Przetwarzanie i transmisja danych multimedialnych

Wykład 2

Podstawy kompresji

Przemysław Sękalski

sekalski@dmcs.pl

Politechnika Łódzka
Katedra Mikroelektroniki i Technik Informatycznych
DMCS

Zawartość wykładu

1. Wprowadzenie do kompresji i transmisji danych
2. **Podstawy kompresji**
3. Kodowanie Shannona–Fano i Huffmana
4. Kodowanie arytmetyczne
5. Algorytmy słownikowe
6. Algorytm predykcji przez częściowe dopasowanie (PPM)
7. Transformata Burrowsa–Wheelera (BWT)
8. Wybrane algorytmy specjalizowane
9. Dynamiczny koder Markowa (DMC) i algorytm kontekstowych drzew ważonych (CTW)
10. Bezstratna kompresja obrazów
11. Stratna kompresja obrazów
12. Stratna kompresja dźwięku
13. Kompresja wideo

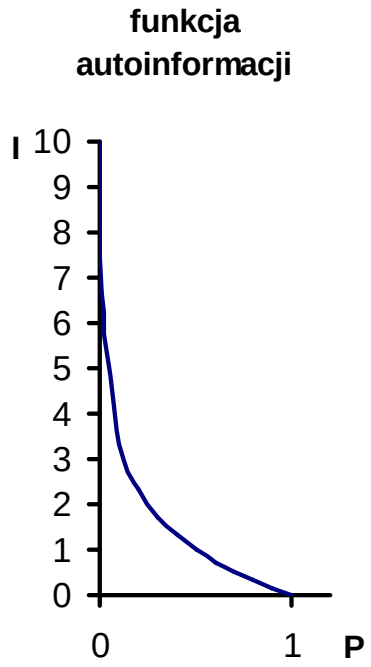
Plan wykładu

- Podstawowe definicje
  - Autoinformacja
  - Entropia
- Paradygmat kompresji
  - Modelowanie
  - Kodowanie
- Przykłady prostych kodów
  - Kody statyczne
  - Kody semi-statyczne

Autoinformacja

Autoinformacja (informacja własna) związana z wystąpieniem zdarzenia *A*, którego prawdopodobieństwo wystąpienia wynosi P(A) określana jest zależnością (Hartley 1928):

$$I(A) = \log_n \frac{1}{P(A)} = -\log_n P(A)$$



Jednostka, w której mierzona jest autoinformacja zależy od podstawy logarytmu:

- → n = 2 — bit (ang. bit [**bi**nary digi**t**]),
- → n = e — nat (ang. nat [**na**tural digi**t**]),
- → n = 10 — hartley (od nazwiska Ralpha Hartleya)

Obliczanie autoinformacji

Przykład 1:

Załóżmy, że mamy źródło które generuje symbole A = {a_1, a_2, a_3, a_4} z prawdopodobieństwem P = {½, ¼, 1/8,1/8}.

Autoinformacja wynosi wówczas:

$$I(a_1) = \log_2 \frac{1}{P(a_1)} = -\log_2 \frac{1}{2} = 1bit$$

$$I(a_2) = -\log_2 \frac{1}{P(a_2)} = -\log_2 \frac{1}{4} = 2bity$$

$$I(a_3) = I(a_4) = -\log_2 \frac{1}{P(a_3)} = -\log_2 \frac{1}{8} = 3bity$$

Obliczanie autoinformacji

Przykład 2:

Załóżmy, że mamy źródło które generuje symbole A = {a_1, a_2, a_3, a_4} z prawdopodobieństwem P = {3/8, 3/8, 1/8,1/8}.

Autoinformacja wynosi wówczas:

$$I(a_1) = I(a_2) = \log_2 \frac{1}{P(a_1)} = -\log_2 \frac{8}{3} = 1{,}415 bita$$

$$I(a_3) = I(a_4) = -\log_2 \frac{1}{P(a_3)} = -\log_2 \frac{1}{8} = 3 bity$$

Funkcja autoinformacji koncentruje się na jednej literze,
a nie na zbiorze liter (wiadomości)

Entropia

Pojęcie **entropii** wprowadził Shannon.

Jest to funkcja postaci:

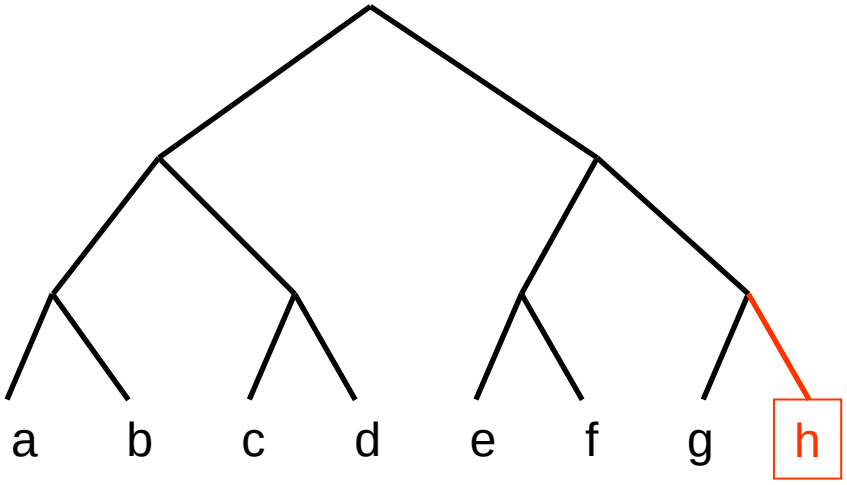
$$H(p_1,\ldots,p_n) = H(A) = -\sum_{i=1}^{n} p_i \lg p_i = \sum_{i=1}^{n} p_i I_i$$

Taką funkcję nazywamy entropią stowarzyszona
ze zbiorem *n* niezależnych zdarzeń A = $\{a_1, \ldots, a_n\}$
i ze zbiorem prawdopodobieństw ich zajścia P = $\{p_1, \ldots, p_n)\}$

Entropia jest średnią autoinformacją

Entropia

W kontekście kodowania wiadomości entropia reprezentuje ograniczenie na średnią długość słowa kodu używanego przy kodowaniu wartości wyjściowych.



Przykład drzewa decyzyjnego

$$p_i = \frac{1}{7}, I_i \approx 2{,}8bit$$

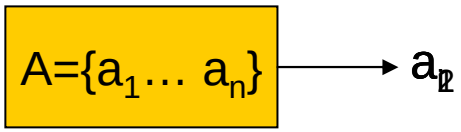
$$H(A) = \sum_{i=1}^{n} p_i I_i = 7 * \frac{1}{7} * 2{,}8 = 2{,}8bit$$

$$p_i = \frac{1}{8}, I_i = 3bity$$

$$H(A) = \sum_{i=1}^{n} p_i I_i = 8 * \frac{1}{8} * 3 = 3bity$$

Eksperyment i jego entropia

Eksperymentem nazywamy generowanie przez źródło symboli a_i ze zbioru alfabetu A



Entropią eksperymentu nazywamy miarę określającą średnią liczbę symboli binarnych potrzebnych do zakodowania ciągu utworzonego z symboli kolejno wygenerowanych przez źródło.

Najlepszym wynikiem jaki można uzyskać stosując kompresje bezstratna jest zakodowanie sekwencji symboli tak, aby średnia liczba bitów przypadająca na symbol była równa **entropii źródła**.

Shannon

Entropia źródła

Z definicji entropia wynosi:

$$H(p_1,\ldots,p_n) = H(A) = -\sum_{i=1}^{n} p_i \lg p_i$$

Entropia źródła S generującego ciąg symboli x_1, x_2,… x_n należących do wspólnego alfabetu A = {1, 2, 3, … ,m} wynosi:

$$H(S) = -\lim_{n\to\infty} \frac{1}{n} \left(\sum_{i_1=1}^{m} \sum_{i_2=1}^{m} \cdots \sum_{i_n=1}^{m} p_{(x_1=i_1, x_2=i_2, \ldots, x_n=i_n)} \log p_{(x_1=i_1, x_2=i_2, \ldots, x_n=i_n)} \right)$$

Przypadki szczególne …

Przy założeniu, że wszystkie elementy serii mają rozkład identyczny i niezależny od siebie (równe prawdopodobieństwa wystąpienia zdarzeń $\Leftrightarrow$ $p_i = \text{const} = 1/n \quad \forall\, i \in \langle 1,n \rangle$), otrzymujemy:

$$H(S) = -\sum_{i_1=1}^{m} p_{(x_1=i_1)} \log p_{(x_1=i_1)}$$

Jest to entropia źródła pierwszego rzędu

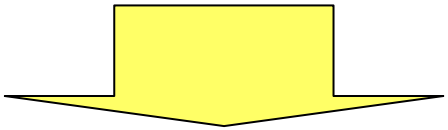
Znajomo wyglądający wzór ??!!

… a jak jest w rzeczywistości ?

W rzeczywistych źródłach symbole nie są generowane z jednakowym prawdopodobieństwem

W większości przypadków symbole są zależne od siebie.

Nie znany jest także rozkład wszystkich próbek $n \to \infty$, a tylko ograniczona ich ilość



Entropia pierwszego rzędu nie jest dobrą miarą entropii źródła

Entropia fizycznego źródła nie jest znana

Policzmy entropię pierwszego rzędu

Dla przykładowej sekwencji danych:

W= {1 2 3 2 3 4 5 6 7 8 9 2 3 3 5 6}

$$P(3) = \frac{4}{16} \qquad \log P(3) = -2$$

$$P(2) = \frac{3}{16} \qquad \log P(2) = -2{,}41$$

$$P(5) = P(6) = \frac{2}{16} \qquad \log P(5{,}6) = -3$$

$$P(1) = P(4) = P(7) = P(8) = P(9) = \frac{1}{16} \qquad \log P(1{,}4{,}7{,}8{,}9) = -4$$

$$H(W) = -\sum_{i=1}^{9} p_{(i)} \log_2 p_{(i)} \approx 2{,}95 bit$$

Entropia pierwszego rzędu

Pomiędzy poszczególnymi symbolami istnieje zależność (korelacja). Zastępując kolejne elementy wiadomości W różnicą pomiędzy dwoma kolejnymi symbolami otrzymujemy nowy ciąg:

W= {1 2 3 2 3 4 5 6 7 8 9 2 3 3 5 6}

W_2={1 1 1 -1 1 1 1 1 1 1 1 -7 1 0 2 1}

$$P(1) = \frac{12}{16} \qquad \log P(1) = -0{,}41$$

$$P(-7) = P(-1) = P(0) = P(2) = P(2) = \frac{1}{16} \qquad \log P(-7,-1,0,2) = -4$$

$$H(W_2) = -\sum_{i=-7}^{2} p_{(i)} \log_2 p_{(i)} \approx 1{,}31 bit$$

Plan wykładu

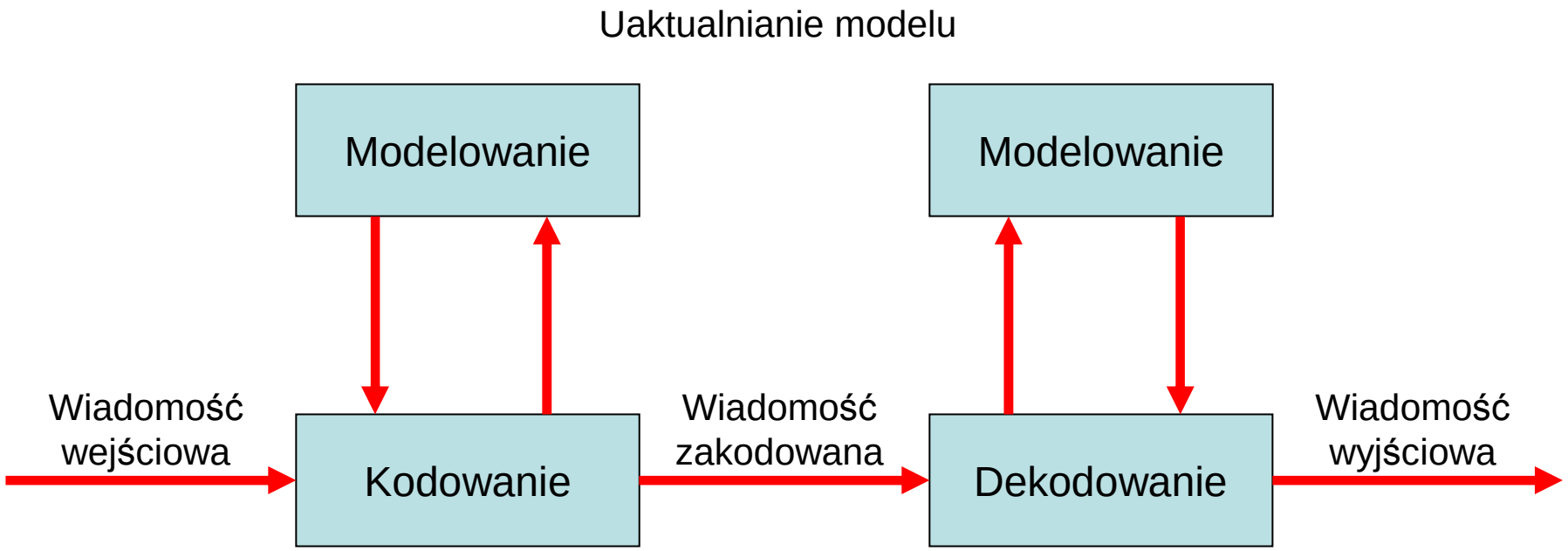
- Podstawowe definicje
  - Autoinformacja
  - Entropia
- Paradygmat kompresji
  - Modelowanie
  - Kodowanie
- Przykłady prostych kodów
  - Kody statyczne
  - Kody semi-statyczne

Model

- W zależności jak przedstawimy informację otrzymamy różną wartość entropii.
- Znając strukturę wiadomości można zmieniać (najlepiej zmniejszać) entropię.
- Znając zależności pomiędzy symbolami (korelację) można budować model.

Modelem nazywamy zależności pomiędzy kolejnymi symbolami ciągu

Modelowanie



Współczesny paradygmat kompresji

Paradygmat kompresji

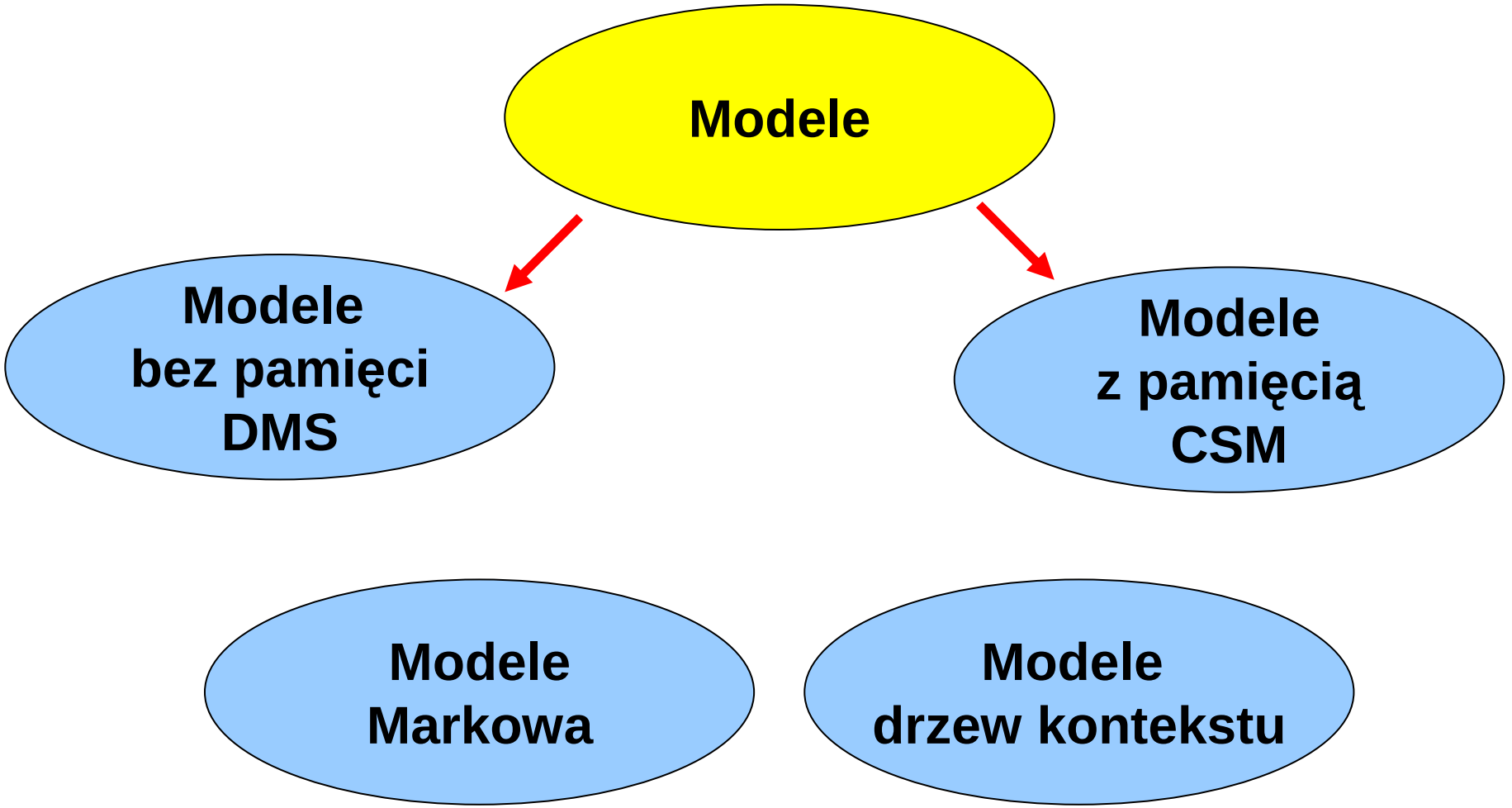
1. **Modelowanie:**

- Pierwszy etap kompresji podczas, którego badane są zależności pomiędzy symbolami i budowany jest model.
- Informacja nadmiarowa (redundantna) jest usuwana.
- Istnieje wiele metod budowania modelu.
- Niezbędna jest wiedza o źródle sygnału.

2. **Kodowanie:**

- Drugi etap kompresji, podczas którego kodowany jest sam model oraz informacja o odchyleniu wiadomości od modelu.
- Jest wiele rodzajów kodowania
- Metody kodowania oparte są na matematyce

Modelowanie



Model bez pamięci - DMS

- Najprostszą postacią źródła informacji jest dyskretne źródło bez pamięci - model DMS (discrete memoryless source), w którym emitowane symbole są od siebie niezależne.
- Model DMS często zwany jest modelem fizycznym
- Może, ale nie musi dokładnie opisywać źródło.
- Nie wnosi żadnej informacji o źródle i nie może być wykorzystany do skutecznego kodowania.
- **Nie jest wykorzystywany w kompresji danych**

Model z pamięcią - CSM

- Model źródła z pamięcią – zwany też modelem warunkowym (conditional source model) uwzględnia zależność pomiędzy wystąpieniem danego symbolu od wystąpienia wcześniejszych – kontekstu czasowego.
- Kontekst może być ograniczony do jednego lub kilku symboli
- Pełny kontekst to wszystkie dotychczas wyemitowane symbole
- Znając kontekst w danej chwili można wnioskować o prawdopodobieństwie wystąpienia kolejnego symbolu $P(\cdot \mid \mathbf{s}^t)$, gdzie $\mathbf{s}^t = (s_1, s_2, \ldots, s_t)$ jest sekwencją t symboli wygenerowanych w przeszłości (kontekstem rzędu t)
- Prawdopodobieństwo warunkowe $P(a_i \mid b_j)$ dla i = 1… n oraz j = 1… k obliczane jest z wzoru:

$$P\left(a_i \middle| b_j\right) = \frac{N\left(a_i, b_j\right)}{N\left(b_j\right)} = \frac{\text{Liczba wystąpień symbolu } a_i \text{ i kontekstu } b_j}{\text{Liczba wystąpień kontekstu } b_j}$$

Model CSM - podsumowanie

Model źródła jest określony przez:

- Alfabet symboli źródła $A_s=\{a_1, a_2, \ldots, a_n\}$
- Zbiór kontekstów *C* dla źródła *S* opisanego wzorem:

$$A_C^S = \{b_1, b_2, ..., b_k\}$$

- Prawdopodobieństwa warunkowe:

$$P\left(a_i \middle| b_j\right) = \frac{N\left(a_i, b_j\right)}{N\left(b_j\right)} = \frac{\text{Liczba wystąpień symbolu } a_i \text{ i kontekstu } b_j}{\text{Liczba wystąpień kontekstu } b_j}$$

dla i = 1… n (liczba symboli w alfabecie) oraz j = 1… k (liczba kontekstów)

- Zasadę określania kontekstu *C* w każdej chwili czasowej *t* jako funkcję *f(·)* symboli wcześniej wyemitowanych przez źródło. Funkcja *f(·)* przekształca wszystkie możliwe sekwencje symboli z A_s o długości *t* lub mniejszej w A_C^S

Model źródła Markowa

- Model źródła CSM oraz DMS jest szczególnym przypadkiem modelu źródła zaproponowany przez Markowa (1906).
- Dla źródła Markowa rzędu m kontekst $C^{(m)}$ jest ograniczony do m znaków poprzedzających generowany symbol s_l.
- Prawdopodobieństwo wystąpienia symbolu a_i z alfabetu A zależy jedynie od m symboli, jakie pojawiły się bezpośrednio przed nim. Określane jest jako prawdopodobieństwo warunkowe:

$$\forall_{i\in\langle 1,n\rangle} \; \forall_{j_1,j_2,\ldots,j_m\in\langle 1,n\rangle} P\left(a_i \middle| \mathbf{b_j}\right) = P\left(a_i \middle| a_{j_1}, a_{j_2}, \ldots, a_{j_m}\right)$$

Modele źródła Markowa

- Źródła Markowa często reprezentowane są za pomocą diagramu stanów.
- Dla źródła Markowa pierwszego rzędu jest tyle stanów ile symboli w alfabecie A (*n*).
- Dla źródła Markowa *m*-tego rzędu jest n^m stanów.
- Wyższe rzędy są z reguły wykorzystywane do kompresji obrazów (2D i 3D)
- Entropia źródła Markowa określona jest jako:

$$H(S|C^{(m)}) = -\sum_{j_1=1}^{n}\sum_{j_2=1}^{n}\dots\sum_{j_1=m}^{n}\sum_{i=1}^{n} P\left(a_{j_1}, a_{j_2}, ..., a_{j_m}, a_i\right)\log_2 P\left(a_i \middle| a_{j_1}, a_{j_2}, ..., a_{j_m}\right)$$

- Wartość entropii źródła Markowa mieści się pomiędzy entropią źródła

$$H(S_{łącznej}) \le H(S|C^{(m)}) \le H(S_{DMS})$$

Modelowanie - podsumowanie

- Istnieje wiele rodzajów modeli źródeł
- Do modelowania niezbędna jest znajomość źródła
- Im więcej informacji zależnych od siebie tym lepsza możliwość kompresji,
- Dla tekstu angielskiego w 1951 roku Shannon oszacował entropię przy założeniu kontekstu długości dwóch znaków na 3,1 bit/znak
- Współczesne modele szacują entropię tekstu angielskiego na 1,45 bit/znak

Plan wykładu

- Podstawowe definicje
  - Autoinformacja
  - Entropia
- Paradygmat kompresji
  - Modelowanie
  - Kodowanie
- Przykłady prostych kodów
  - Kody statyczne
  - Kody semi-statyczne

Kodowanie

Założenia:

- Źródło *S* emitujące sygnały a_i z alfabetu $A=\{a_1,\ldots,a_n\}$
- Znane jest prawdopodobieństwo wystąpienia symbolu a_i, oznaczone jako $p_i = p(a_i)$. Zbiór wartości wszystkich prawdopodobieństw określany będzie jako $P(p_1,\ldots,p_i)$.
- Symbolom a_i odpowiadają słowa kodu, które należą do zbioru $K=\{k_1,\ldots,k_n\}$

Kodowanie - definicje

- Kodem nazywamy odwzorowanie alfabetu A na zbiór słów kodowych K ($f : A \rightarrow K$)
- Inaczej mówiąc: każde słowo kodu k_i musi opisywać symbol z alfabetu a_i
- Celem kompresji jest zmniejszanie średniego kosztu, danego wzorem:

$$L_{avg} = \sum_{1}^{n} p_i l_i$$

gdzie $i \in \langle 1,n \rangle$,
l_i długość słowa kodowego k_i,
p_i prawdopodobieństwo wystąpienia słowa k_i

Efektywność kompresji

Efektywność kompresji wyrażona w procentach mierzona jest jako stosunek entropii do średniego kosztu:

$$\frac{H}{L_{avg}} 100\%$$

Definicje

- Kod jednoznacznie dekodowalny to taki kod, który każdemu symbolowi a_i z alfabetu A przypisuje jedno i tylko jedno słowo kodowe k_i różne dla różnych symboli.
- Kodowanie jest przekształceniem różnowartościowym „jeden w jeden”. Matematyczne określenie to bijekcja.
- Przykładem kodu jednoznacznie dekodowalnego jest kod dwójkowy o stałej długości słowa.
- Który kod jest JD, a który nie ??:

$AK_1 = \{1, 01, 001, 0001\}$	jest JD
$AK_2 = \{0, 10, 110, 111\}$	jest JD
$AK_3 = \{0, 10, 11, 111\}$	Nie jest JD bo 11 10 = 111 0
$AK_4 = \{0, 01, 11\}$	Jest JD
$AK_5 = \{001, 1, 100\}$	Nie jest JD bo 1 001 = 100 1

Kody przedrostkowe

- Kod nazywamy przedrostkowym (prefix code) jeśli nie możemy otrzymać żadnego słowa kodu z innego słowa kodu przez dodanie do niego zer lub jedynek (żadne słowo kodu nie jest przedrostkiem innego słowa kodu).
- Kody przedrostkowe są łatwe do dekodowania (natychmiastowe dekodowanie)
- W literaturze anglosaskiej spotykane są także nazwy prefix condition codes, prefix-free codes, comma-free code.
- Przedrostkowość kodu jest warunkiem wystarczającym jego jednoznacznej dekodowalności (ale nie koniecznym)
- Przykład: AK4 = {0, 01, 11}, C ($\varsigma_1\varsigma_2\ \varsigma_3\ \varsigma_1$) = 0 01 11 0
  jak to zdekodować 001110 ?

Nierówność Krafta-MacMillana

Dla kodów jednoznacznie dekodowalnych zawierających n słów kodowych o długości l_i zachodzi nierówność:

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1$$

Równanie może służyć do określania czy kod jest jednoznacznie dekodowalny

Kod statyczny

Kod statyczny nie korzysta z informacji o prawdopodobieństwie wystąpienia symbolu w kodowanej sekwencji

Zalety:
- prosta budowa, szybkość
- brak konieczności przesyłania informacji do dekodera o modelu

Wady:
- niski współczynnik kompresji
- możliwa ekspansja danych po zakodowaniu (model jest niezmienny i może być nieadekwatny do danych kompresowanych)

Kod unarny

Kod α-Eliasa (kod unarny) – każdy z *x* symboli reprezentowany jest jako *x-1* symboli „1” po których następuje „0”

symbol	kod
1	0
2	10
3	110
4	1110
5	11110
6	111110
7	1111110
8	11111110
9	111111110

Cechy:
- prosty, ale bardzo długi (możliwa ekspansja)
- długość słowa zależna wprost od ilości symboli x,
- część składowa innych kodów,
- stosowany, gdy jeden symbol występuje znacznie częściej niż inne

Kod binarny

Kod β-Eliasa (kod binarny) – każdy z *x* symboli reprezentowany jest jako *liczba binarna z pominięciem zer wiodących*

symbol	kod
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Cechy:
- prosty,
- nie jest jednoznacznie dekodowalny,
- brak informacji o początku i końcu danego słowa
- długość słowa zależna od ilości symboli x, $(\lfloor \log_2 x \rfloor + 1)$
- część składowa innych kodów,

Kod γ-Eliasa

- Kod γ-Eliasa jest kodem złożonym
- Pierwsza część kodu kodowana według schematu α określa ilość kodu binarnego
- Druga część kodu określa symbol i jest kodowana według schematu β bez wiodącego bitu „1”

Cechy:
- prosta budowa,
- długość słowa opisana wzorem $2\lfloor \log_2 x \rfloor + 1$

symbol	kod
1	0
2	100
3	101
4	11000
5	11001
6	11010
7	11011
8	1110000
9	1110001

Reprezentacja Zeckendorfa

- Liczby Fibonacciego powstają poprzez zsumowanie dwóch poprzednich liczb przy czym pierwsze dwie są równe 1:

$$F_n=F_{n-1}-F_{n-2}$$

$$F = \{1, 2, 3, 5, 8, 13, 21,\ldots\}$$

Własność: Każdą liczbę całkowitą można przedstawić jako sumę liczb Fibonacciego

- Reprezentacja Zeckendorfa to zapis binarny liczby, w którym każdy bit odpowiada jednej liczbie Fibonacciego.
  Wartość bitu równa 1 oznacza, że dana liczba Fibonacciego wchodzi do sumy. Pomijana jest w reprezentacji liczba F_1

10 = 0x1 + 1x2 + 0x3 + 0x5 + 1x8 => 01001 lub 10010

11 = 1x1 + 1x2 + 0x3 + 0x5 + 1x8 => 11001 lub 10011

odwrócona prosta

Reprezentacja Zeckendorfa

- Skoro każdą liczbę można przedstawić jako sumę poprzednich liczb Fibonacciego zatem można przekształcić tak reprezentację Zeckendorfa, aby nie występowały po sobie kolejne jedynki.

$$11 = 1x1 + 1x2 + 0x3 + 0x5 + 1x8 \Rightarrow 11001 \rightarrow 10011$$
$$11 = 0x1 + 0x2 + 1x3 + 0x5 + 1x8 \Rightarrow 00101 \rightarrow 10100$$

- Aby dodać znacznik końca kodu wystarczy dodać na końcu „1”. Dwie jedynki po sobie następujące będą definiowały koniec słowa kodowego (kod Fraenkela–Kleina *C*1)

Fraenkela–Kleina C1

symbol	kod
	12358X
1	11
2	011
3	0011
4	1011
5	00011
6	10011
7	01011
8	000011
9	100011

Wada:
Jak rozdzielić słowa ?

Fraenkela–Kleina *C2*

Kod Fraenkela–Kleina *C2* tworzymy łącząc bity „**10**” z odwrócona reprezentacja Zeckendorfa liczby x pomniejszonej o 1 (x-1).

Liczba 1 jest reprezentowana jako 1 (wyjątek)

Zaleta:
Dwie jedynki po sobie jednoznacznie definiują podział słowa kodowego

symbol	X-1	kod
		1012358
1		1
2	1	101
3	2	1001
4	3	10001
5	4	10101
6	5	100001
7	6	101001
8	7	100101
9	8	1000001

Inne kody statyczne

- Inne kody Fraenkela–Kleina
- Kody Apostolico–Fraenkela oparte na liczbach Fibonacciego (także liczby Fibonacciego wyższych rzędów)
- Kody Golomba
- Kody Rice'a
- Kody Goldbacha
- Kody Wheelera
- Kodowanie interpolacyjne

Kody semi-statyczne

- Kody semi-statyczne to kody statyczne, które powstały po przeanalizowaniu kodowanej wiadomości.
- Symbole występujące najczęściej są kodowane za pomocą najkrótszych słów kodowych.
- Dla różnych wiadomości niezbędna jest inna funkcja przekształcająca alfabet na słowa kodowe.
- Konieczność przekazania kodu do dekodera !!
- Bardziej uniwersalne niż kody statyczne, lepszy współczynnik kompresji.

Dziękuję za uwagę i proszę o pytania