Kodowanie słownikowe

Kodowanie i kompresja danych - Wykład 4

Maciek Gębala

18 marca 2020

Notatki

Motywacje

- Dane, które kompresujemy, bardzo często nie tworzą ciągu wartości niezależnych (kolejne symbole są zależne od poprzednich).
- Pewne ciągi (słowa) powtarzają się bardzo często.

Notatki

Słowniki statyczne

- Mamy ustalony stały słownik.
- Tekst kodujemy jako ciąg pozycji ze słownika.
- Co z elementami których nie ma w słowniku?
  Można np. umieścić pojedyncze litery w słowniku.
- Niestety słowniki statyczne są wrażliwe na zmianę charakteru danych.

Przykład

Kodowanie tekstów programów - statyczny słownik słów kluczowych.

Notatki

Kodowanie digramowe

- Ustalamy wielkość słownika np. 256 (2^8)
- Słownik składa się ze wszystkich liter i tylu par liter (digramów), ile się jeszcze zmieści (najbardziej prawdopodobne pary).
- Na przykład w kodzie ASCII używamy przeważnie tylko 95 znaków (znaki drukowalne) i pozostałe 161 możemy zastąpić najczęściej występującymi parami.

Notatki

Słowniki dynamiczne

- Słownik tworzymy w trakcie kodowania.
- Słownik dostosowuje się do charakteru danych.
- Dekoder może go odtworzyć w oparciu o odkodowaną część danych (nie trzeba go dołączać do danych).

Notatki

Ziv, Lempel, LZ77

Idea:
Słownikiem jest zakodowana/odkodowana część tekstu.

Dla zakodowanej części długości n i niezakodowanej długości m, czyli dla ciągu $x_1 \dots x_n x_{n+1} \dots x_{n+m}$ szukamy najdłuższego podsłowa w ciągu $x_1 \dots x_n$ będącego prefiksem ciągu $x_{n+1} \dots x_{n+m}$ (dopasowanie).

Jako kod podajemy trzy liczby:
- wielkość przesunięcia w lewo
- ilość kopiowanych znaków
- kod pierwszej niepasującej litery.

Jeśli pojawia się nowa litera której nie można znaleźć w zakodowanej części to wysyłamy trójkę $(0, 0, kod)$

Notatki

Przykład - kodowanie

- Ustalamy $n = 7$ i $m = 8$ $(2^k - 1$ i $2^k)$.
- wabba-wabba-wabba-woo-woo-woo (0,0,k(w))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(b))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(a))
- wabba-wabba-wabba-woo-woo-woo (0,0,k(-))
- wabba-wabba-wabba-woo-woo-woo (6,7,k(a))
- wabba-wabba-wabba-woo-woo-woo (6,5,k(o))
- wabba-wabba-wabba-woo-woo-woo (1,1,k(-))
- wabba-wabba-wabba-woo-woo-woo (4,6,k(o))

Notatki

Przykład - dekodowanie

- Ustalamy $n = 7$ i $m = 8$ $(2^k - 1$ i $2^k)$.
- (0,0,k(w)) w
- (0,0,k(a)) wa
- (0,0,k(b)) wab
- (1,1,k(a)) wabba
- (0,0,k(-)) wabba-
- (6,7,k(a)) wabba-wabba-wa
- (6,5,k(o)) wabba-wabba-wabba-wo
- (1,1,k(-)) wabba-wabba-wabba-woo-
- (4,6,k(o)) wabba-wabba-wabba-woo-woo-woo

Notatki

LZ77 - podsumowanie

- Mamy bufor słownika i bufor kodowania - razem nazywamy to oknem.
- **Co można usprawnić:** Zamiast 3 liczb wysyłać tylko dwie: kopiowany ciąg (bez kodu ostatniej litery) lub 0 i kod nowej litery.
- Krotki kodowane przy pomocy algorytmu Huffmana (adaptacyjne)
- Zastosowanie: zip, gzip, png, arj, rar, ...

Notatki

Ziv, Lempel, LZ78

- Powtórzenia nie muszą być na małej odległości - wada LZ77.
- Słownik to zbiór numerowanych słów.
- Zawartość słownika jest tworzona w oparciu o zakodowaną część tekstu.
- Kodowanie to ciąg indeksów ze słownika tworzący kodowany tekst.

Notatki

LZ78 - Algorytm

- Na początku słownik jest pusty (ewentualnie ma jeden wpis $0 - \varepsilon$).
- Szukamy w słowniku najdłuższego prefiksu tekstu i wysyłamy numer tego słowa oraz kod następnej litery. Do słownika dodajemy kolejne nowe słowo (to które wysłaliśmy). Jeśli w słowniku nie ma takiego prefiksu to wysyłamy 0 i kod pierwszej litery.
- Czasami na samym końcu wysyłamy krótszy prefiks i ostatnią literę.

Notatki

Przykład - kodowanie

wabba-wabba-wabba-woo-woo-woo

- (0,k(w)) 1 - w abba-wabba-wabba-woo-woo-woo
- (0,k(a)) 2 - a bba-wabba-wabba-woo-woo-woo
- (0,k(b)) 3 - b ba-wabba-wabba-woo-woo-woo
- (3,k(a)) 4 - ba -wabba-wabba-woo-woo-woo
- (0,k(-)) 5 - - wabba-wabba-woo-woo-woo
- (1,k(a)) 6 - wa bba-wabba-woo-woo-woo
- (3,k(b)) 7 - bb a-wabba-woo-woo-woo
- (2,k(-)) 8 - a- wabba-woo-woo-woo
- (6,k(b)) 9 - wab ba-woo-woo-woo
- (4,k(-)) 10 - ba- woo-woo-woo
- (1,k(o)) 11 - wo o-woo-woo
- (0,k(o)) 12 - o -woo-woo
- (5,k(w)) 13 - -w oo-woo
- (12,k(o)) 14 - oo -woo
- (13,k(o)) 15 - -wo o
- (0,k(o))

Notatki

Przykład - odkodowywanie

- (0,k(w)) 1 - w w
- (0,k(a)) 2 - a wa
- (0,k(b)) 3 - b wab
- (3,k(a)) 4 - ba wabba
- (0,k(-)) 5 - - wabba-
- (1,k(a)) 6 - wa wabba-wa
- (3,k(b)) 7 - bb wabba-wabb
- (2,k(-)) 8 - a- wabba-wabba-
- (6,k(b)) 9 - wab wabba-wabba-wab
- (4,k(-)) 10 - ba- wabba-wabba-wabba-
- (1,k(o)) 11 - wo wabba-wabba-wabba-wo
- (0,k(o)) 12 - o wabba-wabba-wabba-woo
- (5,k(w)) 13 - -w wabba-wabba-wabba-woo-w
- (12,k(o)) 14 - oo wabba-wabba-wabba-woo-woo
- (13,k(o)) 15 - -wo wabba-wabba-wabba-woo-woo-wo
- (0,k(o)) wabba-wabba-wabba-woo-woo-woo

Notatki

Ziv, Lempel, Welch - LZW

- Poprawki do LZ78 - rezygnacja z drugiego elementu pary.
- Potrzebny słownik początkowy zawierający wszystkie używane litery.
- Słownik konstruujemy tak samo jak w LZ78.

Notatki

Przykład - kodowanie (0 - a, 1 - b, 2 - o, 3 - w, 4 - -)

wabba-wabba-wabba-woo-woo-woo

- 3 5 - wa abba-wabba-wabba-woo-woo-woo
- 0 6 - ab bba-wabba-wabba-woo-woo-woo
- 1 7 - bb ba-wabba-wabba-woo-woo-woo
- 1 8 - ba a-wabba-wabba-woo-woo-woo
- 0 9 - a- -wabba-wabba-woo-woo-woo
- 4 10 - -w wabba-wabba-woo-woo-woo
- 5 11 - wab bba-wabba-woo-woo-woo
- 7 12 - bba a-wabba-woo-woo-woo
- 9 13 - a-w wabba-woo-woo-woo
- 11 14 - wabb ba-woo-woo-woo
- 8 15 - ba- -woo-woo-woo
- 10 16 - -wo oo-woo-woo
- 2 17 - oo o-woo-woo
- 2 18 - o- -woo-woo
- 16 19 - -woo o-woo
- 18 20 - o-w woo
- 3 21 - wo oo
- 17

Notatki

Przykład - dekodowanie

Słownik początkowy

0 - a, 1 - b, 2 - o, 3 - w, 4 - -

- 3 w 5 - w?
- 0 wa 5 - wa, 6 - a?
- 1 wab 6 - ab, 7 - b?
- 1 wabb 7 - bb, 8 - b?
- 0 wabba 8 - ba, 9 - a?
- 4 wabba- 9 - a-, 10 - -?
- 5 wabba-wa 10 - -w, 11 - wa?
- 7 wabba-wabb 11 - wab, 12 - bb?
- 9 wabba-wabba- 12 - bba, 13 - a-?
- 11 wabba-wabba-wab 13 - a-w, 14 - wab?
- 8 wabba-wabba-wabba 14 - wabb, 15 - ba?
- 10 wabba-wabba-wabba-w 15 - ba-, 16 - -w?
- ...

Notatki

Przykład - trudny przypadek

Słowo abababababab... i słownik 1 - a i 2 - b.

Kodowanie

1 3 - ab, 2 4 - ba, 3 5 - aba, 5 6 - abab, ...

Dekodowanie

- 1 a 3 - a?
- 2 ab 3 - ab, 4 - b?
- 3 abab 4 - ba, 5 - ab?
- 5 ababab?

Ten problem można rozwiązać przez porównanie z algorytmem kodowania i wywnioskować, że 5 - aba.

Notatki

Polecenie compress

- Jedno z pierwszych zastosowań algorytmu LZW.
- Rozpoczynamy ze słownikiem o rozmiarze 512 elementów (9 bitów).
- Podwajamy (i zwiększamy o jeden bit indeks) jak się zapełni.
- Kontynuujemy aż osiągniemy zadany rozmiar (zwykle 2^{16}) słownika.
- Monitorujemy poziom kompresji, jeżeli spada, to czyścimy słownik i rozpoczynamy z początkowym słownikiem.
- Algorytm jest szybki, ale oferuje niski poziom kompresji.

Notatki

Algorytm GIF

- Oparty o algorytm LZW, opracowany przez *Compuserve Information Service*.
- Pierwszy bajt oznacza minimalną liczbę bitów na piksel b (np. 8).
- Liczba (indeks) 2^b jest zarezerwowana jako „kod oczyszczający” (przywraca wartości początkowe).
- Słownik w razie potrzeby jest podwajany aż do wartosci 4096.

Zastosowanie i ograniczenia

- Stosowany do kompresji „sztucznych” obrazów.
- Niespecjalnie nadaje się do zdjęć (kodowanie arytmetyczne różnic pikseli daje zwykle lepsze efekty).
- Kolory są „indeksowane” (np. 255 możliwych kolorów).
- Algorytm kodowania opatentowany (dekodowania – nie).

Notatki

Standard V.42bis

- Do kompresji (i korekcji błędów) danych w sieciach telefonicznych.
- Może działać w dwóch trybach: przezroczystym (bez kompresji) i z kompresją algorytmem LZW.
- Standard sugeruje (choć brak specyfikacji sposobu etc.) testowanie poziomu kompresji w trakcie działania.

Notatki

Standard V.42bis – algorytm kompresji

Początkowy rozmiar słownika – negocjowany (min 512, sugerowany 2048).

Kody specjalne

0. ETM (przejście do trybu przezroczystego)
1. FLUSH („oczyszczanie" danych (?))
2. STEPUP (zwiększenie rozmiaru słownika $\times 2$)

Gdy liczba elementów w słowniku (trzy kody sterujące + słowa/znaki) przekroczy wartość zmiennej C_3, wysyłany jest kod STEPUP, słownik jest podwajany a indeks zwiększany o jeden bit.

Notatki

Standard V.42bis – algorytm kompresji (2)

Gdy słownik jest wypełniony rozpoczynana jest procedura „oczyszczania":

- w zmiennej N_5 pamiętane jest położenie pierwszego elementu będącego napisem (tzn. spoza pierwotnego słownika),
- począwszy od N_5 zwiększamy licznik C_1 szukając elementu, który nie jest prefiksem żadnego innego elementu słownika (ten algorytm znajduje najstarszy element, który nie pojawił się od chwili wpisania do słownika).

Aby uprościć algorytm, nie wolno używać ostatniego elementu słownika.

W celu zredukowania skutków ew. błędów, długość pojedynczego napisu ogranicza się (domyślnie 6, maksymalnie 250 znaków).

Notatki

Notatki

Notatki