TIMKoD – Lab 5 – Kompresja bezstratna – Kodowanie Huffmana

Opis pliku z zadaniami

Wszystkie zadania na zajęciach będą przekazywane w postaci plików `.pdf`, sformatowanych podobnie do tego dokumentu. Zadania będą różnego rodzaju. Za każdym razem będą one odpowiednio oznaczone:

- Zadania do wykonania na zajęciach oznaczone są symbolem $\triangle$ – nie są one punktowane, ale należy je wykonać w czasie zajęć.
- Punktowane zadania do wykonania na zajęciach oznaczone są symbolem $\diamond$ – należy je wykonać na zajęciach i zaprezentować prowadzącemu, w wypadku nie wykonania zadania w czasie zajęć lub nieobecności, zadania staje się zadaniem do wykonania w domu.
- Zadania programistyczne można wykonywać w dowolnym języku programowania, używając jedynie biblioteki standardowej dostępnej dla tego języka.

Cel zajęć

Na poprzednich zajęciach przygotowaliśmy framework do kompresji i dekompresji używając prostego kodowania symbol po symbolu z kodami o stałej długość. Dzisiaj porównamy tą metodę z kodowaniem Huffmana.

Przygotowanie do zajęć

- Do wykonania zadań potrzebny będzie korpus tekstowy, który można pobrać z odpowiedniej sekcji w systemie e-kursy.
- Teksty są znormalizowane, zawierają jedynie 26 małych liter alfabetu łacińskiego, cyfry i spacje (czyli w sumie 37 znaków).

Podstawowe twierdzenie Shannona (Shannon's source coding theorem)

> Dane jest źródło o entropii H (bitów na symbol) i kanał o przepustowości C (bitów na sekundę). Możliwe jest znalezienie takiego kodu, przypisującego transmitowanym symbolom różne ciągi bitowe, żeby prędkość transmisji wynosiła $\frac{C}{H} - \epsilon$ dla dowolnie małego ϵ. Ponadto nie można uzyskać średniej transmisji szybszej niż $\frac{C}{H}$ (symboli na sekundę).

Nieformalnie dotyczy ono ograniczenia na minimalną średnią długość słów kodowych (wartością oczekiwaną) w kodowaniu utworzonym do zapisu symboli generowanych przez pewne dyskretne źródło danych (dyskretną zmienną losową) o określonej entropii (średniej liczbie bitów na symbol). Źródło takie nie może zostać zakodowane kodem, który charakteryzowałem by się krótszą niż entropia źródła średnią długością słów kodowych, bez utraty informacji. Innymi słowy jeśli źródło informacji zostało zakodowane kodem, którego średnia długość jest mniejsza od entropi źródła możemy być pewni, że informacja została stracona.

1 Efektywność kodowania △

Treść

Dla dowolnego kodowania można obliczyć efektywność kodowania (efficiency):

$$\text{Eff} = H/L,$$

gdzie H to entropia a L to średnia długość słów kodowych (wartością oczekiwana). Najlepsze kodowanie charakteryzuje się efektywnością 100%.

Oblicz efektywność kodowania z ostatnich zajęć.

2 Kodowanie Huffmana *10pt*◇

Treść

Używając szablonu programu do kompresji i dekompresji stworzonego na poprzednich zajęciach zaimplementuj w nim kodowania Huffmana.

Policz średnią długość kodu i oblicz efektywność tego kodowania. Stwórz program służący do zapisu i odczytu danych tekstowych używając kodowania Huffmana.

Ciekawostka

Kodowania Huffman zostało wymyślone w 1951 przez Davida Huffmana podczas swoich studiów na MIT. Na zaliczenie u Profesora Robert Fano z teorii informacji, można było wybrać albo pisanie egzaminu albo pracy. Huffman dostał zadanie napisania pracy, na temat efektywnego kodowania w kodzie binarnym. Był bliski poddania się i zaczęcia uczenia się do egzaminu, kiedy wpadł na pomysł użycia posortowanego po częstościach wystąpień binarnego drzewa. Następnie szybko udało mu się udowodnić jego efektywność.

Tym samym Huffman wymyślił lepsze kodowanie niż jego Profesor, który razem z Shannonem pracował wtedy nad podobnym algorytmem znanym dzisiaj jako kodowanie Shannona-Fano.

Źrodła

- https://en.wikipedia.org/wiki/Shannon%27s_source_coding_theorem
- https://en.wikipedia.org/wiki/Shannon%E2%80%93Fano_coding
- https://en.wikipedia.org/wiki/Huffman_coding
- http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf
- http://www.inference.org.uk/itprnn/book.pdf