

Kodowanie i kryptografia

Blokowe kody liniowe

dr Robert Borowiec

Politechnika Wrocławska

Instytut Telekomunikacji i Akustyki

pokój 908, C-5

tel. 3203083

e-mail: robert.borowiec@ita.pwr.wroc.pl

www: lstwww.ita.pwr.wroc.pl/~RB/

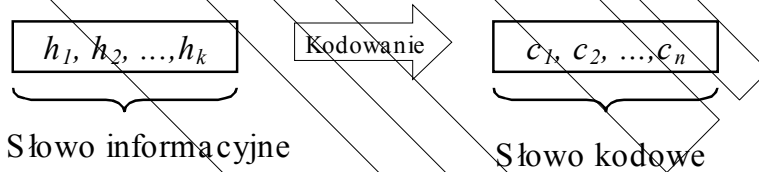
Wykład IV

Plan wykładu

- Definicja blokowego kodu liniowego
- Parametry kodu blokowego
- Sposoby kodowania informacji
- Tworzenie kodu
- Kody dualne
- Metryka przestrzeni
- Zdolność korekcyjna kodu
- Przykłady wybranych kodów liniowych

Definicja blokowego kodu liniowego

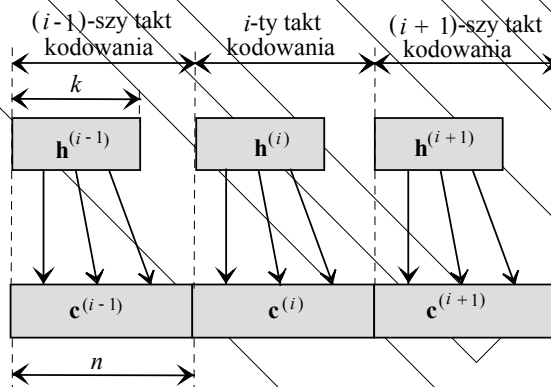
- Kodowanie blokowe polega na przekształceniu k -pozycyjnych q -narych ciągów informacyjnych $\mathbf{h}=(h_1, h_2, \dots, h_k)$ w n -pozycyjne q -narne ciągi kodowe $\mathbf{c}=(c_1, c_2, \dots, c_n)$



- Formalnie proces kodowania blokowego można zapisać:

$$\bigwedge_{\mathbf{h}_i \in \{\mathbf{h}\}} \bigvee_{\mathbf{c}_i \in \{\mathbf{c}\}} f : \mathbf{h}_i \rightarrow \mathbf{c}_i, \text{ przy czym } f : \mathbf{h} \Leftrightarrow \mathbf{c}$$

Proces kodowania blokowego



Parametry kodu blokowego

- Blokowy kod nadmiarowy oznaczamy symbolem (n, k) . Jednoznaczne określenie kodu (n, k) wymaga podania zbiorów $\{\mathbf{h}\}$ i $\{\mathbf{c}\}$ oraz funkcji f , a więc

$$(n, k) \equiv (\{\mathbf{h}\}, \{\mathbf{c}\}, f).$$

- Parametry kodu blokowego

Nadmiar kodowy

$$\rho_k = \frac{n - k}{n} = 1 - \frac{k}{n}$$

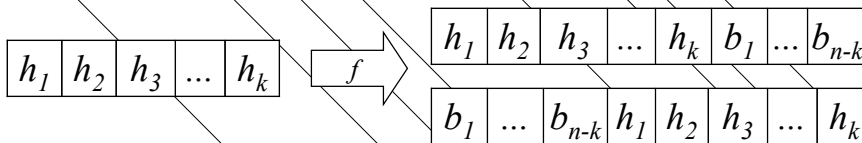
Sprawność

$$\eta_k = \frac{k}{n} = 1 - \rho_k$$

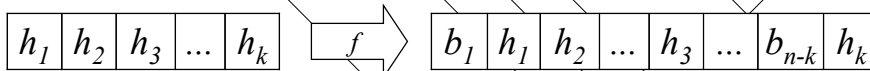
Podział kodów

ze względu sposób transmisji bitów informacyjnych

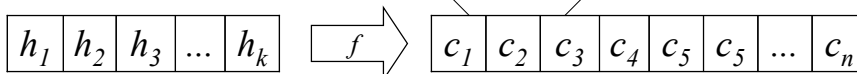
Kod systematyczny rozdzielny



Kody systematyczny nierozdzielny

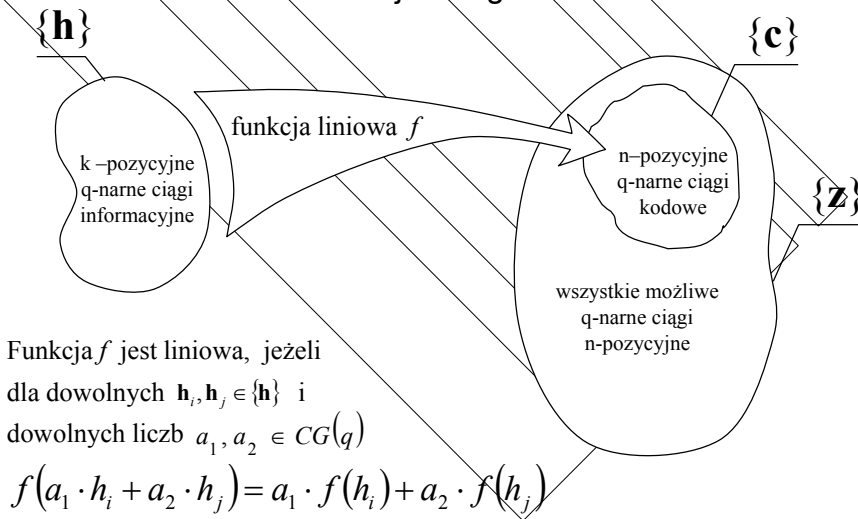


Kod niesystematyczny



Blokowe kody liniowe

Podójście ogólne



Robert Borowiec

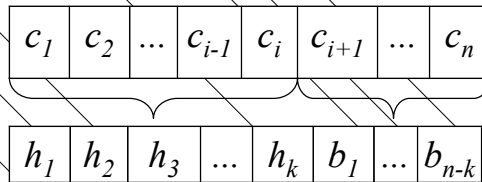
Kodowanie i kryptografia
Wykład IV. strona 7/54

Blokowe kody liniowe

Podójście szczególne

Struktura słowa kodu

$$c_i = \begin{cases} h_i & i = 1, 2, \dots, k \\ b_{i-k} & i = k+1, k+2, \dots, n \end{cases}$$



wszystkie $(n-k)$ bitów parzystości są liniowymi sumami bitów informacyjnych:

$$b_i = p_{1,i} \cdot h_1 + p_{2,i} \cdot h_2 + \dots + p_{k,i} \cdot h_k$$

gdzie:

$$p_{ij} = \begin{cases} 1 & \text{jeśli } b_i \text{ zależy od } h_j \\ 0 & \text{w przypadku przeciwnym} \end{cases}$$

Robert Borowiec

Kodowanie i kryptografia
Wykład IV. strona 8/54

Blokowe kody liniowe

Tworzenie kodu-podejście szczególne

Zapiszmy równania w postaci macierzowej

$$\mathbf{h}=[h_1, h_2, \dots, h_k]$$

$$\mathbf{b}=[b_1, b_2, \dots, b_{n-k}]$$

$$\mathbf{c}=[c_1, c_2, \dots, c_n]$$

$$\mathbf{b}=\mathbf{hP}$$

$$\mathbf{P} \equiv \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1,n-k} \\ p_{21} & p_{22} & \dots & p_{2,n-k} \\ \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{k,n-k} \end{bmatrix}$$

Ponieważ wektor \mathbf{c} jest złożeniem

wektora \mathbf{h} oraz \mathbf{b} to: $\mathbf{c}=[\mathbf{h}|\mathbf{b}]$

$$\text{Stąd: } \mathbf{c}=\mathbf{h}[\mathbf{I}_k|\mathbf{P}]$$

$$\mathbf{I}_k \equiv \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{G}=[\mathbf{P}|\mathbf{I}_k]$$

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 9/54

Blokowe kody liniowe

Tworzenie kodu-podejście szczególne

I w efekcie możemy zapisać:

$$\mathbf{c}=\mathbf{hG}$$

gdzie:

$$\mathbf{G}=[\mathbf{I}_k|\mathbf{P}]$$

Macierz \mathbf{G} jest nazywana *macierzą generującą* kod liniowy (n, k)

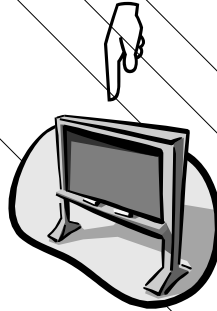
$$\mathbf{G} \equiv \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \dots & \dots & \dots & \dots \\ g_{k1} & g_{k2} & \dots & g_{kn} \end{bmatrix}$$

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 10/54

Przykład 2.0

Dlaczego zastosowanie kodu nadmiarowego pozwala wykryć błędy?



Blokowe kody liniowe

Podejście ogólne

Niech zbiór k liniowo niezależnych wektorów $\{\alpha\} = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ stanowi bazę przestrzeni $\{h\}$, a zbiór n liniowo niezależnych wektorów $\{\beta\} = \{\beta_1, \beta_2, \dots, \beta_n\}$ stanowi bazę przestrzeni $\{z\}$.

$$f(\alpha_l) = \sum_{i=1}^n a_{l,i} \beta_i, \quad l = 1, 2, \dots, k.$$

lub macierzowo

$$\mathbf{A} \rightarrow \mathbf{M}_f^{\mathbf{A}, \mathbf{B}} \mathbf{B},$$

$$\mathbf{A} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}, \quad \mathbf{M}_f^{\mathbf{A}, \mathbf{B}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kn} \end{bmatrix} \mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

Blokowe kody liniowe

Podejście ogólne

Przedstawmy ciąg informacyjny \mathbf{h} jako liniową kombinację wektorów bazy $\{\alpha_i\}$

$$\mathbf{h} = \sum_{l=1}^k b_l \alpha_l \xrightarrow{f(\alpha_l) = \sum_{i=1}^n a_{li} \beta_i} \mathbf{c} = f(\mathbf{h}) = f\left(\sum_{l=1}^k b_l \alpha_l\right) = \sum_{l=1}^k b_l f(\alpha_l) = \sum_{l=1}^k b_l \sum_{i=1}^n a_{li} \beta_i = \sum_{i=1}^n \left(\sum_{l=1}^k b_l a_{li}\right) \beta_i = \sum_{i=1}^n d_i \beta_i, \quad \Leftarrow d_i = \sum_{l=1}^k b_l a_{li}$$

To samo w zapisie macierzowym

$$\mathbf{h} = \mathbf{bA} \rightarrow \mathbf{bM}_f^{\mathbf{A}, \mathbf{B}} \mathbf{B} = \mathbf{dB} = \mathbf{c},$$

w której k -poziomy wektor \mathbf{b} określa liniową kombinację wektorów $\alpha_i \in \{\alpha\}$, a n -poziomy $\mathbf{d} = \mathbf{bM}_f^{\mathbf{A}, \mathbf{B}}$ wektor - kombinację liniową wektorów $\beta_i \in \{\beta\}$ odpowiadającą ciągowi kodowemu \mathbf{c} .

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 13/54

Blokowe kody liniowe

Wyznaczenie ciągów kodowych-podejście ogólne

Przy zadanych bazach \mathbf{A} i \mathbf{B} oraz znanej funkcji f , ciąg kodowy \mathbf{c} odpowiadający ciągowi informacyjnemu \mathbf{h} wyznaczamy następująco:

-z równania $\mathbf{h} = \mathbf{bA}$ określamy $\mathbf{b} = \mathbf{hA}^{-1}$,

-obliczamy $\mathbf{d} = \mathbf{bM}_f^{\mathbf{A}, \mathbf{B}}$

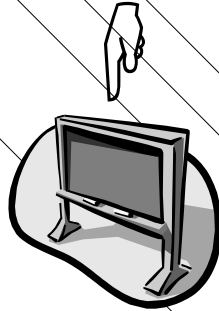
- wyznaczamy ciąg kodowy $\mathbf{c} = \mathbf{dB}$.

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 14/54

Przykład 2.1

Wyznaczanie ciągów kodowych



Blokowe kody liniowe

Wyznaczenie ciągów kodowych-podejście ogólne

Jeżeli zadane bazy A i B są bazami jednostkowymi oznaczonymi A_1 i B_1 , to

$$h = b \rightarrow hM_f^{A_1, B_1} = c,$$

a więc macierz kodowania

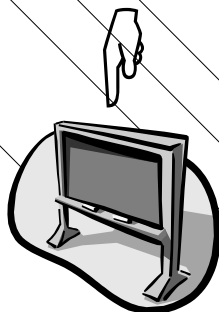
$$G \equiv M_f^{A_1, B_1}$$



$$c = hG$$

Przykład 2.2

Wyznaczanie ciągów kodowych w bazach jednostkowych



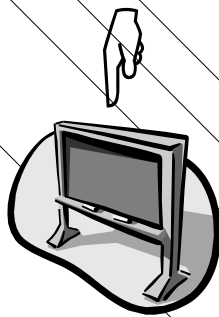
Blokowe kody liniowe

Kody ściśle równoważne

h	hG'	hG''
0 0 0	0 0 0 0 0	0 0 0 0 0
0 0 1	0 1 1 0 1	0 1 0 1 0
0 1 0	1 1 1 1 1	0 1 1 0 1
0 1 1	1 0 0 1 0	0 0 1 1 1
1 0 0	0 0 1 1 1	1 1 0 0 0
1 0 1	0 1 0 1 0	1 0 0 1 0
1 1 0	1 1 0 0 0	1 0 1 0 1
1 1 1	1 0 1 0 1	1 1 1 1 1

Przykład 2.3 i 2.4

Wyznaczanie kodów ściśle równoważnych



Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 19/54

Liniowe kody dualne

Macierz kontrolna kodu, Syndrom

W n -wymiarowej przestrzeni liniowej $\{\mathbf{z}\}$ złożonej z wszystkich możliwych q -narych ciągów n -pozycyjnych istnieje jedna i tylko jedna podprzestrzeń $(n-k)$ -wymiarowa $\{\mathbf{v}\}$ związana z podprzestrzenią kodową $\{\mathbf{c}\}$ w ten sposób, że dla dowolnie wybranych $\mathbf{c} \in \{\mathbf{c}\}$ i $\mathbf{v} \in \{\mathbf{v}\}$ znika iloczyn skalarny

$$\mathbf{c} \cdot \mathbf{v} = 0$$



Kod liniowy (n, k) można określić - z dokładnością do kodu ściśle równoważnego - przez podanie macierzy \mathbf{H} generującej kod dualny $(n, n-k)$.

$$\mathbf{H} = \left[-\mathbf{P}_{k,r}^T \mathbf{I}_r \right], \quad \text{przy czym } r = n - k.$$

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 20/54

Liniowe kody dualne

Macierz kontrolna kodu, Syndrom

Iloczyn macierzy generującej \mathbf{G} kodu (n,k) i macierzy generującej kod dualny $(n,n-k)$ zeruje się

$$\mathbf{GH}^T = \mathbf{O}_{kr}$$

$$\mathbf{GH}^T = [\mathbf{I}_k \mathbf{P}_{kr}] \begin{bmatrix} -\mathbf{P}_{kr} \\ \mathbf{I}_r \end{bmatrix} = -\mathbf{I}_k \mathbf{P}_{kr} + \mathbf{P}_{kr} \mathbf{I}_r = -\mathbf{P}_{kr} + \mathbf{P}_{kr} = \mathbf{O}_{kr}$$

Macierz \mathbf{H} nazywa się *macierzą kontrolną kodu* (n, k) . Dla kodów binarnych ($q = 2$) wyrażenie opisujące macierz kontrolną, przyjmuje postać

$$\mathbf{H} = [\mathbf{P}_{k,r}^T \mathbf{I}_r].$$

Liniowe kody dualne

Syndrom

Z macierzą kontrolną kodu wiąże się pojęcie *syndromu błędów*, zwanego krótko *syndromem*. Syndrom n -pozycyjnego ciągu \mathbf{a} definiujemy następująco

$$\mathbf{S}(\mathbf{a}) \equiv \mathbf{aH}^T$$

Syndrom jest ciągiem $r=n-k$



Syndrom dla ciągu kodowego jest równy wektorowi zerowemu

$$\mathbf{S}(\mathbf{c}) = \mathbf{cH}^T = \mathbf{hGH}^T = \mathbf{O}_r$$

Liniiowe kody dualne

Dekodowanie z syndromem

Oznaczmy ciąg odebrany przez $\mathbf{y} = (y_1, y_2, \dots, y_n)$

$$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

gdzie \mathbf{e} jest wektorem błędów zdefiniowanym następująco

$$\mathbf{e} = (e_1, e_2, \dots, e_n) \quad e_i = \begin{cases} 1 & \text{gdy jest błąd na } i\text{-tej pozycji} \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

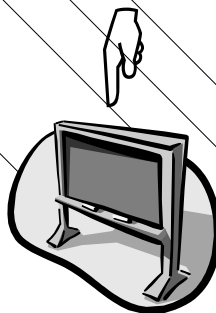


$$\mathbf{s} = S(\mathbf{y}) = S(\mathbf{c} + \mathbf{e}) = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T = S(\mathbf{e})$$

Syndrom jest ciągiem $r=n-k$ -pozycyjnym zależnym tylko od rozkładu błędów

Przykład 2.6

Syndrom



Metryka przestrzeni kodowej

Analizę właściwości kodów nadmiarowych ułatwia zdefiniowanie metryki przestrzeni $\{z\}$, to jest miary odległości pomiędzy dowolnymi "punktami" (elementami) tej przestrzeni. Jeżeli $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \{z\}$, to miara odległości $d(\mathbf{a}, \mathbf{b})$ musi spełniać następujące aksjomaty:

- (1) $d(\mathbf{a}, \mathbf{b}) \geq 0$; $d(\mathbf{a}, \mathbf{b}) = 0$ tylko dla $\mathbf{a} = \mathbf{b}$;
- (2) $d(\mathbf{a}, \mathbf{b}) \leq d(\mathbf{a}, \mathbf{c}) + d(\mathbf{c}, \mathbf{b})$;
- (3) $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$.

Metryka Hamminga

Wagą Hamminga pozycji a_i ciągu \mathbf{a} nazywamy liczbę

$$w_H(a_i) \equiv \begin{cases} 0, & \text{jeżeli } a_i = 0; \\ 1, & \text{jeżeli } a_i \neq 0. \end{cases} \quad (2.39)$$

Wagą Hamminga ciągu \mathbf{a} nazywamy sumę wag jego pozycji

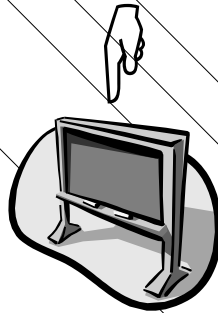
$$w_H(\mathbf{a}) \equiv \sum_{i=1}^n w_H(a_i). \quad (2.40)$$

Odległością Hamminga między ciągami \mathbf{a} , \mathbf{b} nazywamy liczbę

$$d_H(\mathbf{a}, \mathbf{b}) \equiv w_H(\mathbf{a} - \mathbf{b}). \quad (2.41)$$

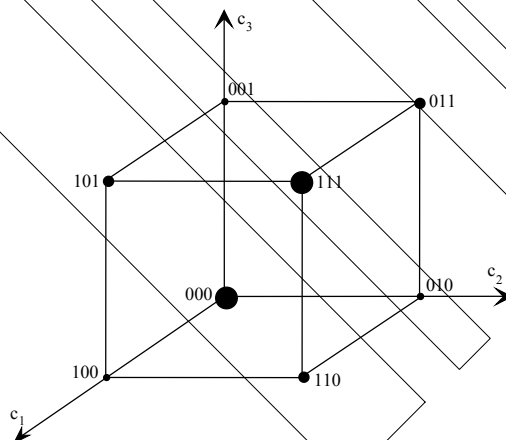
Przykład 2.8

Metryka Hamminga



Metryka Hamminga

Interpretacja graficzna



Metryka Lee

Wagą Lee pozycji a_i q -narnego ciągu \mathbf{a} stanowi liczba

$$w_L(a_i) \equiv \begin{cases} |a_i|, & \text{jeżeli } 0 \leq |a_i| < \frac{q}{2}, \\ q - |a_i|, & \text{jeżeli } \frac{q}{2} \leq |a_i| \leq q-1. \end{cases} \quad (2.44)$$

Wagą Lee n -pozycyjnego q -narnego ciągu \mathbf{a} nazywamy sumę wag jego pozycji

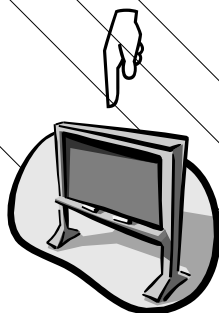
$$w_L(\mathbf{a}) \equiv \sum_{i=1}^n w_L(a_i). \quad (2.45)$$

Odległość Lee dwóch n -pozycyjnych q -narnych ciągów \mathbf{a} i \mathbf{b} definiuje się jako

$$d_L(\mathbf{a}, \mathbf{b}) \equiv w_L(\mathbf{a} - \mathbf{b}). \quad (2.46)$$

Przykład 2.10

Metryka Lee



Zdolność detekcyjna kodu nadmiarowego

Odległość minimalna kodu nadmiarowego

$$d_{\min} \equiv \min_{\substack{i, j = 1, 2, \dots, q^k \\ i \neq j}} d(\mathbf{c}_i, \mathbf{c}_j). \quad (2.47)$$

Zdolność detekcyjna kodu - największa krotność błędów wykrywanych przez blokowy kod nadmiarowy (n, k) o odległości minimalnej d_{\min} wynosi

$$\gamma = d_{\min} - 1. \quad (2.48)$$

Zdolność korekcyjna kodu nadmiarowego

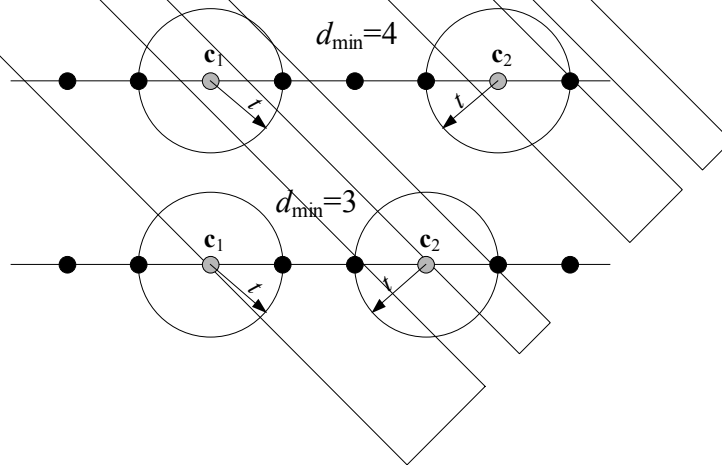
Reguła decyzyjna

Optymalną regułą decyzyjną w przypadku dekodowania korekcyjnego jest reguła maksymalizująca prawdopodobieństwo a posteriori $P(\mathbf{c}|\mathbf{y})$ nadania ciągu $\mathbf{c} = \mathbf{c}^*$, przy warunku że został odebrany ciąg \mathbf{y} .

$$\begin{aligned} \mathbf{c}^* &= \mathbf{c}_l \text{ taki, że } d(\mathbf{c}_l, \mathbf{y}) \leq d(\mathbf{c}_i, \mathbf{y}); \\ i &= 1, 2, \dots, q^k; \\ i &\neq l \end{aligned} \quad (2.49)$$

Zdolność korekcyjna kodu nadmiarowego

Reprezentacja graficzna



Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 33/54

Zdolność korekcyjna kodu nadmiarowego

Zdolność korekcyjna kodu określa największą krotność błędów korygowanych przez liniowy kod nadmiarowy (n, k) o odległości minimalnej d_{\min} i wynosi

$$t = E\left\{\frac{d_{\min} - 1}{2}\right\}, \quad (2.50)$$

przy czym $E\{x\}$ oznacza część całkowitą liczby x .

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 34/54

Metody wyznaczanie d_{\min}

1. Korzystając z właściwości blokowych kodów liniowych:
Suma dwóch dowolnych wektorów kodowych daje w wyniku inny wektor kodowy, a więc:

$$d_{\min} = \min(w_H(\mathbf{c}_i)) \quad i = 1, 2, \dots, q^k \quad \text{przy czym } \mathbf{c}_i \neq \mathbf{0}$$

1. Korzystając z macierzy kontrolnej H^T .
Minimalna liczba wierszy macierzy H^T sumujących się do wektora zerowego wyznacza d_{\min} kodu

Dekodowanie korekcyjne

Tablica dekodowania q -narnego blokowego kodu liniowego (n, k)

$\zeta_1 = \mathbf{c}_1 = \mathbf{O}_n$	\mathbf{c}_2	...	\mathbf{c}_l	...	\mathbf{c}_{q^k}
ζ_2	$\mathbf{c}_2 + \zeta_2$...	$\mathbf{c}_l + \zeta_2$...	$\mathbf{c}_{q^k} + \zeta_2$
...
ζ_m	$\mathbf{c}_2 + \zeta_m$...	$\mathbf{c}_l + \zeta_m$...	$\mathbf{c}_{q^k} + \zeta_m$
...
ζ_{q^r}	$\mathbf{c}_2 + \zeta_{q^r}$...	$\mathbf{c}_l + \zeta_{q^r}$...	$\mathbf{c}_{q^k} + \zeta_{q^r}$

Dekodowanie korekcyjne

Właściwości tablicy dekodowania

- Tablica dekodowania charakteryzuje się następującymi właściwościami:
 - ⇒ w tablicy nie występują powtarzające się elementy,
 - ⇒ elementy tej samej warstwy mają ten sam syndrom,
 - ⇒ różnica dwóch ciągów należących do tej samej warstwy jest ciągiem kodowym,
 - ⇒ różnica dwóch ciągów należących do różnych warstw nie jest ciągiem kodowym,
 - ⇒ elementy należące do różnych warstw mają różne syndromy,
 - ⇒ w tej samej warstwie nie może wystąpić więcej niż jeden ciąg o wadze większej niż zdolność korekcyjna kodu t ,
 - ⇒ oznaczmy kolumnę tablicy dekodowania zawierającą ciąg kodowy \mathbf{c}_l przez Γ_l ; dla $i \neq l$ nie istnieje żaden taki ciąg kodowy \mathbf{c}_i , który leżałby bliżej ciągów \mathbf{z} należących do kolumny Γ_l niż ciąg kodowy \mathbf{c}_l .

Dekodowanie korekcyjne

Reguła korygowania błędów

Korygowanie błędów opiera się na następującej regule:

jeżeli $\mathbf{y} \in \Gamma_m$, to $\mathbf{c}^* = \mathbf{c}_m$. (2.55)

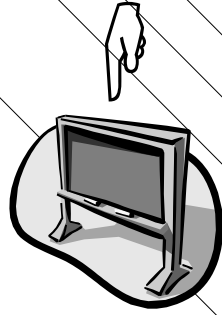
Dekodowanie korekcyjne przebiega w trzech etapach:

- obliczenie syndromu $S(\mathbf{y})$ odebranego ciągu \mathbf{y} ,
- odszukanie, na podstawie $S(\mathbf{y})$, elementu ζ tworzącego warstwę, w której leży ciąg \mathbf{y} ,
- skorygowanie błędów według zależności

$$\mathbf{c}^* = \mathbf{y} - \zeta$$

Przykład 2.11

Przykład tworzenia kodu oraz jego tablicy dekodowania



Wybrane kody liniowe

- Kod z kontrolą parzystości
- Kod z m -krotnym powtarzaniem
- Kod iterowany m -krotnie
- Kody Hamminga
- Wydłużony kod liniowy
- Kody równoległe
- Kody Mac Donalda
- Kody Reeda-Mullera

Wybrane kody liniowe

Kod z kontrolą parzystości

Ciąg kodowy tego kodu powstaje w wyniku dołączenia do ciągu informacyjnego jednej pozycji kontrolnej, tak aby ilość jedynek w ciągu była parzysta.

h_1	h_2	h_3	\dots	h_{n-1}	b_1
-------	-------	-------	---------	-----------	-------

Parametry kodu: $(n, n-1)$

Sprawność: $\eta = 1 - \frac{1}{n}$

Odległość minimalna: $d_{\min} = 2$

Kod wykrywa pojedyncze błędy

Macierz generująca

$$\mathbf{G} = [\mathbf{I}_{n-1} \mid \mathbf{1}_{n-1}]$$

Macierz kontrolna

$$\mathbf{H} = [\mathbf{1}_n]$$

Reguła kodowania

$$\sum_{i=1}^n c_i = 0$$

Wybrane kody liniowe

Kod z m -krotnym powtórzeniem

Ciągi kodowe tego kodu powstają w wyniku m -krotnego powtórzenia ciągu informacyjnego

h_1	h_2	\dots	h_k	h_1	h_2	\dots	h_k	\dots	\dots	h_1	h_2	\dots	h_k
1				2				\dots		m			

Oznaczenie kodu: (mk, k)

Sprawność: $\eta = 1/m$

Odległość minimalna: $d_{\min} = m$

Reguła kodowania

$$\mathbf{c} = (\underbrace{\mathbf{h}, \mathbf{h}, \dots, \mathbf{h}}_{m \text{ razy}})$$

Macierz kontrolna

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{I}_k \\ \vdots \\ \mathbf{I}_k \\ \mathbf{I}_{(m-1),k} \end{bmatrix}$$

$\underbrace{\mathbf{I}_k}_{m-1 \text{ razy}}$

Macierz generująca

$$\mathbf{G} = [\underbrace{\mathbf{I}_k \mathbf{I}_k \dots \mathbf{I}_k}_{m \text{ razy}}]$$

Wybrane kody liniowe

Kod iterowany m -krotnie

Ciągi kodowe tego kodu powstają w wyniku m -krotnego kodowania informacji m kodami według następujących reguł:

1. Ciąg informacyjny \mathbf{h} o długości $k=k_1 \times k_2 \times k_3 \times \dots \times k_m$ przedstawia się w postaci m -wymiarowego "prostopadłościanu" o wymiarach $k_1 \times k_2 \times k_3 \times \dots \times k_m$ w układzie współrzędnych $x_1, x_2, x_3, x_1, \dots, x_m$.
2. Podciągi informacyjne o długości k_1 , umieszczone równolegle do osi Ox_1 , koduje się kodem (n_1, k_1) . W wyniku tej operacji powstaje prostopadłościan o wymiarach $n_1 \times k_2 \times k_3 \times \dots \times k_m$.
3. Podobnie, podciągi informacyjne o długości k_2 , umieszczone równolegle do osi Ox_2 , koduje się kodem (n_2, k_2) , otrzymując prostopadłościan o wymiarach $n_1 \times n_2 \times k_3 \times \dots \times k_m$.
4. Postępując w ten sposób ze wszystkimi podciągami informacyjnymi, tzn. kodując je kodami $\dots, (n_3, k_3), (n_4, k_4), \dots, (n_m, k_m)$, otrzymujemy ciąg kodowy w postaci m -wymiarowego prostopadłościanu o wymiarach $n_1 \times n_2 \times n_3 \times \dots \times n_m$.

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 43/54

Kod iterowany m -krotnie cd..

Oznaczenie kodu:

$$\left(\prod_{i=1}^m n_i, \prod_{i=1}^m k_i \right)$$

Sprawność:

$$\eta = \prod_{i=1}^m \eta_i.$$

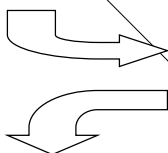
Odległość minimalna:

$$d_{\min} = \prod_{i=1}^m d_{\min,i};$$

Przykład

Kod dwukrotnie iterowany (40, 16) można uzyskać stosując następujące dwa kody: kod z powtarzaniem (8, 4) oraz kod z kontrolą parzystości (5, 4)

$\mathbf{h} = 1001110000111000$



1	0	0	1	1	0	0	1
1	1	0	0	1	1	0	0
0	0	1	1	0	0	1	1
1	0	0	0	1	0	0	0
1	1	1	0	1	1	1	0

$\mathbf{c} = 1001100111001100001100111000100011101110$

Robert Borowiec

Kodowanie i kryptografia
Wykład IV, strona 44/54

Wybrane kody liniowe

Kod Hamminga

Obecnie jest to każdy blokowy kod liniowy (n, k) , spełniający następujące parametry: $n=2^m-1$, $k=2^m-m-1$, $d_{\min}=3$ przy czym m jest liczbą naturalną. Przykładem kodu Hamminga może być kod $(7, 4)$ generowany przez macierz:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Macierz kontrolna kodu

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Wybrane kody liniowe

Wydłużony kod liniowy

Wydłużony kod liniowy powstaje z kodu liniowego $(n-1, k)$ o nieparzystej odległości minimalnej d'_{\min} . Procedura wydłużania polega na wprowadzeniu do wyjściowego ciągu kodowego dodatkowej pozycji kontrolnej, sprawdzającej parzystość jedynek według reguły

$$c_i = \begin{cases} c_i & \text{dla } i = 1, 2, \dots, n-1; \\ \sum_{i=1}^{n-1} c_i & \text{dla } i = n. \end{cases}$$

Wydłużony kod liniowy ma parzystą odległość minimalną $d_{\min} = d'_{\min} + 1$.

Wybrane kody liniowe

Kod równoodległy

Ciąg kodowy tego kodu zawiera wszystkie kombinacje liniowe pozycji ciągu informacyjnego, tzn.:

$$\binom{k}{1} = k \text{ pozycji informacyjnych: } h_1, h_2, \dots, h_k;$$

$$\binom{k}{2} \text{ pozycji kontrolnych typu } h_i + h_j; i, j = 1, 2, \dots, k; i \neq j;$$

$$\vdots$$

$$\binom{k}{m} \text{ pozycji kontrolnych typu } h_i + h_j + \dots + h_m; i, j, m = 1, 2, \dots, k; i \neq j \neq \dots \neq m;$$

$$\binom{k}{k} = 1 \text{ pozycję kontrolną typu } \sum_{i=1}^k h_i.$$

Wybrane kody liniowe

Kod równoodległy cd..

Długość ciągu	Oznaczenie kodu:	Odległość minimalna:
$n = \sum_{i=1}^k \binom{k}{i} = 2^k - 1$	$\Rightarrow (2^k - 1, k)$	$d_{\min} = 2^{k-1}$

Przykład

Dla $k = 4$ otrzymuje się kod równoodległy (15, 4), którego ciągi opisuje zależność:

$$\mathbf{c} = h_1, h_2, h_3, h_4, (h_1 + h_2), (h_1 + h_3), (h_1 + h_4), (h_2 + h_3), (h_2 + h_4), (h_3 + h_4), \\ (h_1 + h_2 + h_3), (h_1 + h_2 + h_4), (h_1 + h_3 + h_4), (h_2 + h_3 + h_4), (h_1 + h_2 + h_3 + h_4).$$

Wybrane kody liniowe

Kod Mac Donalda

Ciąg kodowy zawiera:

-pozycje informacyjne w liczbie k , dla których zachodzi

$$c_{i_j} = h_i, \quad j = 1, 2, \dots, k;$$

-pozycje kontrolne w postaci co najmniej dwuargumentowych kombinacji liniowych pozycji ciągu informacyjnego.

Przykład

Kod Mac Donalda (15,4), którego ciąg opisuje zależność:

$$\mathbf{c} = h_1, h_2, h_3, h_4, (h_1 + h_2 + h_3), (h_1 + h_2 + h_4), (h_1 + h_3 + h_4), (h_2 + h_3 + h_4), (h_1 + h_2 + h_3 + h_4)$$

Wybrane kody liniowe

Kod Reeda-Mullera

Parametry binarnych kodów RM są następujące:

$$n = 2^m; \quad k = \sum_{i=0}^{\lambda} \binom{m}{i}; \quad d_{\min} = 2^{m-\lambda};$$

przy czym m, λ - liczby naturalne ($m > \lambda$); λ - rząd kodu

Konstrukcja kodów RM opiera się na n -wymiarowej przemiennej i łącznej algebrze, jaka powstaje w wyniku wprowadzenia w n -wymiarowej przestrzeni liniowej operacji mnożenia wektorowego ciągów, zdefiniowanej następująco:

$$\mathbf{a} \times \mathbf{b} \equiv (a_1 b_1, a_2 b_2, \dots, a_n b_n).$$

Kod Reeda-Mullera cd.. Tworzenie bazy

Dla zadanej wartości m konstruuje się algebrę, której bazę \mathbf{B}_m stanowią następujące n -pozycyjne ciągi binarne:

- ciąg \mathbf{u}_0 złożony z samych jedynek;
- m ciągów \mathbf{u}_i , $i=1,2,\dots,m$, które wypisane pod sobą tworzą macierz o kolumnach będących m -pozycyjnymi ciągami binarnymi;
- $\binom{m}{2}$ ciągów typu $\mathbf{u}_{ij} = \mathbf{u}_i \times \mathbf{u}_j$ dla $i, j = 1, 2, \dots, m$, $i \neq j$;
- $\binom{m}{3}$ ciągów typu aż do ciągu typu:

$$\mathbf{u}_{ijl} = \mathbf{u}_i \times \mathbf{u}_j \times \mathbf{u}_l \text{ dla } i, j, l = 1, 2, \dots, m, \ i \neq j, l, \ j \neq l \text{ itd.},$$

Kod Reeda-Mullera cd.. Przykład bazy \mathbf{B}_4

\mathbf{u}_0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
\mathbf{u}_1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
\mathbf{u}_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
\mathbf{u}_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
\mathbf{u}_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
\mathbf{u}_{12}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
\mathbf{u}_{13}	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
\mathbf{u}_{14}	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
\mathbf{u}_{23}	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1
\mathbf{u}_{24}	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
\mathbf{u}_{34}	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1
\mathbf{u}_{123}	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
\mathbf{u}_{124}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
\mathbf{u}_{134}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
\mathbf{u}_{234}	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
\mathbf{u}_{1234}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Wybrane kody liniowe

Kod Reeda-Mullera cd..

Macierz generująca i kontrolna kodów RM jest złożona z następujących wektorów bazowych bazy \mathbf{B}_m :

$$\mathbf{G} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \\ \mathbf{u}_{12} \\ \mathbf{u}_{13} \\ \vdots \\ \mathbf{u}_{i_1 i_2 \dots i_l} \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_m \end{bmatrix}$$

← Wszystkie $\mathbf{u}_{i_1 i_2 \dots i_l}$ dla $l \leq \lambda$.

KONIEC

Dziękuję za uwagę