Kanoniczne kody Huffmana

Szymon Mikler, zad. 2.3

Kanoniczne kodowanie Huffmana to efektywna metoda implementacji kodów Huffmana. Zapewnia ona szybsze dekodowanie tekstu i spośród wszystkich możliwych kodów Huffmana (dla zadanych prawdopodobieństw) procedura tworzy kody najbardziej "zrównoważone", tj. o najmniejszej wariancji długości słów kodowych. Opiszę najprostszy wariant algorytmu, tzn. wariant używający tablic do nadawania słów kodowych i dekodowania.

Kodowanie

Tworzenie drzewa zaczyna się od tworzenia normalnego drzewa Huffmana i jego normalizacji. Nadawać słowa kodowe możemy używając drzewa lub tablicy l, jak opisano poniżej. Druga metoda jest ciekawsza, bo nie wymaga przechowywania całego drzewa, a jedynie tablicy o długości $\log n$.

Tworzenie drzewa odbywa się tak jak w standardowym drzewie Huffmana, ale remisy rozstrzygane są przez preferowanie drzew powstałych wcześniej, tzn. we wcześniejszym kroku[1]. Przyjmujemy, że drzewa jednowierzchołkowe powstały równocześnie, więc jedynie w ich przypadku może dojść do remisu – nawet wtedy kształt drzewa pozostanie jednoznaczny.

Podczas procedury normalizacji zamieniamy kolejność wierzchołków tak, aby wszystkie liście znajdowały się po prawej stronie. Procedurę tę pominę, bo drzewo po normalizacji można (jednoznacznie!) opisać tablicą l taką, że l_i to ilość liści na wysokości i. Opiszę więc, jak nadać słowa kodowe korzystając z tablicy l. Zaczynamy ze słowem kodowym 0. Dla l_1, l_2, ..., l_n:

1. Jeśli $l_i = 0$ do aktualnego słowa kodowego dopisujemy 0,
2. Jeśli $l_i \neq 0$ to nadajemy aktualne słowo kodowe elementowi o najmniejszym prawdopodobieństwie (i bez nadanego słowa) na poziomie i. Do aktualnego słowa kodowego binarnie dodajemy 1 i powtarzamy krok 2 dopóki wszystkie słowa na poziomie i nie mają nadanych słów kodowych.

Stworzymy też tablicę g, której elemet g_i to największe słowo kodowe na poziomie i lub 0 gdy nie ma liści na poziomie i. Tablicę g można łatwo stworzyć podczas tworzenia l. Do dekodowania nie potrzebujemy już drzewa.

Dekodowanie

Wykorzystamy tablicę g. Odczytujemy słowo kodowe bit po bicie i aktualnie wczytane słowo długości n porównujemy z g_n. Jeśli g_n jest mniejsze (jako liczba w systemie binarnym), to doczytujemy kolejny bit do aktualnego słowa. W przeciwnym wypadku dekodujemy aktualnie wczytane słowo.

Zalety

- Uzyskujemy drzewo Huffmana o minimalnej wysokości,
- Drzewo jest jednoznaczne dla zadanych prawdopodobieństw,
- Do nadawania słów kodowych wystarczy tablica l, nie potrzeba całego drzewa,
- Do dekodowania wystarczy nam tablica g. Używamy więc pamięć proporcjonalną do wysokości drzewa,
- Dekodowanie korzystające z tablicy jest w implementacji szybsze niż dekodowanie poruszając się po drzewie, bo zapewnia sekwencyjny dostęp do pamięci.

[1] Podczas tworzenia drzewa Huffmana w każdym kroku wybieramy dwa drzewa o najmniejszych prawdopodobieństwach p_1, p_2 i zastępujemy je jednym, o prawdopodobieństwie $p_1 + p_2$. W kanonicznych kodach Huffmana, będziemy dodatkowo zapisywać, w którym kroku drzewo było ostatnio łączone.