A-009

アルファベットサイズが未知の情報源に対する効率的なベイズ符号化法の一考察

A Note on an Efficient Bayes Coding Algorithm for Unknown Alphabet Sources

岩間大輝 * 石田崇 † 後藤正幸 ‡
Hiroki IWAMA Takashi ISHIDA Masayuki GOTO

1 はじめに

情報通信技術の発達に伴い, 取り扱われるデータは大容量化の一途を辿っている. 大容量のデータをそのまま扱うことは, 通信路の高負荷や保存媒体の容量不足につながるため, 情報圧縮の技術が重要である. 情報圧縮の技術の一つであるユニバーサル符号は未知の情報源の確率構造を推定しつつ圧縮を行う手法であり, 従来より様々の研究がなされている.

ベイズ最適性を有するベイズ符号 [6] はユニバーサル符号のひとつである. ベイズ符号は確率モデルの性能に依存しており, 実際のデータに対して有用な確率モデルクラスを用いる必要がある.

人間が自然に生成するテキストデータは出現記号の生起確率に偏りがあるだけではない. ある条件の下では必ずしも全シンボルが出現するわけではないと言った特徴を持つことが考えれる. これは単語や決まった語の並び等のパターンが例として挙げられる.

このような, 現実を踏まえ, Tjalkens ら [3], Rasihd ら [4] は, Context Tree Weighting Method を応用した記号の出現パターンを考慮した符号化法を提案した. Tjalkens らは, アルファベットが 2 値の場合による出現パターンの重み付けによる符号化を示した. また Rashid らは, アルファベットを多値に拡張し, 有限窓法による符号化法を提案している.

南茂ら [2] による研究では, 実際の情報源系列で情報源アルファベットのすべての記号が出現するとは限らないことに注目したベイズ符号化法を提案した. しかし, この研究では符号化確率の計算式は示されていたものの, アルゴリズム中にある組み合わせ計算の計算量が膨大となり実装が困難であった. また, 南茂らは independent and identically distributed 情報源 (i.i.d. 情報源) を想定しているが, 推定精度を上げるためにはマルコフ情報源への拡張も必要不可欠である. しかし, マルコフ情報源で行う場合も繰り返し計算を行うためには, 計算量の問題解決が必要不可欠である.

そこで, 筆者らはすでに, アルファベットサイズが未知の情報源に対する効率的なベイズ符号を提案している [1]. この研究では, 南茂らが示したアルゴリズム中の組み合わせ計算を簡略化するとともに木情報源への拡張を行った. 組み合わせ計算の簡略化は, 計算結果を一時的に保持することで次の計算に利用できる特性を活かして実現した. これにより南茂らの手法では計算量がアルファベットサイズの二乗程度であるのに対し, 定数オーダーに削減が可能になった.

本発表では, [1] で提案した手法の有効性を検証するため, 様々な条件の下で数値実験を行う.

実験の方法は大きく分けて 2 種類である. 一つは情報源を設定し, 人工的に生成したデータを用いた数値実験, であり, もう一つはカルガリーコーパスを使った自然データを用いた実験である.

* 早稲田大学大学院創造理工学研究科
† 早稲田大学メディアネットワークセンター
‡ 早稲田大学理工学術院

2 準備

2.1 木情報量

情報源アルファベットを $A = \{a_1, a_2, \cdots, a_{|A|}\}$ とする. 情報源から出現する長さ n の情報源系列を $x^n = x_1 x_2 \cdots x_n, x_t \in A$ と定義する $(t = 1, 2, \cdots, n)$.

時点 t から D だけ過去の情報源系列, $x_{t-D}, x_{t-(D-1)}, \cdots, x_{t-1}$ から一意に決まる状態 s_t によって次のシンボルの生起確率が決められている情報源が, マルコフ情報源である. D 次のマルコフ情報源は深さ D で完全 $|A|$ 分木の葉ノードに各シンボルの生起確率を付与した木構造で表現できる. 木情報源は深さが一定でない構造を許容した情報源である. 深さ D の木のモデルの集合を M, モデル $m \in M$ の状態集合を $S(m)$ とする. 図 1 において, 状態集合 $S(m)$ は $S(m) = \{s(1), s(00), s(10)\}$ となる. 葉ノード $s(1), s(00), s(10)$ は各シンボルの生起確率を保持されている.

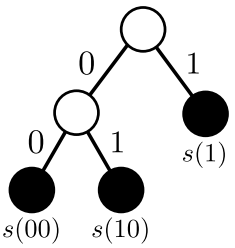

図 1: 深さ $D = 2$ の木情報源の例

2.2 ベイズ符号

ベイズ符号は, ベイズ冗長度を最小とするユニバーサル符号である. ベイズ符号は木情報源の最大次数 D は既知であるが, 真の木構造とそのパラメータが未知である場合を考える. このとき, 考えられるすべての木構造は, 深さ D の完全 $|A|$ 分木の部分木で表せる. そこで, すべての木のモデルの混合を取るため, 深さ D の完全 $|A|$ 分木を用意し, これを文脈木と呼ぶ.

$S_f\left(x^{t-1}\right)$ を系列 x^{t-1} から決まる文脈木の葉ノード s_t^D と根ノードをむすぶパス上にあるノード集合とする. $S_f\left(x^{t-1}\right) = \left\{s_t^0, s_t^1, \cdots, s_t^D\right\}$ とする.

$S_f\left(x^{t-1}\right)$ に含まれる深さ d のノード s_t^d における各記号の出現確率ベクトルを $\boldsymbol{\theta}\left(s_t^d\right) = \left(\theta_1\left(s_t^d\right), \theta_2\left(s_t^d\right), \cdots, \theta_{|A|}\left(s_t^d\right)\right)$ とし, これをすべての $s \in S$ について集めたパラメータベクトルを $\boldsymbol{\theta}$ とし, $P\left(s_t^d|x^{t-1}\right)$ を時点 t での, s_t^d の確率とすると, 符号化確率を

$$AP_D\left(x_t|x^{t-1}\right) = \sum_{d=0}^{D} \int_{\boldsymbol{\theta}(s_t^d)} P(x_t|x^{t-1}, \boldsymbol{\theta}(s_t^d), s_t^d) \times P(s_t^d|x^{t-1}) d\boldsymbol{\theta}(s_t^d) \quad (1)$$

で計算することができる.

3 南茂らによる従来研究

南茂らは, 情報源を i.i.d. 情報源と仮定し, 情報源のアルファベットサイズ $|A|$ に比べて系列に実際に出現する記号数 r^* が小さい情報源について考えベイズ符号化確率を求めるアルゴリズムを示した [2]. r_t を情報源系列 x^t 内で出現した記号数とすると, $t \to \infty$ のとき, $r_t \to r^*$ となる r^* を真の出現記号数と呼ぶ. r^* は有限時点では知ることができない. そのため, 真の出現記号数 r^* についても, ベイズ的に推定しながら符号化を進めている. この符号化法は, 各記号の出現確率の推定だけでなく, 出現記号数の事後確率分布 $P\left(j|x^t\right)$ の推定を行い, 重み付けをする方法で与えられる. (ただし, $j \in \{r_t, r_t+1, \cdots, |A|\}$). 推定した $P\left(j|x^t\right)$ を用いて重み付けを行って符号化確率を求める式が, 次式である.

$$AP_D\left(x_t|x^{t-1}\right) = \sum_{j=r_t}^{|A|} \binom{|A|-r_t}{j-r_t} P(j|x^{t-1}) Q_D\left(x_t|x^{t-1}, j\right) \quad (2)$$

ここで, $Q_D\left(x_t|x^{t-1}, j\right)$ は $N\left(x_t|x^{t-1}\right)$ を時点 $t-1$ までの x_t の出現回数として,

$$Q_D\left(x_t|x^{t-1}, j\right) = \frac{N\left(x_t|x^{t-1}\right) + \alpha}{t-1+j\times\alpha} \quad (3)$$

で与えられる. これはアルファベットサイズを j と仮定した際の記号の出現確率の推定量である. しかし, この式中に存在する組み合わせ計算は乗算回数が $2\left(|A|-r_t\right)^2$ に至る. したがって, アルファベットサイズが大きくなると, 計算機上での実装が難しくなるという問題がある. さらに, 木情報源を扱う場合, 計算量の合計は木の最大深さの値 D をかけた値になる. 南茂らの手法は確率の推定精度を向上させるものではあるが, このような観点から実装上の問題があった.

4 アルファベットサイズを未知の情報源としたベイズ符号化法

著者らは, 既に [1] でアルファベットサイズが未知の情報源としたベイズ符号化法を提案している. この手法は, アルゴリズム内の組み合わせ計算において, 直前の組み合わせ計算の計算結果を一時的に記憶し, 続けて計算に利用する再帰的なアルゴリズムを示し, 各ノードでは南茂らの手法と圧縮性能を変えることなく計算量の削減を達成する. また, この計算手法により, 計算量の削減が実現することで, アルファベットサイズの拡大と, 木情報源への対応が可能になった. $r_{s_t^d}$ をノード s_t^d における, 時点 t における, 出現記号数とする. 組み合わせ計算 $\binom{|A|-r_{s_t^d}}{j-r_{s_t^d}}$ は次式で与えられる.

$$\binom{|A|-r_{s_t^d}}{j-r_{s_t^d}} = \frac{(|A|-r_{s_t^d})!}{(j-r_{s_t^d})!\,(|A|-j)!} \quad (4)$$

(4) 式で表した通りに計算を行うと, 乗算回数は $2\left(|A|-r_{s_t^d}\right)$ であり, (2) 式で考えると, $j = r_{s_t^d}$ から, $j = |A|$ までを計算する必要がある. よって (2) 式全体では 乗算回数が $2\left(|A|\right)$ になることがわかる.

5 実験

本稿ではアルファベットサイズを未知の情報源としたベイズ符号化法 [1] の有用性を検証するため, 2 種類の方法で実験を行った. 一つは圧縮性能がどの様な推移になるかを確認するために人工データ対する実験を行った. もう一つはアルゴリズムのベンチマークとしてカルガリーコーパスに対する実験を行った. なお, [1] の手法の有効性を示すため比較手法としてアルファベットサイズを $|A|$ で既知としたベイズ符号化法とした.

5.1 人工データによる実験

人工データによる実験は各手法がエントロピーに収束するまでの収束の速さの比較を行う. さらに, アルファベットサイズや木の深さによる圧縮性能の違いを示す.

5.1.1 実験条件

実験条件は表 1 のとおりである. この実験条件の下で実験を行い, 一シンボルあたりの符号長を算出する.

表 1: 人工データ実験の情報源

	アルファベットサイズ	木の深さ
実験 1	256	0
実験 2	128	0
実験 3	64	0
実験 4	32	0
実験 5	256	2
実験 6	128	2
実験 7	64	2
実験 8	32	2

5.1.2 実験結果

実験結果は以下の図 2 ～ 9 の通りである. 図中の method1 は [1] の手法を method2 は比較手法を示している. この結果から, アルファベットサイズを未知の情報源としたベイズ符号化法は真の出現記号数がアルファベットサイズに等しいとき以外は真のアルファベットサイズが $|A|$ と仮定したものより推定精度がよくなることがわかる. それ以外の場合でも, 時

点 t が小さい時点ではラプラス推定法を使った手法が精度がいいことがわかる.

アルファベットサイズを未知の情報源としたベイズ符号化法の方が推定精度が上がる時点はアルファベットサイズが小さくなるにつれて早くなり, また木情報源にすることでより早くなることがわかる.

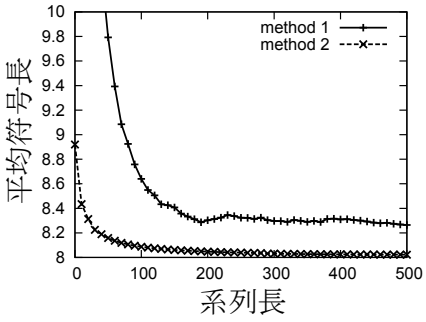

図 2: 実験 1

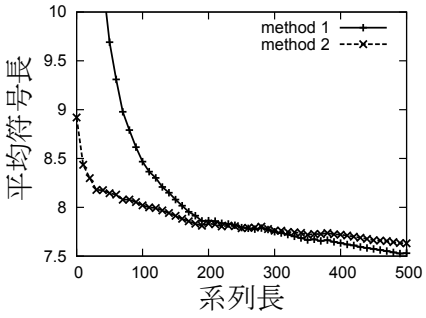

図 3: 実験 2

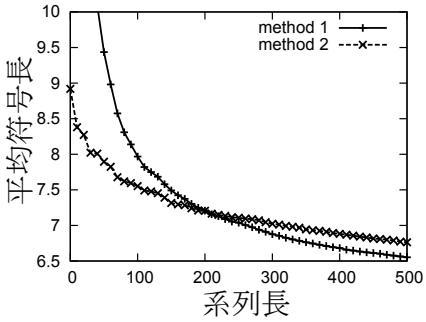

図 4: 実験 3

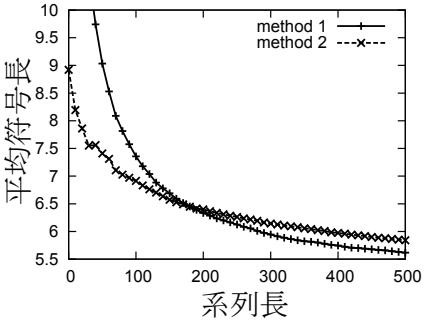

図 5: 実験 4

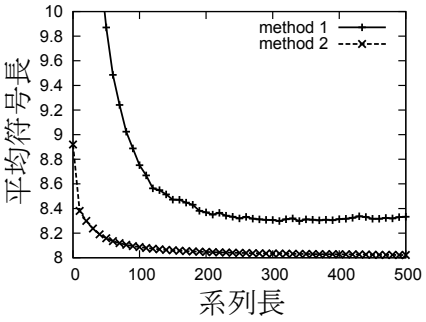

図 6: 実験 5

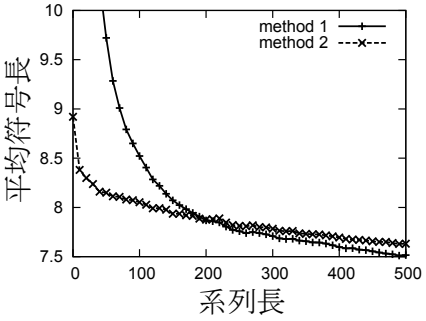

図 7: 実験 6

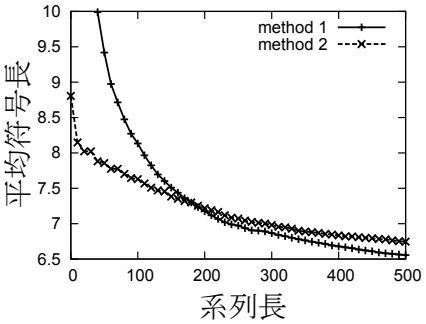

図 8: 実験 8

5.2 カルガリーコーパスを使った実験

この実験では自然データに対する, アルファベットサイズを未知の情報源としたベイズ符号化法の有用性を示すために比較実験を行う. 対象とするデータは符号化の研究において多く使われているカルガリーコーパス [5] である.

5.2.1 実験条件

本実験では上記の 2 つの手法をカルガリーコーパスに適用している.

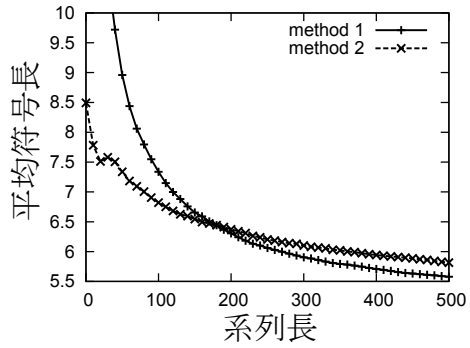

図 9: 実験 9

5.2.2 実験結果

実験結果は表 2 の通りである．どの場合においてもアルファベットサイズを未知の情報源としたベイズ符号化法の圧縮性能が良いことがわかる．

表 2: 自然データによる実験結果 (一部)

データ	サイズ [byte]	手法 1[byte]	手法 2[byte]
bib	117,543	49,727	72,503
paper3	47,628	22,560	27,285
paper4	15,582	6,927	7,931
paper5	12,276	6,710	7,775
paper6	39,126	194,15	24,010
progc	41,100	20,277	25,892

6 考察

5 章の実験から以下のことが考えられる．5.1 節で示した人工データによる実験では, ある時点 t からアルファベットサイズを未知の情報源としたベイズ符号化法が真のアルファベットサイズを $|A|$ としたベイズ符号化法より推定精度が良くなることがわかる．これは, 十分の系列長を有する情報源系列に対してはこの手法が有効であることがわかる．一見, 系列が短い時点で推定精度が低いことは, アルゴリズムとして問題があるように思える．しかし, データ圧縮アルゴリズムであるため, ある程度長い情報源系列に対し圧縮がなされることを考えると, この点は問題にならないと考えられる．

実際, 5.2 節で行ったカルガリーコーパスを用いた実験では性能がいいことが示されている．5.1 節の人工データの実験に比べ　自然データは単語や特定の語の並びなどがあるため, 情報源の木の深さが深いことが考えられる．そのため, ラプラス型の推定方法を使ったものより, アルファベットサイズを未知の情報源としたベイズ符号化法は圧縮率が良くなったということが考えられる．

7 まとめと今後の課題

本稿では, アルファベットサイズが未知の情報源に対するベイズ符号に対し, それまで不十分であった実験を拡充させた．実験は人工データと自然データに対して確率を推定すること行った人工データの実験にはアルファベットサイズと木の深さを変えた 8 種類の情報源から出現させた情報源系列を用い, 一方自然データの実験にはカルガリーコーパスを用いて実験を行った．これらの実験によりアルファベットサイズを未知の情報源としたベイズ符号化法の有効性を確認することができた．

しかし, 6 章で述べたとおり, 実験から真のアルファベットサイズが大きい場合, 時点が十分に長くないとき, アルファベットサイズを未知の情報源としたベイズ符号化法は真のアルファベットサイズを $|A|$ としたベイズ符号化法に対して有効でないことが分かる．データ圧縮アルゴリズムとしては有効であると考えらるが, 今後この手法は確率推定アルゴリズムとして利用していく場合, 十分なデータが必要になることはアルゴリズムとして解消することが望ましい．この点を解消するためには出現記号数の重みの事前分布を適切に設定することで解消することが考えられる．この事前分布の設定方法には今後検討の余地があり, 研究の課題である．

謝辞

著者の一人である岩間大輝は, 本研究を行うにあたり, 湘南工科大学, 小林学先生, 並びに早稲田大学嘱託研究員雲居玄道氏には多大なるご助言, ご支援を賜り, 深く感謝いたします．

参考文献

[1] 岩間大輝, 寺本賢一, 石田崇, 後藤正幸 “アルファベットが未知の木情報源に対する効率的ベイズ符号化アルゴリズム, ” 電子情報通信学会技術研究報告. IT, Vol.110, No.137, pp.1–6, July, 2010.

[2] 南茂 龍之介, 小泉 大城, 松嶋 敏泰 “記号の出現パターンを考慮した情報源に対するベイズ符号に関する研究” 電子情報通信学会技術研究報告. IT, Vol.106, No.184, pp.25–30, July, 2006.

[3] T. J. Tjalkens, P. A. J. Volf, and E. M. J. Willems “A Contex tree weighting method for text-generating sources” *Proc. IEEE Data Compression Conference*, p.472, March, 1997.

[4] M M. Rashid, and T. Kawabata “Analysis of zero-redundancy estimator with a finite window for Markovian source, ” 第 27 回情報理論とその応用シンポジウム予稿集, pp. 14–17, Dec. 2004.

[5] The Calgary Corpus, http://corpus.canterbury.ac.nz/descriptions/#calgary.

[6] T. Matsushima, H. Inazumi, and S. Hirasawa, “A class of distortionless codes designed by Bayes decision theory” *IEEE Trans. Inf. Theory,* Vol.37, No.5, pp.1288–1293, 1991.