STUDIA INFORMATICA 2003
Volume 24 Number 1 (52)

Roman STAROSOLSKI
Politechnika Śląska, Instytut Informatyki

NOWOCZESNE ALGORYTMY BEZSTRATNEJ KOMPRESJI DANYCH

Streszczenie. W niniejszym artykule omówiono zwięźle kilka nowoczesnych algorytmów z zakresu bezstratnej kompresji danych: algorytm predykcji przez częściowe dopasowanie, algorytm ważenia drzewa kontekstów, algorytm Burrowsa–Wheelera oraz algorytm Wolfa i Willemsa. Pod względem uzyskiwanych współczynników kompresji algorytmy te uznawane są obecnie za najlepsze wśród uniwersalnych algorytmów bezstratnej kompresji danych.

Słowa kluczowe: bezstratna kompresja danych, modelowanie, predykcja przez częściowe dopasowanie, ważenie drzewa kontekstów, algorytm Burrowsa–Wheelera, algorytm Wolfa i Willemsa.

MODERN LOSSLESS DATA COMPRESSION ALGORITHMS

Summary. In this paper we present a brief overview of four modern lossless data compression algorithms: Prediction by Partial Matching, Context Tree Weighting, Burrows–Wheeler Block Sorting Compression Algorithm and the Switching Method of Volf and Willems. In terms of average compression ratio presented algorithms are regarded as best among universal lossless data compression algorithms.

Keywords: lossless data compression, data modeling, PPM, CTW, Burrows–Wheeler Block Sorting, Switching Method of Volf and Willems.

1. Wprowadzenie

W niniejszym artykule omówiono kilka nowoczesnych algorytmów z zakresu bezstratnej kompresji danych, tj. algorytm predykcji przez częściowe dopasowanie, algorytm ważenia drzewa kontekstów, algorytm Burrowsa–Wheelera oraz algorytm Wolfa i Willemsa.

Celem niniejszego artykułu jest zwięzłe przedstawienie istoty ww. algorytmów. W artykule podano również literaturę dotyczącą tych algorytmów. Pełniejsze omówienie ww. algorytmów wykracza poza ramy niniejszego artykułu, dlatego też nie dążono do uzyskania takiego poziomu szczegółowości, jak w niektórych pozycjach z cytowanej literatury.

Polska terminologia w dziedzinie algorytmów kompresji danych jest niepełna, gdyż niektóre z algorytmów przedstawionych tutaj nie były do tej pory omawiane w literaturze polskojęzycznej. W przypadkach gdy brak jest uznanej polskiej nazwy, zdecydowano się stosować ogólnie przyjęte angielskie nazwy bądź skróty.

Wprowadzeniem do niniejszego artykułu jest [27], w którym przedstawiono podstawowe pojęcia, metody i algorytmy z zakresu bezstratnej kompresji danych, będące bazą dla algorytmów omówionych tutaj. Zagadnienia dotyczące zarówno podstaw bezstratnej kompresji danych, jak i algorytmów kompresji stratnej oraz specjalizowanych algorytmów kompresji konkretnych klas danych, takich jak np. obrazy, omówione są obszernie w książkach [15, 22, 23, 24, 33]. Przegląd algorytmów i standardów kompresji obrazów jest tematem oddzielnego opracowania [26].

Obecnie nie istnieje algorytm bezstratnej kompresji danych, który można by uznać za najlepszy. Przyczyną tego stanu rzeczy jest po części to, że współczynnik kompresji uzyskiwany przez algorytm nie jest jedynym kryterium jego oceny. Jeżeli jednak ograniczymy się do analizowania wyłącznie współczynnika kompresji, to okaże się, że nadal istnieje kilka algorytmów, spośród których żaden nie jest najlepszy dla każdego zbioru danych podlegającego kompresji. Obecnie za algorytmy uzyskujące średnio najlepsze współczynniki kompresji uważa się dwa adaptacyjne, statystyczne algorytmy kompresji. Są to: udoskonalany i rozwijany od prawie dwudziestu lat algorytm predykcji przez częściowe dopasowanie (omówiony w punkcie 2) oraz prawie dwukrotnie młodszy algorytm ważenia drzewa kontekstów (punkt 3). Algorytm Burrowsa–Wheelera (punkt 4) uzyskuje średnio współczynniki kompresji nieznacznie gorsze niż dwa ww. algorytmy, ma jednak od nich istotnie mniejsze złożoności, a dla niektórych zbiorów danych uzyskuje lepsze współczynniki kompresji. Interesujący jest również ostatni z przedstawionych algorytmów algorytm Wolfa i Willemsa (punkt 5). Pozwala on wykorzystać dwa różne, adaptacyjne modele danych (na przykład model algorytmu predykcji przez częściowe dopasowanie oraz model ważenia drzewa kontekstów) i uzyskać współczynniki kompresji lepsze niż indywidualnie każdy z algorytmów oryginalnie używających jednego z tych modeli. Zestawienie współczynników kompresji uzyskiwanych przez wybrane implementacje omawianych algorytmów zawarto w punkcie 6. Artykuł kończy podsumowanie (punkt 7).

2. Predykcja przez częściowe dopasowanie

Predykcja przez częściowe dopasowanie (ang. Prediction by Partial Matching, PPM) to, opublikowany w roku 1984 przez J.G. Cleary'ego i I.H. Wittena, algorytm adaptacyjnego kompresora statystycznego. Algorytm ten był następnie rozwijany i udoskonalany przez wielu badaczy [6, 9, 11, 20, 21]. Model danych algorytmu PPM stosowano również w kompresorach słownikowych [19] oraz jako narzędzie do analizy tekstów w języku angielskim [28, 29].

W modelu danych algorytmu PPM przechowuje się liczniki wystąpień symboli w kontekstach o różnej długości. Maksymalną długość kontekstu ogranicza się z góry, wartość tego ograniczenia nazywamy rzędem modelu PPM. Istnieją jednak warianty algorytmu PPM (PPM*) [10] o nieograniczonych długościach kontekstów.

Kodując dany symbol wyszukujemy w modelu najdłuższy kontekst pasujący do przedrostka tego symbolu w komunikacie. Następnie sprawdzamy, czy symbol wystąpił już w wyszukanym kontekście. Jeżeli wystąpił, to określamy rozkład prawdopodobieństwa symboli w tym kontekście na podstawie liczby wystąpień symbolu danego i wszystkich innych, po czym można już przeprowadzić kodowanie koderem entropijnym. W przeciwnym przypadku kodujemy dodatkowy symbol, tzw. *escape* i powtarzamy powyższą operację dla kontekstu o 1 mniejszego. Podstawową różnicą pomiędzy różnymi wariantami algorytmu PPM jest sposób określania prawdopodobieństwa wystąpienia symbolu *escape*. Model PPM zawiera kontekst o rzędzie oznaczanym umownie przez -1, który zawiera wszystkie symbole alfabetu źródła i jest stosowany w przypadku, gdy w kodowanym komunikacie po raz pierwszy pojawia się jakiś symbol. Wyznaczając rozkład prawdopodobieństwa w kontekstach krótszych niż najdłuższy pasujący, bierzemy pod uwagę tylko te symbole, które nie zostały wykluczone przy analizie dłuższych kontekstów. Po zakodowaniu danego symbolu model danych uzupełnia się o informację, iż w kontekście, którego użyto do wyznaczenia prawdopodobieństwa dla zakodowanego symbolu, symbol ten wystąpił.

3. Ważenie drzewa kontekstów

Opublikowany w roku 1993 przez F.M.J. Willemsa, Y.M. Shtarkova i T.J. Tjalkensa algorytm ważenia drzew kontekstów (ang. Context-Tree Weighting, CTW) jest adaptacyjnym modelem danych dla źródeł binarnych [2, 30, 34, 35]. W przypadku komunikatów ze źródeł o większych rozmiarach alfabetu źródła zastosowanie modelu CTW możliwe jest po

dokonaniu transformacji komunikatu na odpowiednią liczbę komunikatów binarnych. Znane są również warianty algorytmu CTW [25] przeznaczone do bezpośredniego modelowania komunikatów ze źródeł o rozmiarze alfabetu większym od 2.

W algorytmie CTW buduje się binarne drzewo kontekstów o długościach nie przekraczających pewnej wartości d. Kierunki odgałęzień kolejnych krawędzi ścieżki od korzenia do danego węzła określają kolejne symbole kontekstu skojarzonego z danym węzłem.

Liśćmi drzewa są konteksty o długości d. Przyjmuje się, iż symbole występujące w danym kontekście U generowane są przez źródło bezpamięciowe, tj. prawdopodobieństwo wystąpienia w danym kontekście symbolu 1 nie zależy od pozostałych symboli i wynosi Θ_U. Założenie to powoduje, iż można zaniedbać kolejność, w jakiej pojawiały się symbole, a co za tym idzie — zamiast przechowywać ciąg symboli, które wystąpiły w kontekście, wystarczy zliczać wystąpienia poszczególnych symboli. Wartość Θ_U nie jest znana, w miarę pojawiania się nowych symboli jest ona szacowana z coraz większą dokładnością. W algorytmie CTW przypisuje się liściowi drzewa wagę równą prawdopodobieństwu wystąpienia ciągu symboli, które się już w tym kontekście pojawiły.

Dla kontekstów krótszych od d, czyli węzłów nie będących liśćmi drzewa kontekstów, prawdopodobieństwo przypisane ciągowi symboli, które wystąpiły w kontekście (wagę węzła), można wyznaczać na 2 sposoby. Można podobnie jak w przypadku liści drzewa kontekstów założyć, iż jest to ciąg wygenerowany przez źródło bezpamięciowe i prawdopodobieństwo jego wystąpienia określić na podstawie liczby wystąpień poszczególnych symboli alfabetu. Wagę tę można również wyznaczyć na podstawie wag synów danego węzła, czyli prawdopodobieństw ciągów w tych kontekstach, dla których dany kontekst jest przyrostkiem. Zamiast wybierać jedno z powyższych rozwiązań w algorytmie CTW wyznacza się ich sumę ważoną.

Waga przypisana korzeniowi drzewa to prawdopodobieństwo wystąpienia całej, już przetworzonej części komunikatu. Po zakodowaniu kolejnego symbolu z komunikatu należy uaktualnić liczniki symboli i wagi we wszystkich węzłach na ścieżce łączącej korzeń z liściem odpowiadającym kontekstowi tego symbolu.

Rozkład prawdopodobieństwa symboli alfabetu źródła wymagany przez koder entropijny do zakodowania kolejnego symbolu komunikatu można wyznaczyć poprzez symulację wstawiania do modelu poszczególnych symboli alfabetu i porównanie wag korzenia po wstawieniu poszczególnych symboli.

4. Algorytm Burrowsa–Wheelera

W algorytmie opublikowanym w roku 1994 przez M. Burrowsa i D. Wheelera [7] komunikat ze źródła przed zakodowaniem z użyciem kodera entropijnego przekształcany jest przez odwracalne transformaty: Burrowsa–Wheelera oraz Move To Front.

Transformata Burrowsa–Wheelera wyznaczana jest w następujący sposób: przekształceniu podlega, oznaczony tutaj przez D, ciąg symboli komunikatu o długości n. W zależności od implementacji algorytmu, D jest albo całym komunikatem, albo odpowiednio długim (kilkaset tysięcy symboli) jego fragmentem. Tworzy się macierz M o rozmiarach $n \times n$, której wiersze to kolejne przesunięcia cykliczne przekształcanego ciągu. Następnie sortuje się (porządek alfabetyczny) wiersze macierzy M uzyskując macierz M'. Wynikiem transformaty Burrowsa–Wheelera jest pozycja ciągu D w M' oraz ostatnia kolumna macierzy M', którą oznaczymy przez D'. Transformata Burrowsa–Wheelera prowadzi do nieznacznej ekspansji danych, gdyż oprócz symboli zawartych w ciągu D wyprowadzany jest indeks tego ciągu w macierzy M'. Dla typowych komunikatów, na przykład tekstów, ciąg D' zawiera długie ciągi powtarzających się symboli (razem zgrupowane w D' są te symbole z D, po których w D występują ciągi o wspólnych przedrostkach).

Ciąg D' następnie jest przekształcany za pomocą transformaty Move To Front [5]. Tworzona jest lista L zawierająca wszystkie symbole alfabetu źródła. Następnie kolejno każdemu symbolowi d_i z D', począwszy od pierwszego, przyporządkowywana jest pozycja d_i w L, po czym d_i jest przemieszczany w liście L na początek. Przyjmuje się, że początek listy to pozycja 0. Wynikiem transformaty Move To Front jest ciąg D'' liczb przyporządkowanych kolejnym symbolom D'. Dla typowego ciągu D' najczęściej występującą liczbą w ciągu D'' jest 0, a częstość występowania poszczególnych liczb w D'' maleje gwałtownie wraz ze wzrostem ich wartości. Ciąg D'' kodowany jest kompresorem statystycznym z modelem danych zerowego bądź niewielkiego rzędu.

Zaletą algorytmów, w których stosuje się transformatę Burrowsa–Wheelera [1, 3, 4, 16, 17, 18], jest niewielka złożoność czasowa i pamięciowa połączona z uzyskiwanymi przez te algorytmy współczynnikami kompresji dorównującymi najlepszym obecnie znanym algorytmom. Badania nad algorytmami, w których stosuje się transformatę Burrowsa–Wheelera, prowadzone są również w Instytucie Informatyki Politechniki Śląskiej, a uzyskane przez S. Deorowicza rezultaty [12, 13, 14] są obecnie najlepsze spośród algorytmów wykorzystujących transformatę Burrowsa–Wheelera.

5. Algorytm Volfa i Willemsa

Opisany w roku 1998 przez P.A.J. Volfa i F.M.J. Willemsa [31, 32] algorytm pozwala wykorzystać dwa różne, adaptacyjne modele danych. Kodowany komunikat jest analizowany równolegle przez oba modele. Idea algorytmu polega na kodowaniu entropijnym części symboli komunikatu na podstawie rozkładu prawdopodobieństwa wyznaczonego przez pierwszy z modeli danych, a pozostałych symboli z zastosowaniem drugiego z modeli. Podział komunikatu na fragmenty kodowane z użyciem tego samego modelu danych jest przeprowadzany na podstawie analizy skutków użycia do kodowania rozkładów prawdopodobieństw symboli uzyskanych przez oba modele. Aby możliwa była dekompresja, oprócz symboli komunikatu kodowana jest również tzw. sekwencja przejść, tj. ciąg bitów określający pozycje początków kolejnych fragmentów kodowanych z użyciem tego samego modelu. Dąży się do wyznaczenia takiej sekwencji przejść, która minimalizuje łączną długość zakodowanego komunikatu i zakodowanej sekwencji przejść.

Autorzy algorytmu proponują kilka jego wariantów o różnych złożonościach. Wyniki eksperymentów z modelami danych, takimi jak między innymi CTW czy model danych algorytmu PPM, pokazują, że zastosowanie każdego z wariantów algorytmu pozwala na uzyskanie współczynników kompresji lepszych niż w przypadku pojedynczego modelu danych.

6. Porównanie wyników algorytmów

W niniejszym punkcie zestawiono współczynniki kompresji uzyskiwane przez omawiane algorytmy. Algorytmy historycznie starsze, tj. algorytm predykcji przez częściowe dopasowanie oraz algorytm Burrowsa–Wheelera, od czasu powstania zostały znacznie udoskonalone — do pierwotnych wersji wprowadzono wiele oryginalnych modyfikacji istotnie wpływających na wyniki uzyskiwane przez te algorytmy. Dlatego też w przypadku tych algorytmów w zestawieniu zawarte są wyniki zarówno wczesnych (pierwotnych) wariantów tych algorytmów, jak i najnowszych ich wersji, uzyskujących najlepsze współczynniki kompresji.

Zbiór danych testowych, będący podstawą zestawienia wyników algorytmów, to zaproponowany w roku 1989 i od tego czasu często stosowany w porównaniach algorytmów kompresji danych korpus Calgary. Twórcami korpusu Calgary są T. Bell, I.H. Witten oraz J.G. Cleary [8]. Współczynniki kompresji zamieszczone w tabeli 1 wyrażone są w bitach na symbol: z/e, gdzie e oznacza liczbę symboli kompresowanego pliku, $e > 0$, a z liczbę bitów pliku po skompresowaniu. Kompresja jest tym lepsza, im wartość

tak zdefiniowanego współczynnika kompresji mniejsza. W zestawieniu zawarto wyniki następujących wersji algorytmów:

- PPMC — standardowa wersja algorytmu predykcji przez częściowe dopasowanie; wyniki zaczerpnięto z [11],
- cPPMII — wariant algorytmu predykcji przez częściowe dopasowanie cPPMII autorstwa D. Shkarina (rząd 64); wyniki zaczerpnięto z [14],
- CTW — algorytm ważenia drzewa kontekstów; wyniki zaczerpnięto z [4],
- BW — pierwotna wersja algorytmu Burrowsa–Wheelera; wyniki zaczerpnięto z [7],
- BW-SD — wariant algorytmu Burrowsa–Wheelera autorstwa S. Deorowicza (metoda Weighted Frequency Count); wyniki zaczerpnięto z [14],
- VW — algorytm Wolfa i Willemsa (użyto modeli danych algorytmu ważenia drzewa kontekstów oraz predykcji przez częściowe dopasowanie, liczba przejść ograniczona była do 5000); wyniki zaczerpnięto z [32].

Tabela 1

Współczynniki kompresji algorytmów [bity na symbol]

Plik	Długość pliku [B]	Algorytm					
		PPMC	cPPMII	CTW	BW	BW-SD	VW
bib	111261	2.110	1.661	1.790	2.067	1.892	1.714
book1	768771	2.480	2.120	2.190	2.487	2.266	2.150
book2	610856	2.260	1.745	1.870	2.130	1.957	1.820
geo	102400	4.780	3.868	4.460	4.451	4.139	4.526
news	377109	2.650	2.088	2.290	2.592	2.409	2.210
obj1	21504	3.760	3.346	3.680	3.978	3.701	3.607
obj2	246814	2.690	1.891	2.310	2.636	2.417	2.245
paper1	53161	2.480	2.121	2.250	2.553	2.400	2.152
paper2	82199	2.450	2.112	2.210	2.514	2.345	2.136
pic	513216	1.090	0.668	0.790	0.835	0.714	0.764
progc	39611	2.490	2.104	2.290	2.582	2.428	2.195
progl	71646	1.900	1.343	1.560	1.801	1.671	1.482
progp	49379	1.840	1.336	1.600	1.789	1.670	1.460
trans	93695	1.770	1.136	1.340	1.570	1.454	1.256
średnio		2.482	1.967	2.188	2.428	2.247	2.123

Wyniki zawarte w tabeli pokazują, w jakim stopniu udało się na przestrzeni lat udoskonalić algorytm predykcji przez częściowe dopasowanie oraz algorytm Burrowsa–Wheelera. Poprawa jest widoczna wyraźniej w przypadku algorytmu predykcji przez częściowe dopasowanie, który spośród algorytmów omawianych w niniejszym artykule jest algorytmem najstarszym. Bardziej obszerne porównanie wyników uzyskiwanych przez większą liczbę wariantów omawianych algorytmów można znaleźć w [14].

7. Podsumowanie

W niniejszym artykule omówiono zwięźle kilka nowoczesnych algorytmów z zakresu bezstratnej kompresji danych oraz podano bibliografię dotyczącą tych algorytmów. Przedstawiono algorytm predykcji przez częściowe dopasowanie, algorytm ważenia drzewa kontekstów, algorytm Burrowsa–Wheelera oraz algorytm Wolfa i Willemsa. Pod względem uzyskiwanych współczynników kompresji algorytmy te uznawane są obecnie za najlepsze wśród uniwersalnych algorytmów bezstratnej kompresji danych.

Algorytmy kompresji danych to dziedzina rozwijana już od ponad pół wieku. Pewne algorytmy nie zostaną już istotnie udoskonalone. Przykładem może być arytmetyczny koder entropijny, którego praktyczne implementacje mają efektywność kodowania na tyle bliską 100 %, że znacząca jej poprawa już nie jest możliwa — można oczekiwać opracowania koderów szybszych, prostszych w użyciu, ale nie większej efektywności kodowania. Poprawę współczynników kompresji osiąga się wyznaczając precyzyjniej dane wejściowe dla kodera entropijnego, czyli konstruując lepsze modele danych. Algorytmy predykcji przez częściowe dopasowanie, ważenia drzewa kontekstów oraz algorytm Wolfa i Willemsa od wcześniej znanych algorytmów odróżnia w istocie jedynie użyty model danych. Interesujący jest algorytm Wolfa i Willemsa, w którym analizuje się efekty zastosowania dwóch różnych modeli danych (na przykład, modelu algorytmu predykcji przez częściowe dopasowanie oraz modelu ważenia drzewa kontekstów). Algorytm Burrowsa–Wheelera jest odmienny od pozostałych algorytmów omówionych w niniejszym artykule, po części zbliżony do algorytmów predykcyjnych oraz transformacyjnych stosowanych w kompresji obrazów. Zamiast konstruowania złożonego modelu danych przekształca się te dane tak, aby lepsze współczynniki kompresji uzyskać stosując prostszy model danych. Algorytm ten uzyskuje średnio współczynniki kompresji nieznacznie gorsze niż pozostałe omówione algorytmy, ma jednak od nich istotnie mniejsze złożoności, a dla niektórych zbiorów danych uzyskuje lepsze współczynniki kompresji.

Współczynnik kompresji uzyskiwany przez algorytm nie jest jedynym kryterium jego oceny, dlatego też współistnieją różne algorytmy dostosowane do różnych wymagań. Spośród algorytmów omówionych w niniejszym artykule obecnie praktycznie stosowane są: algorytm predykcji przez częściowe dopasowanie oraz algorytm Burrowsa–Wheelera.

Podziękowanie. Artykuł niniejszy powstał w wyniku projektu badawczego nr BW-486/RAu-2/2002 zrealizowanego w roku 2002 w Instytucie Informatyki Politechniki Śląskiej.

LITERATURA

1. Arnavut Z., Magliveras S. S.: Block Sorting and Compression. Proceedings of the Data Compression Conference DCC'97, USA, March 1997, pp. 181-90.
2. Aberg J., Shtarkov Y. M.: Text Compression by Context Tree Weighting. Proceedings of the IEEE Data Compression Conference (DCC'97), USA 1997, pp. 377-86.
3. Balkenhol B., Kurtz S.: Universal Data Compression Based on the Burrows-Wheeler Transformation: Theory and Practice. IEEE Trans. on Computers, Oct. 2000, Vol.49(10).
4. Balkenhol B., Kurtz S., Shtarkov Y. M.: Modifications of the Burrows and Wheeler Data Compression Algorithm. Proceedings of the Data Compression Conference DCC'99, USA, March 1999, pp. 188-97.
5. Bentley J. L., Sleator D. D., Tarjan R. E., Wei V. K.: A Locally Adaptive Data Compression Scheme. Communications of the ACM, Apr. 1986, Vol. 29(4), pp. 320-30.
6. Bunton S.: Semantically Motivated Improvements for PPM Variants. The Computer Journal, Great Britain 1997, Vol. 40(2-3), pp. 76-93.
7. Burrows M., Wheeler D. J.: A block-sorting lossless data compression algorithm. SRC Research Report 124, Systems Research Center, Digital Equipment Corporation, Palo Alto, CA, USA, May 1994.
8. Bell T., Witten I. H., Cleary J. G.: Modelling for text compression. ACM Computing Surveys, Vol. 21(4), 1989, pp. 557–591.
9. Cleary J. G., Teahan W. J.: Experiments on the Zero Frequency Problem. Proceedings of the IEEE Data Compression Conference (DCC'95), USA, March 1995, pp. 480.
10. Cleary J. G., Teahan W. J.: Unbounded length contexts for PPM. Computer Journal, Great Britain 1997, Vol. 40(2-3), pp. 67-75.
11. Cleary J. G., Teahan W. J., Witten I. H.: Unbounded length contexts for PPM. Proc. of the Data Compression Conference (DCC'95), USA, March 1995, pp. 52-61.
12. Deorowicz S.: Improvements to Burrows-Wheeler compression algorithm. Software—Practice and Experience, Vol. 30(13). September 2000, pp. 1465-83.
13. Deorowicz S.: Second step algorithms in the Burrows-Wheeler compression algorithm. Software—Practice and Experience, Vol. 32(2), 2002, pp. 99-111.
14. Deorowicz S.: Universal lossless data compression algorithms. Rozprawa doktorska, Institute of Computer Science, Silesian University of Technology, Gliwice, 2003.
15. Drozdek A.: Wprowadzenie do kompresji danych. WNT, Warszawa 1999.
16. Effros M.: Universal Lossless Source Coding with the Burrows Wheeler Transform. Proceedings of the Data Compression Conference DCC'99, USA 1999, pp. 178-87.
17. Fenwick P.: Block Sorting Text Compression. Proceedings of the 19th Australian Computer Science Conference, Melbourne, Australia 1996, pp. 193-202.

18. Fenwick P.: The Burrows–Wheeler Transform for block sorting text compression: principles and improvements. The Computer Journal, 1996, Vol. 39(9), pp. 731-40.
19. Hoang D. T.: Fast and Efficient Algorithms for Text and Video Compression. Rozprawa doktorska, Department of Computer Science, Brown University, Providence, Rhode Island 02912, May 1997.
20. Howard P. G.: The design and analysis of efficient lossless data compression systems. Rozprawa doktorska, Department of Computer Science, Brown University, Providence, Rhode Island 02912, June 1993.
21. Moffat A.: Implementing the PPM data compression scheme. IEEE Transactions on Communications, 1990, Vol. 38(11), pp. 1917-21.
22. Sayood K.: Kompresja danych wprowadzenie. RM, Warszawa 2002.
23. Skarbek W.: Metody reprezentacji obrazów cyfrowych. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993.
24. Praca zbiorowa pod redakcją W. Skarbka: Multimedia algorytmy i standardy kompresji. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998.
25. Sadakane K., Okazaki T., Imai H.: Implementing the Context Tree Weighting Method for Text Compression. Proceedings of the IEEE Data Compression Conference (DCC'2000), USA 2000, pp. 123-32.
26. Starosolski R.: Algorytmy bezstratnej kompresji obrazów. Studia Informatica, Vol. 23, Nr 4(51), Gliwice 2002, ss. 277-300.
27. Starosolski R.: Algorytmy bezstratnej kompresji danych. Studia Informatica, Vol. 24, Nr 1(52), Gliwice 2003, ss. 137-58.
28. Teahan W. J., Cleary J. G.: The Entropy of English Using PPM-based Models. Proceedings of the IEEE Data Compression Conference (DCC'96), USA 1996, pp. 53-62.
29. Teahan W. J., Inglis S., Cleary J. G., Holmes G.: Correcting English text using PPM models. Proceedings of the IEEE Data Compression Conference (DCC'98), March-April 1998, pp. 289-98.
30. Tjalkens T. J., Volf P. A. J., Willems F. M. J.: A Context-Tree Weighting Method for Text Generating Sources. Proceedings of the IEEE Data Compression Conference (DCC'97), USA 1997, pp. 472.
31. Volf P. A. J., Willems F. M. J.: Switching Between Two Universal Source Coding Algorithms. Proceedings of the IEEE Data Compression Conference DCC'98, USA, March 1998, pp. 491-500.
32. Volf P. A. J., Willems F. M. J.: The Switching Method: Elaborations. Proceedings of the 19th Symposium on Information Theory in the Benelux, Veldhoven, The Netherlands, May 1998, pp. 13-20.

33. Witten I. H., Moffat A., Bell T. C.: Managing Gigabytes. Van Nostrand Reinhold, 1994.
34. Willems F. M. J., Shtarkov Y. M., Tjalkens T. J.: The Context Tree Weighting Method: Basic Properties. IEEE Trans. on Information Theory, May 1995, Vol. 41, pp.653-64.
35. Willems F., Shtarkov Y., Tjalkens T.: Reflections on "The Context-Tree Weighting Method: Basic Properties". IEEE Information Theory Society Newsletter, Vol. 47, March 1997, pp. 20-7.

Recenzent: Prof. dr hab. inż. Konrad Wojciechowski

Wpłynęło do Redakcji 11 grudnia 2002 r.

Abstract

In this paper we present a brief overview of four modern lossless data compression algorithms and present bibliography concerning those algorithms. We overview Prediction by Partial Matching and Context Tree Weighting — in terms of average compression ratio those algorithms are regarded as best among adaptive universal lossless data compression algorithms. We also describe the Switching Method of Volf and Willems that combines two different adaptive data models (like Prediction by Partial Matching and Context Tree Weighting) and achieves overall compression ratio better than achieved with each of the models used separately. We also present the Burrows–Wheeler Block Sorting Compression Algorithm. As compared to other presented algorithms this algorithm achieves a little worse average compression ratios, however for some files the compression ratios are best. It is also significantly faster than others. Compression ratio is not the only criterion that determines the practical applicability of the compression algorithm. For this reason there are many different algorithms that suit different needs of practical systems. From among described algorithms the Prediction by Partial Matching and the Burrows–Wheeler Block Sorting Compression Algorithm are used presently in practical systems.

Adres

Roman STAROSOLSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, rstaros@star.iinf.polsl.gliwice.pl .