ベイズ符号化法によって推定された木情報源の類似度を用いた自動文書分類

情報数理応用研究　　5210C004-0　岩間大輝
指導教員　後藤正幸

Text Classification by Similarity of Tree Sources Estimated from Bayes Coding Algorithm

IWAMA Hiroki

1　はじめに

情報機器や通信機器の普及に伴い，誰もが多量の情報の発信や受信をできる時代を迎えた．しかし，人間が情報を理解・認識できる量には限界があり，利用者が必要とする情報のみを受け取る要望は後を絶たない．そのため，機械による自動分類手法は今後も発展が望まれる技術となっている．

自動文書分類の分野では，汎用的な分類技術として，圧縮プログラムを用いた手法が研究されている [1]．これは，圧縮率を用いて文書間の類似度を比較する方法であり，比較対象となるデータのデータ構造や内容の事前知識を必要としない．そのため，文書であれば言語の違いなどにも対応することができる．今後，圧縮プログラムを用いた分類手法が発展すると DNA や画像など，文字情報以外の分類にも応用が考えられる．

文書分類に用いられる圧縮手法は，圧縮する系列の情報源の確率構造を推定しながら圧縮を進めるユニバーサル符号である．ユニバーサル符号は情報源の確率構造が未知のもとで圧縮を行い，系列が無限長になった時の符号長が，圧縮限界であるエントロピーに収束することが知られている．すなわち，ユニバーサル符号は，データ圧縮を進めるうちに，情報源の確率構造を学習する機能を内在しており，これを文書データの統計的構造の学習に援用していることになる．ユニバーサル符号の中で，圧縮手法として現在主流となっている Lempel-Ziv (LZ) 法 [2] は辞書式と呼ばれる圧縮手法の一つである．一方，辞書式とは異なるユニバーサル符号には統計型と呼ばれる圧縮手法がある．統計型の圧縮手法にはベイズ符号化法 [3], Context Tree Weighting (CTW) 法 [4] などがある．

LZ 法や CTW 法の圧縮率を用いた文書分類手法はすでに報告されている [5][6][7][8]．これらの手法は，文書 A を圧縮することで生成される辞書やモデルを使って文書 B を圧縮した時の符号長が両文書のデータサイズが無限大になったとき，文書 B から見た文書 A の Divergence に収束するという性質を利用している．しかし，有限長のデータに関しては実験的にその分類性能を示すことしかできない．また，圧縮率を利用する分類手法は，分類時に分類対象文書を学習データ数と同じ回数だけ圧縮して類似度を測る必要があるという計算量の問題もある．

これに対し，ベイズ符号化法は有限長のデータに対するベイズ最適性を有しており，圧縮の際に二次的に符号木が生成される．文書 A と文書 B を別々に圧縮して得られる符号木を，各々の確率モデルとみなして直接的に距離を測ることができれば，ベイズ最適性を有する確率構造 (混合予測分布の確率構造) を用いた類似性の判断が可能になり，分類性能の向上も期待できる．事前に学習データを圧縮しておくことで，分類時には分類対象データのみを圧縮すれば比較可能になるという点で，分類時の計算量の面からも効率的である．

そこで本研究では，ベイズ符号化法からの出力である推定された混合モデルと確率構造を用いて測定した情報源の距離を類似度として文書分類を行う手法を提案する．さらに，提案手法を著者推定の分類問題に適用し，その有効性を示す．

2　準備

2.1　圧縮率を用いた自動文書分類 [7]

情報源アルファベット A を $A = \{a_1, a_2, \ldots, a_{|A|}\}$ とする．情報源から出現する長さ n の情報源系列を $x^n = x_1 x_2 \ldots x_n$ と定義する．ただし，$x_t \in A$ である $(t = 1, 2, \ldots, n)$．ここでは，簡単のために $A = \{0, 1\}$ とする．

圧縮する系列を x^n, x^n を圧縮プログラムによって圧縮したあとの符号長を $C(x^n)$ とする．x^n の冗長度が高いほど圧縮率 $\frac{C(x^n)}{n}$ は小さくなる．

ここで分類対象データを $z^r = z_1 z_2 \ldots z_r$ とする．x^n と z^r の共通する部分系列が多くなったとき類似度は高いと考えられる．x^n と z^r を連接した系列を $x^n z^r$ としたとき，類似度が高ければ，ユニバーサル符号の学習が進むため，圧縮率 $\frac{C(x^n z^r)}{n+r}$ は小さくなることが期待できる．このように，圧縮率を用いた自動文書分類は，学習データと分類対象データの圧縮率を用いて，両者の類似度を推定し分類を行う方法である．

2.2　ベイズ符号化法

松嶋らのベイズ符号化法 [3] はベイズ最適性のもとで，木情報源から出現する系列の符号化確率を効率的に計算する手法である．本節では木情報源と効率的なベイズ符号化法について述べる．

2.2.1　木情報源

時点 t から $\overline{D}$ だけ過去の情報源系列, $x_{t-\overline{D}}, x_{t-(\overline{D}-1)}, \cdots, x_{t-1}$ から一意に決まる状態 s_{t-1} 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．

$\overline{D}$ 次のマルコフ情報源は，深さ $\overline{D}$ の完全 $|A|$ 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 $\overline{D}$ 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 $S(m)$ は葉ノードの集合 $S(m) = \left\{s^{(0)}, s^{(01)}, s^{(11)}\right\}$ となる．葉ノード $s^{(0)}, s^{(01)}, s^{(11)}$ には各シンボルの生起確率が保持されている．

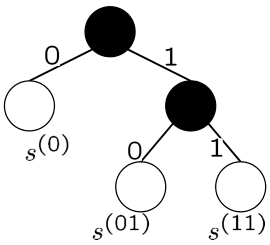

図 1．深さ $\overline{D} = 2$ の木情報源の例

2.2.2 効率的なベイズ符号化法 [3]

ベイズ符号化法は, ベイズ冗長度を最小とするユニバーサル符号である．いま，木情報源の最大次数 $\overline{D}$ 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, 考え得るすべての木構造は, 深さ $\overline{D}$ の完全 $|A|$ 分木の部分木で表せる．木構造の最大次数を $\overline{D}$ と仮定したとき，木モデルの数 $Md(\overline{D})$ は

$$Md(\overline{D}) = \begin{cases} 1 & (\overline{D} = 1) \\ Md(\overline{D}-1)^{|A|} + 1 & \text{otherwise} \end{cases} \tag{1}$$

となる．そのため混合したモデルには効率的な計算が必要である．松嶋らは木情報源に対して効率的な符号化法を提案している [3]．任意の葉ノードを s^D (ただし, $0 \leq D \leq \overline{D}$), s^D と根ノードを結ぶパス上のノード集合を $\mathcal{S} = \{s^0, s^1, \ldots, s^D\}$, $s \in \mathcal{S}$ とする．特に系列 x^{t-1} で決まる葉ノードを s^D_{t-1}, s^D_{t-1} と根ノードを結ぶパス上のノード集合を $\mathcal{S}_{t-1} = \{s^0_{t-1}, s^1_{t-1}, \ldots, s^D_{t-1}\}$, $s_{t-1} \in \mathcal{S}_{t-1}$ とする．ノード s_{t-1} における各記号の出現確率ベクトルを $\boldsymbol{\theta}(s_{t-1}) = \left(\theta_1(s_{t-1}), \theta_2(s_{t-1}), \ldots, \theta_{|A|}(s_{t-1})\right)$ 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, $P\left(s_{t-1}|x^{t-1}\right)$ を時点 t での s_{t-1} の確率とすると, 符号化確率 $AP_D\left(x_t|x^{t-1}\right)$ は以下のように計算することができる．

$$\begin{aligned} &AP_D\left(x_t|x^{t-1}\right) \\ &= \sum_{s_{t-1}\in\mathcal{S}_{t-1}} \int_{\boldsymbol{\theta}(s_{t-1})} P(x_t|x^{t-1}, \boldsymbol{\theta}(s_{t-1}), s_{t-1}) \\ &\qquad \times P(s_{t-1}|x^{t-1}) d\boldsymbol{\theta}(s_{t-1}) \\ &= P_C(x_t \mid s_{t-1} = s^0_{t-1}) \end{aligned} \tag{2}$$

$$\begin{aligned} &P_C(x_t \mid s_{t-1}) \\ &= \begin{cases} P(x_t \mid x^{t-1}, s_{t-1}) & (s_{t-1}\text{が葉ノードの時}) \\ (*) & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

$$\begin{aligned} (*) =& (1 - g(s_{t-1} \mid x^{t-1}))P(x_t \mid x^{t-1}, s_{t-1}) \\ &+ g(s_{t-1} \mid x^{t-1}) P_C(x_t \mid s'_{t-1}). \\ g(s_{t-1} \mid x^t) =& \frac{g(s_{t-1} \mid x^{t-1}) P_C(x_t \mid s'_{t-1})}{P_C(x_t \mid s_{t-1})}. \\ & s'_{t-1}\text{は } s_{t-1}\text{のパス上の子ノード.} \end{aligned} \tag{4}$$

また，松嶋らのベイズ符号化法は空間計算量を効率的に利用するために圧縮する系列によって木構造を変化させながら圧縮をしていくパトリシア木を利用している．そのため，系列ごとに符号木の構造が異なる．

2.3 定常マルコフ情報源の Divergence [9][10]

定常マルコフ情報源はモデルの構造が同じであれば情報源距離を測ることが可能である．モデルの構造が同じであるためには，次の 3 点が同じ条件である必要がある．

1. 状態の数
2. 状態遷移の構造
3. アルファベットサイズ

上記の条件が同じであるモデル同士の距離を測る際には次の式で Divergence (KL 情報量) を用いることができる．モデル m_z, m_x の Divergence $\mathcal{D}(m_z||m_x)$ は次の式で示される．

$$\begin{aligned} \mathcal{D}(m_z||m_x) =& \sum_{s^D} P(s^D|m_z) \sum_{a\in A} P(a|s^D, m_z) \\ &\times \log_2\left(\frac{P(a|s^D, m_z)}{P(a|s^D, m_x)}\right) \end{aligned} \tag{5}$$

木情報源はマルコフ情報源の表現の一つであるため，同じ構造のモデル同士であればこの手法による距離の比較が可能になる．

3 提案手法

ベイズ符号化法は系列を圧縮すると，その系列の符号化確率を出力するが，その計算過程で得られる符号木を情報源の確率構造を推定したモデルとして利用可能である．確率構造が木構造で与えることができれば 2.3 節で示したマルコフ情報の Divergence を利用することで，木情報源同士の距離の測定が可能である．そのため，ベイズ符号化法によって推定された情報源同士の距離の比較が可能である．

しかし，2.2.2 節で述べた通り，ベイズ符号は空間計算量を削減するために，木構造はパトリシア木を利用しており，圧縮するデータ毎に出力される木構造が異なる．そのため，ベイズ符号化法で出力される木構造のままでは，直接情報源同士の距離を測ることができない．本提案手法ではこの問題を解決し，情報源同士の距離を測り文書を分類する手法を提案する．

3.1 混合モデルからマルコフモデルへの木の変換

ベイズ符号化法で系列を圧縮した際に得られる符号木は木の混合モデルを計算するための統計量が保存された木である．このままでは情報源同士の距離を測ることができないため，等価な一つの確率モデルに変換する必要がある．

等価なマルコフモデルは全ての葉ノードとシンボル毎に混合モデルと等価な符号化確率 $P_{Ma}(a \mid s^D)$ を求めることで表現できる．

ベイズ符号化法では符号化確率を計算する際に葉ノードから根ノードを結ぶパス上におけるすべてのノードの統計量を用いて混合モデルの符号化確率を算出している．全ての葉ノード s^D とシンボルにおいて，$P_{Ma}(a \mid s^D)$ にベイズ符号化法の符号化確率 $P_c(a \mid s = s^0, \mathcal{S})$ を代入することで等価なマルコフモデルとなる．

$P_{Ma}(a \mid s^D)$ は次式によって求める．

$$P_{Ma}(a \mid s^D) = P_c(a|s = s^0, \mathcal{S}) \tag{6}$$

$$P_c(a|s, \mathcal{S}) = \begin{cases} P(a|s, \mathcal{S}) & (s \text{ が葉ノードの時}) \\ (1 - g(s, \mathcal{S}))P(a|s, \mathcal{S}) & \\ \quad +g(s)P_c(a|s', \mathcal{S}) & \text{otherwise} \end{cases} \tag{7}$$

s'は s のパス上の子ノード．

一方，木情報源同士の Divergence を測る際には葉ノード s の定常確率 $P(s)$ が必要になる．ベイズ符号化法において，$P(s)$ は求められていない．葉ノードの各シンボルの確率は状態遷移確率でもあるため，推定された確率構造から定常確率を求めることは，定常方程式を解くことによって理論上可能になる．しかし，モデル m の葉ノードの数 $|S(m)|$ は $|A|$ 進木で深さが $\overline{D}$ となるとき

$$|S(m)| \leq |A|^{\overline{D}} \tag{8}$$

になる．2 進木で深さを 10 の完全木を仮定した場合 $|S(m)| = 1024$ となり，定常確率を求めるには 1024 元連立方程式を解くことになり大変困難である．そこで本手法では葉ノード s^D の出現回数を $N(s^D)$ とし，$P(s^D)$ を次式の通り推定する．

$$P(s^D) = \frac{N(s^D)}{n} \tag{9}$$

3.2 木構造の展開

木情報源同士の距離を測る際に，比較する情報源同士が同じ木構造を持たない場合，Divergence の計算は非常に複雑になる．生成される木がすべて完全木であったり，同じ木構造であれば比較は可能である．

ベイズ符号化法で生成される符号木の構造は圧縮をするデータによってそれぞれ異なる．これはベイズ符号化法が符号化確率を求める際に，パトリシア木を用いたためで，必要な空間計算量を削減するためには大変有効な手法であった．

しかし，Divergence の計算を簡単に行うためには比較する符号木の構造が同じである必要がある．そこで，生成された符号木をすべて完全木に拡張することで，どの符号木とも比較が可能になる．

符号木を完全木に展開するとは，その符号木のノードの中で最大深さでないものを図 2 のように展開することである．符号木の深さ $D(D < \overline{D})$ の葉ノード s^D から展開された完全木の葉ノードを $\widetilde{s_w}$ とする (ただし，$1 \leq w \leq (\overline{D} - D)^{|A|}$)．この時 $\widetilde{s_w}$ が保持する次のシンボル a の生起確率 $P(a \mid \widetilde{s_w})$ は次式のように s^D と同じ生起確率を持つ．

$$P(a \mid \widetilde{s_w}) = P(a \mid s^D) \tag{10}$$

また，$\widetilde{s_w}$ はすべて等確率で出現すると仮定し，$\widetilde{s_w}$ の定常確率 $P(\widetilde{s_w})$ は s の定常確率 $P(s)$ を展開したノードの数 $(\overline{D} - D)^{|A|}$ で割った値

$$P(\widetilde{s_w}) = \frac{P(s^D)}{(\overline{D} - D)^{|A|}} \tag{11}$$

とする．

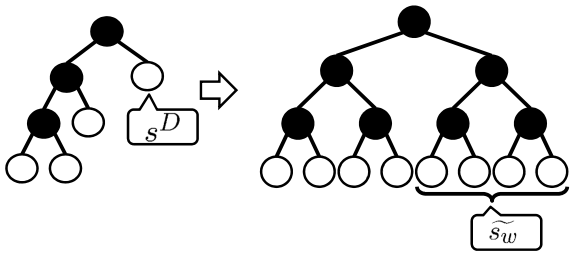

図 2． 木構造の展開の例

3.3 データの類似度測定の手順

学習データ $x_j^{n(c)}$ と分類対象データ z^r の類似度を測る手順を以下に示す．

step1 学習データ $x_j^{n(c)}$ と分類対象データ z^r のそれぞれをベイズ符号化器によって圧縮を行う．学習データと分類対象データの確率構造と木モデルの構造を推定する．

step2 step1 で得た入力データの推定された確率構造と木モデルを 3.1 節で示した手順によりマルコフ情報源への変換を行う．

step3 step2 で変換を行ったマルコフ情報源の木モデルを 3.2 節の計算によって完全木に展開を行う．

step4 $x_j^{n(c)}$ から推定されたモデルを m_x，分類対象データから推定されたモデルを m_z とし，step3 で拡張した完全木のモデルが持つ確率構造を (5) 式に代入して $x_j^{n(c)}$ と z^r の Divergence を測定し類似度として出力する．

4 評価実験

提案手法の性能を検討するため，文学作品の著者推定を事例として分類実験を行い，分類精度の評価を行う．

4.1 実験方法

実験では夏目漱石，森鴎外，宮沢賢治の 3 著者の作品の著者推定問題を行った．各著者につき 60 作品ずつ用意し，50 作品を学習データ，10 作品をテストデータとする．全著者の 150 作品の学習データと各テストデータとの類似度を比較し評価を行なった．分類方法は各手法で求めた類似度を k 近傍法 (k-NN) を用いて分類する．また本実験は木モデルの距離による比較をする提案手法 1, 2 とベイズ符号の圧縮率によって分類を行う比較手法 1, 2 との分類率の差の評価を行う．実験手法は次の通りである．

表 1． 実験手法

	類似度の計算法	木の深さ
提案手法 1	Divergence	15
提案手法 2	Divergence	10
比較手法 1	圧縮率	50
比較手法 2	圧縮率	15

なお提案手法においては比較手法 1 のように木の深さを 50 程度に設定すると，拡張した木を表現をする際の空間計算量が問題になるため，深さを 15 としている．

4.2 実験結果

実験結果を以下の図に示す．

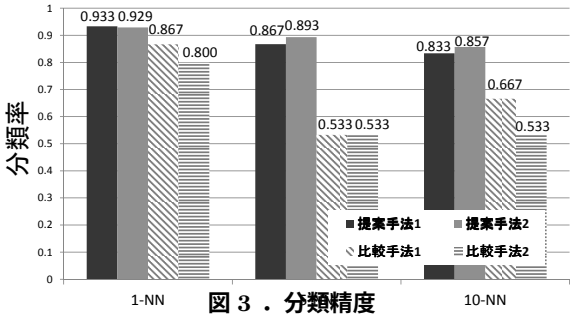

図 3 ． 分類精度

最も分類手法が良いのは 1-NN で分類した提案手法 1 であった．どの手法も 1-NN による分類が最も正解率が良く 80% を超す結果が得られた．また，1-NN，5-NN，10-NN のすべてにおいて提案手法が従来手法を上回る正解率となった．従来手法では木の深さを 50 とした比較手法 1 が 1-NN，5-NN，10-NN のすべてにおいて比較手法 2 よりも優れている．しかし提案手法では 5-NN，10-NN において木の深さが浅い提案手法 2 の正解率が最も良い結果となっている．

5 考察

本研究の実験では，提案手法は従来手法よりも圧縮の際の木の最大深さが浅いにも関わらず，分類精度の面で優れている結果になった．

ベイズ符号化法は，木の深さを大きく設定した方が圧縮率が良い．2.1 節で述べたように，圧縮率が良い方が分類精度も良いとされており，確かに圧縮率で比較した比較手法 1, 2 でも木の深い比較手法 1 の分類精度が良くな っている．しかし比較手法 2 よりも木の深さが浅い提案手法 1, 2 の方が分類精度が向上している．提案手法と従来手法どちらもベイズ符号化法で圧縮を行っているにも関わらず，分類精度に違いが出るのは類似度の計算方法に起因すると考えられる．

従来手法は学習データで生成した木モデルを用いてテストデータの圧縮を行っているが，テストデータの圧縮の際に学習データから推定される木構造が持つパスによる符号化確率を計算することで圧縮率が求まる．そして，この圧縮率が比較のための指標となる．しかし，テストデータを圧縮する際に学習データの木構造の全てのパスを用いて符号化確率が計算されるとは限らない．そのため，テストデータの圧縮の際に利用しなかった学習データの木構造の葉ノードが特徴的な確率構造を保持していた場合，圧縮率の変化には寄与しない．

一方，提案手法で用いた木モデルの距離を直接測る手法は全てのパスについて比較を行う．そのため，従来手法では比較できていなかった部分が提案手法では比較が可能になる．この点が分類精度の向上に寄与した部分であると考えられる．

6 まとめと今後の課題

本研究ではベイズ符号化法によって推定された木の距離を用いた文書分類法を提案した．また，評価実験により，提案手法の有効性を示した．本研究では自動文書分類を対象としたが，圧縮プログラム自体はどのようなデータでも圧縮可能であるため文書以外も比較が可能である．今後は絵画や音楽データの作者を推定などの応用も考えられる．また，空間計算量が許す限りベイズ符号化法の木の深さを深くすることができる．木の深さと分類精度の関連性について解析し，効率的な木の深さを求めることも今後の課題である．

参考文献

[1] David P. Coutinho and Mario A. T. Figueiredo, “Information Theoretic Text Classification Using the Ziv-Merhav Method,” *Lecture Notes in Computer Sciencee*, Vol. 3523, pp. 355–362, 2005.

[2] Jacob Ziv and Abraham Lempel, “Compression of individual sequences via variable-rate coding,” *IEEE Transactions on Information Theory*, Vol. 24, No. 5, pp. 530–536, Sep., 1978.

[3] T.Matsushima and S.Hirasawa “Reducing the Space Complexity of a Bayes Coding Algorithm using an Expanded Context Tree.” *IEEE International Symposium on Information Theory*, pp. 719–723, July, 2009.

[4] Frans M. J. Willems , Yuri M. Shtarkov and Tjallin J. Tjalkens, “The Context-Tree Weighting Method: Basic Propeties,” *IEEE Transactions on Information Theory*, Vol. 41, No. 3, May. 1995.

[5] 相澤彰子, “多クラス文書分類問題における Ziv-Merhav Crossparsing の適用と評価,” 情報処理学会論文誌 , Vol. 52, No.10, pp. 2953–2964, Oct., 2011.

[6] 安形輝, “圧縮プログラムを応用した著者推定,” *Library and information science*, no. 54, pp. 1–18, 2005.

[7] Zaher Dawy, Joachim Hagenauuer and Andreas Hoffmann, “Implementing the Context Tree Weighting Method For Content Recognition,” *Data Compression Conference*, p. 534, Mar., 2004.

[8] 小林学, 後藤正幸, 松嶋敏泰, 平澤茂一, “文脈木重みづけ法を用いた文書分類の誤り確率について,” 電子情報通信学会技術研究報告, vol. 111, no. 276, NLP2011-111, pp. 109-114, Nov., 2011.

[9] Haixiao Cai and Sanjeev R. Kulkarni, “Universal Divergence Estimation for Finitie-Alphabet Sources,” *IEEE Transaction on Information Theory*, Vol.52, No.8, Aug., 2006.

[10] R. M. Gray, “Entropy and Information Theory,” *New York: SpringerVerlag*, 1990.