

# Kodowanie predykcyjne

## Kodowanie i kompresja danych - Wykład 6

Maciek Gębala

1 kwietnia 2020

Maciek Gębala Kodowanie predykcyjne

## Motywacje

- W tekstach naturalnych symbole bardzo często zależą od siebie.
- W językach naturalnych na podstawie już przeczytanych symboli można z bardzo dużym prawdopodobieństwem przewidzieć następny symbol (mocna zależność od historii).
- Często zamiast kompresować ciąg wejściowy kompresujemy różnicę między tym ciągiem a ciągiem generowanym przez pewien „zgadywacz”.

Maciek Gębala Kodowanie predykcyjne

## Przykład wykorzystania kontekstu

	A	B	C
A	0.1	0.3	0.6
B	0.6	0.1	0.3
C	0.3	0.6	0.1

- Łatwo sprawdzić, że  $P(A) = P(B) = P(C) = \frac{1}{3}$ , stąd normalny kod Huffmana będzie miał średnią długość  $\frac{5}{3}$ .
- Co będzie jeśli wykorzystamy informację o tym po jakiej literze występuje kolejna, czyli stworzymy trzy kodowania Huffmana, kolejno dla liter występujących po A, B, C?
- Łatwo policzyć, że wtedy średnia długość kodu spadnie do  $\frac{13}{10}$ .
- Gdybyśmy wydłużyli historię (kontekst) moglibyśmy uzyskać większy stopień kompresji, ale ilość kodów roslaby szybko.

Maciek Gębala Kodowanie predykcyjne

## Predykcja z częściowym dopasowaniem (PPM)

- Algorytm dynamiczny wykorzystujący kontekst.
- Specjalny symbol wyjścia oznaczający brak istniejącego kontekstu danej długości (<esc>).
- Ustalamy maksymalny rozmiar kontekstu.
- Dla danej litery szukamy maksymalnego kontekstu, jeśli nie istnieje to wysyłamy symbol wyjścia i sprawdzamy krótszy kontekst. Jeśli istnieje to wysyłamy odpowiedni kod a liczbę użycia litery w tym kontekście zwiększamy o 1.
- Jeśli symbol pojawia się po raz pierwszy dodajemy go bez kontekstu, (kontekst długości -1) z prawdopodobieństwem równym dla każdej litery.

Maciek Gębala Kodowanie predykcyjne

Notatki

Notatki

Notatki

Notatki

## Przykład

- Kodujemy tekst: *this-is-the-tithe*.
- Przyjmujemy, że najdłuższy kontekst ma długość 2 (mamy konteksty długości -1, 0, 1 i 2).
- Po zakodowaniu *this-is* mamy odpowiednie tabele kontekstów.

Litera	Licznik
t	1
h	1
i	1
s	1
-	1

Litera	Licznik
t	1
h	1
i	1
s	1
-	1

## Przykład (this-is)

Kontekst długości 0	
Litera	Licznik
<esc>	1
t	1
h	1
i	2
s	2
-	1

Litera	Licznik
<esc>	1
t	1
h	1
i	2
s	2
-	1

## Przykład (this-is)

Kontekst	Litera	Licznik
t	<esc>	1
	h	1
h	<esc>	1
	i	1
i	<esc>	1
	s	2
s	<esc>	1
	-	1
-	<esc>	1
	i	1

Kontekst	Litera	Licznik
t	<esc> h	1 1
h	<esc> i	1 1
i	<esc> s	1 2
s	<esc> -	1 1
-	<esc> i	1 1

## Przykład (this-is)

Kontekst	Litera	Licznik
th	<esc>	1
	i	1
hi	<esc>	1
	s	1
is	<esc>	1
	-	1
s-	<esc>	1
	i	1
-i	<esc>	1
	s	1

Kontekst	Litera	Licznik
th	<esc> i	1 1
hi	<esc> s	1 1
is	<esc> -	1 1
s-	<esc> i	1 1
-i	<esc> s	1 1

Notatki

Notatki

Notatki

Notatki

- Dla każdego kontekstu możemy teraz utworzyć dynamiczne kody Huffmana.
- Albo wykorzystać dany kontekst z liczbą wystąpień jako prawdopodobieństwa do podziału odcinka w kodowaniu arytmetycznym.
- Najczęściej za maksymalną długość kontekstu przyjmuje się 5.

## CALIC - Context Adaptative Lossless Image Compresion

		NN	NNE
	NW	N	NE
WW	W	X	

Sprawdzamy czy w sąsiedztwie są krawędzie pionowe lub poziome, w tym celu liczymy wartości pomocnicze

$$d_h = |W - WW| + |N - NW| + |NE - N|$$

$$d_v = |W - NW| + |N - NN| + |NE - NNE|$$

## CALIC - Context Adaptative Lossless Image Compresion

Pseudokod algorytmu:

```

if  $d_h - d_v > 80$  then  $\hat{X} \leftarrow N$ 
else if  $d_v - d_h > 80$  then  $\hat{X} \leftarrow W$ 
else
   $\hat{X} \leftarrow (N + W)/2 + (NE - NW)/4$ 
  if  $d_h - d_v > 32$  then  $\hat{X} \leftarrow (\hat{X} + N)/2$ 
  else if  $d_v - d_h > 32$  then  $\hat{X} \leftarrow (\hat{X} + W)/2$ 
  else if  $d_h - d_v > 8$  then  $\hat{X} \leftarrow (3\hat{X} + N)/4$ 
  else if  $d_v - d_h > 8$  then  $\hat{X} \leftarrow (3\hat{X} + W)/4$ 

```

Predykcję  $\hat{X}$  można jeszcze bardziej uszczegółowić.

Kodujemy ciąg różnic  $X - \hat{X}$ .

## JPEG-LS (bezstratny)

NW	N
W	X

7 schematów predykcji:

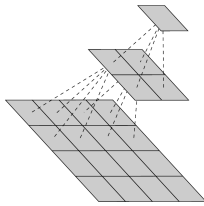
- 1  $\hat{X} = W$
- 2  $\hat{X} = N$
- 3  $\hat{X} = NW$
- 4  $\hat{X} = N + W - NW$
- 5  $\hat{X} = N + (W - NW)/2$
- 6  $\hat{X} = W + (N - NW)/2$
- 7  $\hat{X} = (N + W)/2$

NW	N
W	X

Nowy standard:

if  $NW \geq \max(W, N)$  then  $\hat{X} \leftarrow \max(W, N)$   
 else if  $NW \leq \min(W, N)$  then  $\hat{X} \leftarrow \min(W, N)$   
 else  $\hat{X} \leftarrow W + N - NW$

## Wykorzystanie poziomów rozdzielczości

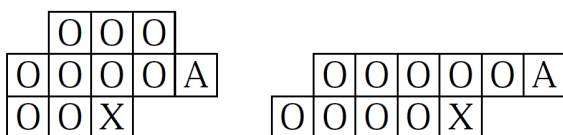


- Kodujemy obraz wysyłając najpierw średni kolor kwadratów  $2^k \times 2^k$  a następnie różnice między tą średnią a średnią kwadratów o rozmiarach  $2^{k-1} \times 2^{k-1}$ .
- Kończymy na pikslach (kwadraty  $2^0 \times 2^0$ ).
- Różnice nie są dużymi liczbami i łatwo poddają się kompresji.

## JBIG – idea

- Obraz przesyłamy „progresywnie”, coraz wyższa rozdzielczość.
- Prawdopodobieństwo wystąpienia czarnego punktu w „białym” sąsiedztwie jest zdecydowanie niższe niż w sąsiedztwie zawierającym już czarne piksele.
- Sąsiedztwo stanowią zakodowane wcześniej piksele (np. kilka poprzednich wierszy obrazu) wraz z wyszczególnionym pikselem **A** (zmiany jego położenia są przesyłane do dekodera).
- Algorytm wykorzystuje od 1024 do 4096 różnych koderów (tzn. dla różnych kontekstów kodowanego piksela).
- W procesie kodowania wykorzystywane jest kodowanie arytmetyczne.

## JBIG – sąsiedztwo



Sąsiedztwo 3- i 2-wierszowe

Piksele **O** i **A** stanowią „sąsiedztwo” kodowanego piksela **X**. Położenie **A** może być zmieniane, tak by oddać charakter danych (np. pionowe pasy co 50 pikseli).

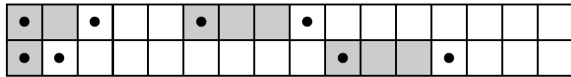
Notatki

Notatki

Notatki

Notatki

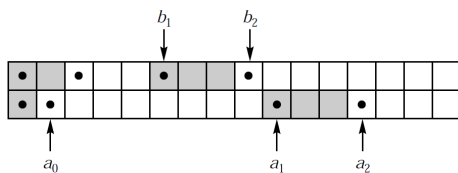
## Kodowanie obrazów czarno-białych (faksy) – MH



- Linie takich obrazów zawierają na przemian bloki białe i czarne.
- Możemy przysłać więc tylko długości takich bloków (zakładając, że pierwszy jest biały).
- Dodatkowo każdą liczbę przedstawiamy jako parę  $m, t$ , gdzie  $l = 64m + t$  dla  $t = 0, \dots, 63$  i  $m = 0, \dots, 27$ .
- Do kodowania białych i czarnych serii używamy różnych koderów.
- ... ale możemy wykorzystać korelację pomiędzy kolejnymi liniami obrazu.

Maciek Gębala Kodowanie predyktoryczne

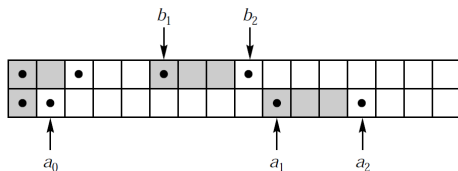
## Kodowanie faksów – MR



- $a_0$  ostatni piksel znany koderowi i dekoderowi (lub fikcyjny biały piksel).
- $a_1$  pierwszy piksel przejścia na prawo od  $a_0$  (innego koloru!).
- $a_2$  drugi piksel przejścia na prawo od  $a_0$ .
- $b_1$  pierwszy piksel przejścia na prawo i powyżej od  $a_0$  o kolorze przeciwnym.
- $b_2$  pierwszy piksel przejścia na prawo od  $b_1$ .

Maciek Gębala Kodowanie predyktoryczne

## Kodowanie faksów – MR (2)

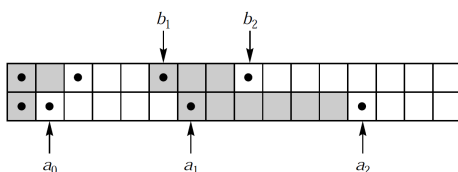


Jeżeli  $b_1$  i  $b_2$  znajdują się pomiędzy  $a_1$  i  $a_2$ , to wysyłamy kod 0001, nowym pikselem  $a_0$  staje się piksel pod  $b_2$ , pozostałe aktualizujemy zgodnie z definicją.

Niektóre wiersze (co pewną wartość) są kodowane bez uwzględnienia wartości poprzedniego wiersza, aby zapobiec propagacji błędów.

Maciek Gębala Kodowanie predyktoryczne

## Kodowanie faksów – MR (3)



Jeżeli  $a_1$  znajduje się przed  $b_2$  rozpatrujemy 2 przypadki:

- jeżeli odległość od  $a_1$  do  $b_2$  mniejsza lub równa trzy, to przesyłamy odpowiedni kod do dekodera i aktualizujemy pozycję ( $a_0$  na  $a_1$  itd.) – tzw. tryb pionowy,
- jeżeli odległość jest większa, to postępujemy podobnie jak w przypadku kodowania jednowymiarowego (MH).

Maciek Gębala Kodowanie predyktoryczne

Notatki

Notatki

Notatki

Notatki

- Techniki oparte na prawdopodobieństwach: kody Huffmana, Tunstalla, kodowanie arytmetyczne.
- Metody słownikowe: LZ77, LZ78, LZW.
- Kodowanie predykcyjne: PPM, CALIC, JPEG-LS, run-length encoding.
- Stosowanie kilku metod po kolei (bzp2).

## Notatki

[illegible]

## Notatki

[illegible]

## Notatki

[illegible]

## Notatki

[illegible]