は数字10種類のうちの0から3の4種類を用いる。学習データ作成に必要な圧縮プログラムはxzを用いた。図2に、数字0から3のデータ例を示す。表1に、ラベルに対する条件付き Kolmogorov 複雑性(非類似度)の推定結果を示す。縦横の軸は数字データを表すラベルである。推定値は非類似度であるため、データが類似しているほど値は小さい。従ってタスクとしては同じ数字のラベルのデータ同士は値が小さくなって欲しいが、MNIST データは手書き文字データであり、図2からも分かるように同じラベルのデータ同士でもコンテンツが異なり必ずしも同じラベル同士の類似度が高いとは限らない。表1より、ラベル1はどのラベルに対しても非類似度が低く似ていると判断されること、逆にラベル0とラベル3は他のラベルに対し非類似度が高く似ていないと判断されていることが分かる。ラベル0同士、ラベル3同士の非類似度の高さからは、ラベル0とラベル3のデータの多様性が現れている。ただし、ニューラルネットワークによる推定は、現在のところ精度が良くない。表2に、データ圧縮プログラムにより計算した非類似度の実現値の例を示す。実現値は推定値を良く近似すると考えられるので、傾向が大きく異なる場合は推定機による推論が失敗している可能性

表 1: 非類似度の計算例

	0	1	2	3
0	250.238	<u>191.488</u>	224.083	226.040
1	217.132	<u>174.504</u>	197.374	<u>193.137</u>
2	<u>212.355</u>	<u>174.278</u>	<u>197.329</u>	207.684
3	257.660	<u>214.761</u>	228.174	248.617

表 2: 圧縮プログラムによる実現値

	0	1	2	3
0	234.037	<u>199.617</u>	229.940	231.106
1	<u>198.438</u>	<u>155.931</u>	<u>191.547</u>	<u>192.871</u>
2	229.095	<u>191.679</u>	225.936	226.088
3	229.849	<u>194.008</u>	224.719	227.876

がある。表1, 表2の場合は、ラベルが(2,2)の場合が特に異なる傾向を示しており、ニューラルネットワークによる推定が失敗していると考えられる。比較のため、Levenshtein 距離を計算した例を表3に示す。ラベル1が他のラベルに対し距離が小さく類似度が高いと判断されていることや、ラベル0及びラベル3は逆に他のラベルに対し距離が大きく類似度が低いと判断されていることは概ね表1の結果と似た傾向が現れている。ただし似た傾向の結果が得られるとは言え、提案手法に比べ他の手法は計算コストが高い。表4に、表1の計算に用いられた提案手法、表2の計算に用いられた圧縮処理、表3の計算に用いられた Levenshtein 距離の計算の、それぞれの処理時間を示す。学習時間は含めずに推論処理のみを4096枚のデータに対し実施したもので、時間の単位はミリ秒である。表4より提案手法による計算時間が最も短いことが確認でき、アルゴリズムの特性を考えるとデータ量の増大によりさらに差がついていくと考えられる。

以上の実験は、Amazon Elastic Compute Cloud (Amazon EC2) 上で実施し、ニューラルネットワークの実装は `deeplearning4j†` を、データ圧縮プログラムには `Apache commons compress††` 用いた。

4. おわりに

本研究では、非構造化データに対し適用可能な、アルゴリズムの情報理論に基づきデータ同士の相対的な類似度を推定する手法を提案した。MNIST データに対し適用実験を行い計算コストの優位性を示したが、MNIST データは分類問題に適するデータセットであるため、類似度推定の精度に関する議論が不十分となった。今後、ニューラルネットワークの改良や適切な入力データを適用により、提案手法の優位性を示していきたい。

参考文献

[1] P. D. Grünwald, P. M. B. Vitányi, Algorithmic Information Theory, Computing Research Repository (CoRR), abs/0809.2754, 2008.

表 3: Levenshtein 距離の計算例

	0	1	2	3
0	<u>140.800</u>	<u>137.500</u>	159.167	159.500
1	159.250	<u>49.375</u>	<u>133.500</u>	<u>126.333</u>
2	168.200	<u>148.000</u>	156.500	182.000
3	177.714	<u>139.500</u>	166.333	146.000

表 4: 非類似度・距離の計算時間

	計算時間 (ミリ秒)
提案手法	55794
圧縮プログラム	67608
Levenshtein 距離	245511

[†]<https://deeplearning4j.org/>

^{††}<https://commons.apache.org/proper/commons-compress/>