5A-03

トーラス埋め込みによる可逆なエントロピー削減方法

Reversible Entropy Reduction Method by Embedding in Torus

Yui Noma * †

概 要

IoT データなど一般のバイナリデータに対し、1 バイト文字列をトーラスに埋め込み、トーラスの自己同相写像を用いることで、1 バイト文字列のエントロピーを削減する方法を提案する。本提案手法は可逆であり、エントロピー削減されたファイルをエントロピー符号化により高効率な圧縮が可能となる。本稿では、実験により実際のデータに対しエントロピーが削減できることを示す。

1 はじめに

ファイルのエントロピーは、エントロピー符号化のファイルサイズの下限値を与える指標である。例えば算術符号化では、エントロピーから定まる下限に任意の精度で近づけるが、それを超えることは不可能である [1] [2]。csv データなど文字列に相関があり、もし、差分などの数値演算ができれば、“abc” “456” は “a11” や “411” に変更できる。その為、’1’ という文字の出現頻度が上がり、エントロピーが減少することが期待される。しかし通常の演算では、定義域と値域の大きさが変わってしまう。例えば 1 バイト整数値 x や y の和を表現するには 2 バイト必要である。その為、可逆演算にするには、計算途中に必要なバイト数を注意深く追う必要があった。

本稿では、ファイル中の 1 バイト記号列を高次元トーラスへ埋め込み、トーラスの自己同相写像を用いることで、可逆な操作でエントロピーを削減する手法を提案する。トーラスに埋め込むことにより、計算途中で必要なバイト数が増えず、エントロピーを変化させることができる。我々の知る限り、様々な種類のデータに対し可逆な操作でエントロピーを削減する手法は本提案が初めてである。

2 エントロピー削減の例

提案手法の詳細を説明する前に、具体例でエントロピーが下がることを示す。ファイルを 1 バイトの列としてみたとき、“0” と “128” のみから成り、“0” が 130 回、“128” が 90 回出現したとする。この時のエントロピーは $-\frac{130}{220}\log_2\frac{130}{220}-\frac{90}{220}\log_2\frac{90}{220}=0.97602$ である。また、ファイルを先頭から 2 バイトずつ組にしたとき、$(0,0)$ が 50 回、$(0,128)$ が 20 回、$(128,0)$ が 10 回、$(128,128)$ が 30 回出現したとする。これら数値を列ベクトルとし、行列 $A=\begin{pmatrix}1&1\\0&1\end{pmatrix}$ を乗算し、256 で剰余を取る操作を行う：

$$\begin{pmatrix}x_1\\x_2\end{pmatrix}\mapsto A\cdot\begin{pmatrix}x_1\\x_2\end{pmatrix}\bmod 256 \tag{1}$$

この操作の結果、$(0,0)\mapsto(0,0)$, $(128,0)\mapsto(128,0)$, $(0,128)\mapsto(128,128)$, $(128,128)\mapsto(0,128)$ となる。この操作により、“0” は 150 回、“128” は 70 回出現する。操作前より記号の出現頻度が偏り、エントロピーは $-\frac{150}{220}\log_2\frac{150}{220}-\frac{70}{220}\log_2\frac{70}{220}=0.90239$ と減少する。逆演算は、$A^{-1}\cdot\begin{pmatrix}x_1\\x_2\end{pmatrix}\bmod 256$ で与えることができる。つまり、可逆な操作でエントロピーを下げることが可能である。この際、データを元に戻すには、行列 A を保持しておけば良い。

この例の様なことがより一般の場合で成り立つ方法を提案し、また実際のデータに対しエントロピーが減少することを実験により示す。

3 トーラス埋め込みによるエントロピー削減

提案手法を説明する前に、提案手法が考えるトーラス中の整数格子とトーラスの自己同相写像を説明する。提案手法は、複数の構成要素から成り、それらは頻度に応じたトーラスへの埋め込み、変換行列の探索方法、変換後の記号列への逆変換データ保持方法である。それぞれを説明する。また、紙面の関係上 2 次元のみ説明するが、同様のことが任意の次元で成立する。

3.1 トーラス中の整数格子と自己同相写像

実数上で 0 と 256 を同一視した空間 $\mathbb{R}/256\mathbb{Z}\cong S^1$ を考える。S^1 上では、$[10]=[266]$ である。但し $[x]$ は $x\in\mathbb{R}$ の属する同値類である。以下では、表記を簡単にするため、同値類とその代表元を同一視し、[ ] を書かないことにする。

2 次元トーラス $T^2=S^1\times S^1$ を考える。この空間中の整数全体から成る集合 $(\mathbb{Z}/256\mathbb{Z})^2\subset T^2$ の元は、2 バイト記号と同じ数を持つ。

T^2 の自己同型写像は数学的に良く知られており [3]、2×2 整数係数行列 A で行列式が ±1 であるものを用いて以下のように表される。

$$\vec{x}\quad\mapsto\quad A\cdot\vec{x}\bmod 256 \tag{2}$$

この写像により、整数格子上の点は整数格子の上に写像され、また逆写像による像も整数格子上である。行列式が -1

の行列は、トーラスの向きを反転させるだけであるため、今後は行列式が 1 の行列だけを考える。

2×2 整数係数行列 $T^{(i,j)}$ を次式で定義する。

$$T^{(1,2)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad T^{(2,1)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \tag{3}$$

Serre の定理により [4]、任意の 2×2 整数係数行列 A は $T^{(i,j)}$ の積で表される。

$$A = (T^{(2,1)})^{\alpha_m} \cdots (T^{(2,1)})^{\alpha_2} (T^{(1,2)})^{\alpha_1} \tag{4}$$

但し、$\alpha_m \in \mathbb{Z}$。

3.2 提案手法

頻度に応じたトーラスへの埋め込み

我々は、1 バイト記号を出現頻度順に決まった方法でトーラスに埋め込む。この埋め込みは、後で説明する変換行列の探索を簡便にする効果がある。ファイルを 1 バイトの列、$d_1 d_2 d_3 \cdots$ と見なし、1 バイト記号の出現頻度を数える。記号の出現頻度順に 256 の 2^n 分の 1 の整数倍、つまり、“0”、“128”、“64”、“192”、“32”、…を割り当てる。出現頻度が高いほど、対応する “n”は小さいものとする。この写像を f とする。

変換行列の探索方法

ファイルを読み込み、出現頻度に応じた割り振り f を施したものを、$2 \times M$ 行列 D とする。

$$D = \begin{pmatrix} f(d_1) & f(d_3) & f(d_5) & \cdots \\ f(d_2) & f(d_4) & f(d_6) & \cdots \end{pmatrix} \tag{5}$$

エントロピーを下げるには、この行列 D に対し、行列を作用させ記号の出現頻度を偏らせる必要がある。その為、最も良い行列を探索する必要がある。整数格子 $(\mathbb{Z}/256\mathbb{Z})^N$ 上では、$(T^{(i,j)})^{256}$ は恒等写像となるため、式 (4) の α_m の取り得る範囲を $[0, 255]$ としても一般性を失わない。全行列を網羅し最も良い行列を選ぶこともできるが、行列の数が多すぎて非現実的である。

そこで、我々は行列の探索を貪欲に実行した。行列 D に対し $T^{(2,1)}$ を 0 回から 255 回乗算させ、エントロピーが最も減少した回数を α_1 とし、さらに $(T^{(2,1)})^{\alpha_1} D$ に対し $T^{(1,2)}$ を 0 回から 255 回乗算させ、エントロピーが最も減少した回数を α_2 とする。同様の手続きを繰り返すことでエントロピーを削減する行列を求める。

変換後の記号列への逆変換

行列を作用させた後のデータを各次元毎に出現頻度を数え、頻度順に 0, 1, 2, ... を割り当て、それを 1 列に並べる。

データ保持方法

データはエントロピーが下がったボディ部と、逆変換に必要なヘッダー部に分けられる。ボディ部は元のファイルと同じバイト数を持つ。

4 実験

実験に用いたデータセットは maximum compression [5] とした。図 1 に様々な次元でのエントロピー削減率をプロットした。エントロピー削減率はファイル種類に大きく依存したため、図を二つに分割し異なるスケールで図示している。図 1 から次元が上がるほどエントロピーが減少する傾向が見て取れる。最も減少率が大きいもので 30% を超えている。また、実際にエントロピーが減少した結果、算術符号化でのファイルサイズが減少する。図 2 ボディ部分に対し算術符号化を施したファイルサイズ比を表した。

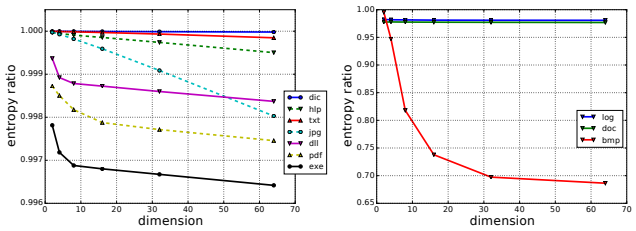

図 1: エントロピー削減率

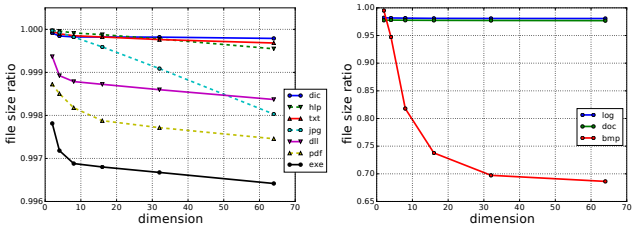

図 2: 圧縮ファイルサイズ比

5 結論

データをトーラスに埋め込むことにより、可逆な操作でエントロピーを減少させる手法を提案し、実験により、様々な種類のデータでエントロピーを減少を確認した。それによりエントロピー符号化で、圧縮後ファイルサイズが小さくなることを示した。エントロピーの減少手法は、我々の知る限り初めての提案であり、今後の高効率圧縮手法の開発を導くと期待する。

参考文献

[1] David Salomon. *Data Compression*. Springer.

[2] Jarek Duda. Asymmetric numeral systems as close to capacity low state entropy coders. *arXiv:1311.2540v1*, 2013.

[3] G Voyatzis and I Pitas. Chaotic mixing of digital images and applications to watermarking. *Proceedings of European Conference on Multimedia Applications*, 2, 1996.

[4] J-P. Serre. *A Course in Arithmetic*. Springer.

[5] Maximum compression `http://www.maximumcompression.com/`.