LA-012 最適データ圧縮のための省スペースな近似アルゴリズム

A Space-Economical Approximation Algorithm for Optimum Data Compression

坂本　比呂志 *
Hiroshi Sakamoto

1. 研究の概要

本研究では，テキストを辞書式圧縮法によってできるだけ小さく圧縮する組み合わせ最適化問題に対して，準最適解を保証する規模耐性の高い線形時間アルゴリズムを構築する．この圧縮問題は，入力テキストに対する最小の文脈自由文法 (CFG) を構成する NP-困難な問題と等価であり，APX-困難であることも示されている [4]．また，入力長 n に対して，最適な圧縮の $O(\frac{\log n}{\log\log n})$ 倍以下の近似も困難であると予想されている．これまでに，LZ77 や LZ78 [7, 8] をはじめ，多くの実用的な圧縮法 ([2] が詳しい) が提案されているものの，最適解に対するそれらの圧縮の理論的な近似率はいずれも $\Omega(n^{\frac{1}{3}})$ より悪い [4]．そこで現在，このような理論的近似率と効率的実装との間のギャップを埋める研究が展開されている．最初のよい近似アルゴリズムは，Charikar [1] らによって示された準線形時間 $O(\log\frac{n}{g_*})$-近似である．ここで g_* は最適解のサイズである．また同時期に，Rytter [5] は，接尾辞木を用いた線形時間 $O(\log\frac{n}{g_*})$-近似アルゴリズムを構築した．その後，Sakamoto [6] は，BPE (byte pair encoding) を改良した線形時間 $O(\log^2 n)$-近似アルゴリズムを提案している．これらによる圧縮は最適解に飛躍的に近づいているが，いずれも接尾辞木などのテキスト全体に対する $O(n)$ 領域のデータ構造を必要とするため，データサイズの増加が主記憶を圧迫してしまう．例えば [6] の主記憶量は BPE の実装に依存し，そのひとつである Repair [3] は，$5n+4k^2+4k'+\sqrt{n}$ 文字分の主記憶を使用する．ここで k と k' はそれぞれ入力と出力アルファベットのサイズである．これに対して，本研究で提案する圧縮法は，各アルファベットがテキスト中に最初に出現する位置情報のみを使用する単純な手法に基づき，最適解に対する近似率 $O((\log g_*)(\log n))$ を保証する．前述の理由から，これは準最適な圧縮といえる．アルゴリズムは，高々 $k+k'$ 個の頂点を持つ二分木上の最近共通祖先 (lowest common ancestor) を毎回計算する必要があるが，この木は完全平衡二分木であるので，この値は，データ構造を構築することなく定数時間で計算できる．最終的な主記憶領域の合計は高々 $k+4k'$ 文字分であり，後半で示すように，$k'=O(g_*(\log g_*)(\log n))$ である．したがって k' の値は通常，入力テキストに比べてかなり小さく，現在知られている *polylog*-近似アルゴリズムの中で最も規模耐性が高い．また，例えば LZW が最終的に構築する辞書のサイズが $\Omega(g_* n^{2/3}/\log n)$ [4] であることを考慮すると，このアルゴリズムは，その他の実用的なアルゴリズムと比べても飛躍的に省スペースであることが保証できる．

*九州大学大学院システム情報科学研究院
hiroshi@i.kyushu-u.ac.jp

2. アルゴリズム

本研究では，整数アルファベットを仮定する．すなわち，各文字 a_i は整数 i によって関連付けられている．まず，最適化問題を定義する．**辞書式圧縮問題**とは，入力文字列 w に対して，$L(G)=\{w\}$ となる CFG G のサイズすなわち生成規則の長さの総和を最小化する問題である．以後，生成規則の集合 D を辞書と呼ぶ．次に，本論文で使用するいくつかの記法およびデータ構造を説明する．$[w]$ はテキスト w 中のすべての文字の集合を表す．$w[i]$ は w の i 番目の文字，$w[i,j]$ は $w[i]$ から $w[j]$ までの区間を表す．特に，$w[i,i+1]$ をペアと呼ぶ．文字 a が連続している始めから終りまでの区間を $w[i,j]=a^+$ などと表記する．

- T_d: 整数 $1,\ldots,d=2^\alpha$ を葉とする完全平衡二分木．
- $lca(i,j)_d$: T_d の葉 i,j の最近共通祖先の**高さ** (葉からの距離)．
- $array_w$: ある $[w]=\{a_1,\ldots,a_d\}$ に対して，次で定義される長さ d の配列: $array_w[i]=j \Leftrightarrow w[j]$ は a_i の最左出現．

$array_w[i]=j$ は T_d の j 番目の葉へのポインタを表す．d が 2 のべき数でないときは，ダミーノードを導入する．この様子を図 1 に示す．ただし，T_d は仮想的なもので陽には構成しない．最後に極大区間の概念を定義し，アルゴリズムを図 2 で与える．

定義 1 $w[i-1,i+1]=a_{j_1}a_{j_2}a_{j_3}a_{j_4}$ とし，各文字に対する $array_w$ の値がそれぞれ $\ell_1,\ell_2,\ell_3,\ell_4$ とする．このとき，$lca(\ell_2,\ell_3)_d > lca(\ell_1,\ell_2)_d, lca(\ell_3,\ell_4)_d$ ならば，ペア $w[i,i+1]$ は**極大**であるという．

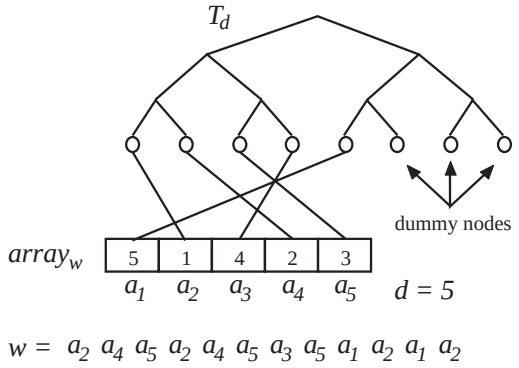

図 1: 極大ペアを計算するデータ構造．この例では，区間 $w[4,7]=a_2a_4a_5a_3$ が極大ペア $w[5,6]=a_4a_5$ を含む．

アルゴリズム LCA(w)

1. すべての $w[i,i+j]=a^+$ を文字 $A_{(a,j)}$ で置き換え，$A_{(a,j)} \to B_{(a,j)}C_{(a,j)} \in D$ を作る．$B_{(a,j)}C_{(a,j)}$ は，下の定義によって再帰的に展開される．
2. $[w]=\{a_1,\ldots,a_d\}$ および $array_w$ を計算する．
3. w のすべての極大ペアを適当な文字 A で置き換え，$A \to ab \in D$ とする．
4. 3 で置き換えなかったすべてのペアを左から順に適当な文字 A で置き換え，$A \to ab \in D$ とする．
5. w が以前と同じ長さならば $D \cup \{S \to w\}$ を出力し，そうでないならば 1 へ戻る．

$$B_{(a,j)}C_{(a,j)} = \begin{cases} A^2_{(a,j/2)}, & j \geq 4 \text{ が偶数} \\ A_{(a,j-1)} \cdot a, & j \geq 3 \text{ が奇数} \\ a^2, & \text{それ以外} \end{cases}$$

図 2: 圧縮アルゴリズム．連続した文字を取り除いた後，lca の値をもとにペアを置き換え，この処理を繰り返す．

3. 近似率と主記憶領域の解析

本研究における主な結果とその証明の概略を述べる．

定理 1 LCA(w) の主記憶量は，入出力アルファベットのサイズ k と k' に対して，高々$k+4k'$ 文字分である．

略証. $lca(i,j)_d$ は i と j のみから定数時間で計算できるので，必要なデータ構造とそのサイズは，再左出現位置のリストに対して $k+k'$ 文字分と辞書参照構造に対して $3k'$ 文字分のみである．したがって，主記憶の総サイズは $k+4k'$ 文字分以下である． □

定理 2 LCA(w) は $O(n)$ 時間で停止する．

略証. アルゴリズムの 1 から 4 までのステップは，定理 1 により線形時間で計算できる．ステップ 1 内で連続する同一文字は取り除かれているため，ステップ 3，4 内において，連続する 3 つのペアのうち少なくとも 1 つは置き換えられる．したがって，現在のテキスト長は直前のループ内でのテキスト長の $\frac{2}{3}$ 以下となる．以上から，走査するテキストの総延長は $O(n)$ である． □

定理 3 LCA(w) の近似率は $O((\log g_*)(\log n))$ である．

略証. w に含まれる独立した区間 f, f' が，同じ部分文字列を表すとする．$[w]$ の大きさを h とすると，アルゴリズムのステップ 3 が処理される段階で，隣り合う 2 つの極大ペアは，高々$\log h$ 以下の区間に現れる．したがって，これらの 2 つの同じ文字列は，アルゴリズムのステップ 3 と 4 で，それぞれ $\alpha\beta\gamma$ および $\alpha'\beta\gamma'$ のような文字列に圧縮され，α,α',β,β' の長さは，高々$\log h$ である．このことを用いて，ステップ 3，4 で新しく生成される非終端記号の数を見積もる．任意の w は，LZ-因子化 [7] によって，$w = f_1 \cdots f_m$ のように一意に分割され，一般に $m \leq g_*$ が成立する [5]．定義より，LZ-因子 f_m は左のある区間に含まれるが，その区間を含む最小の因子列を $F = f_i \cdots f_j$ とする．f_m と F の置き換えに食い違いができるのは，両端の f_i と f_j 内の，極大ペアを含まない連続な区間に限られる．したがって，w に対して 1 回のループで生成される異なる文字数を $\sharp(w)$ とすると，$\sharp(w) \leq \sharp(f_1 \cdots f_{m-1}) + O(\log h) = O(g_* \log h)$ となる．h は現在のアルファベットサイズなので，これを同様に展開すると，$h = O(g_* \log g_*)$ となり，結局 $\sharp(w) = O(g_*(\log g_* + \log\log g_*)) = O(g_* \log g_*)$ を得る．これは 1 回のループで生成される異なる非終端記号数の最大値である．また，定理 2 より，ループの継続回数は $O(\log n)$ であるので，結局，生成される非終端記号の総数は $O(g_*(\log g_*)(\log n))$ である．以上により，LCA(w) の近似率は $O((\log g_*)(\log n))$ である． □

4. まとめ

辞書式圧縮問題について，線形時間 $O((\log g_*)(\log n))$-近似アルゴリズムを示した．主記憶量は，入出力アルファベットサイズ k, k' に対して高々$k+4k'$ 文字分であり，これは知られている *polylog*-近似アルゴリズムの中で最良である．

参考文献

[1] M. Charikar, E. Lehman, D. Liu, R. Panigrahy, M. Prabhakaran, A. Rasala, A. Sahai, and A. Shelat. Approximating the Smallest Grammar: Kolmogorov Complexity in Natural Models. In *Proc. 29th Ann. Sympo. on Theory of Computing*, 792-801, 2002.

[2] T. Kida, Y. Shibata, M. Takeda, A. Shinohara, and S. Arikawa. Collage System: a Unifying Framework for Compressed Pattern Matching. *Theoret. Comput. Sci.* 298(1):253-272, 2003.

[3] N. J. Larsson and A. Moffat. Offline Dictionary-Based Compression. *Proceedings of the IEEE*, 88(11):1722-1732, 2000.

[4] E. Lehman and A. Shelat. Approximation Algorithms for Grammar-Based Compression. In *Proc. 20th Ann. ACM-SIAM Sympo. on Discrete Algorithms*, 205-212, 2002.

[5] W. Rytter. Application of Lempel-Ziv Factorization to the Approximation of Grammar-Based Compression. In *Proc. 13th Ann. Sympo. Combinatorial Pattern Matching*, 20-31, 2002.

[6] H. Sakamoto. A Fully Linear-Time Approximation Algorithm for Grammar-Based Compression. In *Proc. 14th Ann. Sympo. Combinatorial Pattern Matching*, 348-360, 2003.

[7] J. Ziv and A. Lempel. A Universal Algorithm for Sequential Data Compression. *IEEE Trans. on Inform. Theory*, IT-23(3):337-349, 1977.

[8] J. Ziv and A. Lempel. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Trans. on Inform. Theory*, 24(5):530-536, 1978.