Lineární algebra : Lineární kódy

(5. přednáška)

Daniel Dombek, Luděk Kleprlík,
Karel Klouda

daniel.dombek@fit.cvut.cz, ludek.kleprlik@fit.cvut.cz,
karel.klouda@fit.cvut.cz

Katedra aplikované matematiky
Fakulta informačních technologií

České vysoké učení technické v Praze

LS 2021/2022

vytvořeno: 15. února 2022, 13:47

Hlavní body

Abeceda a kódování

- V nejobecnějším slova smyslu je kódování předpis pro přepisování slov na jiná slova.
- Slova (řetězce) jsou konečné posloupnosti písmen (znaků) z abecedy. **Délkou slova** označujeme počet písmen, ze kterých se toto slovo skládá.
- **Abeceda** je jakákoli konečná neprázdná množina.
  Pro abecedu $\mathcal{A}$ zavádíme tyto značky: $\mathcal{A}^+$ je množina (konečných) slov nad touto abecedou, $\mathcal{A}^n$ je množina všech slov nad touto abecedou, které mají délku n.

Definice

*Mějme dvě abecedy $\mathcal{A}$ a $\mathcal{B}$. Každé zobrazení $\kappa : \mathcal{A} \to \mathcal{B}^+$ nazýváme **kódováním**. Obor hodnot tohoto zobrazení H_κ nazýváme **kód**, označujeme jej K a libovolné slovo z $K = H_\kappa$ nazýváme **kódové slovo**.*
Zobrazení přirozeně rozšiřujeme na slova z $\mathcal{A}^+$ takto: buď $u = u_1 u_2 \dots u_n \in \mathcal{A}^n$ slovo délky n s písmeny $u_i \in \mathcal{A}$. Potom

$$\kappa(u) = \kappa(u_1)\kappa(u_2)\dots\kappa(u_n).$$

Příklady kódování

- Buď $\mathcal{A} = \{a, b, c\}$ a $\mathcal{B} = \{0, 1\}$. Definujeme následující kód:

$$\kappa(a) = 0, \quad \kappa(b) = 10, \quad \kappa(c) = 11.$$

  Dle předchozí definice potom platí pro $abccaa \in \mathcal{A}^6$

$$\kappa(abccaa) = \kappa(a)\kappa(b)\kappa(c)\kappa(c)\kappa(a)\kappa(a) = 0\,10\,11\,11\,0\,0\,.$$

- Jako abecedu můžeme vzít např. i všechny prvky $\{0, 1\}^3$ (všechna třípísmenná binární slova) a definovat kódování $\kappa(000) = 0000$, $\kappa(001) = 0011$, $\kappa(010) = 0101$, $\kappa(011) = 0110$, $\kappa(100) = 1001$, $\kappa(101) = 1010$, $\kappa(110) = 1100$, $\kappa(111) = 1111$. Jedná se o tzv. paritní kód (proč to?).

Předchozí kódování lze považovat za zobrazení z vektorového prostoru $\mathbb{Z}_2^3$ do vektorového prostoru $\mathbb{Z}_2^4$. Jelikož je $\mathbb{Z}_p$ konečná množina, je to vlastně i abeceda a jelikož slova jsou vlastně uspořádané $n$tice prvků z abecedy, budeme často **ztotožňovat vektor ze** $\mathbb{Z}_p^n$ a **slovo ze** $(\mathbb{Z}_p)^n$.

K čemu je kódování

- Přepis z „lidské abecedy“ do „počítačové“ binární abecedy: UTF8, ASCII, cp1250, ...
- Přepis slov na jiná takovým způsobem, aby byl výsledný „text“ co nejkratší (tzv. komprese).
- Přepis slov takovým způsobem, že z výsledného textu budeme schopni přečíst původní text, i když byl částečně změněn (došlo k chybám). Příp. budeme alespoň schopni poznat, že k nějakým změnám došlo.

Kódy s vlastností z posledního bodu se budeme zabývat (tzv. samoopravovací kódy / error correcting codes).

Objevování chyb: paritní kód

Jedinou chybu, kterou budeme uvažovat, je přepis jednoho písmene na jiné (typicky 0 na 1 a 1 na 0). V (rozsáhlé) teorii kódování se ale samozřejmě uvažují i jiné chyby.

- Použijeme paritní kód z předpředchozího slajdu k zakódování textu

$$001011000.$$

- Dostaneme

$$\kappa(001)\kappa(011)\kappa(000) = 0011\,0110\,0000.$$

- Předpokládejme, že jsme text poslali po síti, nebo jsme jej uložili na nějaké médium, a došlo ke třem chybám (přepisům):

$$001\mathbf{0}\,\mathbf{1}111\,0000$$

- Když se na text podíváme, vidíme, že první slovo 0010 ze $\mathbb{Z}_2^4$ není kódové slovo a tedy muselo dojít k chybě. Druhé slovo je kódové slovo a tedy nejsme schopni poznat, že došlo k chybě! Třetí slovo je kódové slovo, a tak jej také budeme považovat za správné.

Opravování chyb: opakovací kód

- Paritní kód tzv. **objevuje 1-chyby**: když dojde k jedné chybě, poznáme to, nevíme ale, kde ta chyba je ani jak vypadalo původní slovo (tomu budeme říkat, že neopravuje chyby).
- Definujeme **opakovací kód** např. opět na abecedě $\mathbb{Z}_2^3$:

$$\kappa(u) = uuu \in \mathbb{Z}_2^9,$$

  např. $\kappa(001) = 001001001$, tj. obraz slova je jeho trojnásobné zřetězení.
- Zakódujeme opět text 001011000:

$$001001001\,011011011\,000000000.$$

- Předpokládejme tři chyby:

$$001001001\,011\mathbf{0}10\mathbf{0}1\,00\mathbf{1}000000.$$

- První slovo je kódové, považujeme je tedy za bezchybné. U druhého si všimneme, že kdyby první trojice byla 001, dostaneme kódové slovo, a proto předpokládáme, že došlo k jedné chybě ve druhém bitu a chybně opravíme slovo na 001. Třetí slovo podobně správně opravíme na 000.

Hammingova vzdálenost

- Proč paritní kód byl s to objevit jednu chybu a dvě již ne?
  - Změním-li v jakémkoli kódovém slově jeden znak, nedostanu jiné kódové slovo, a tak poznám, že k chybě došlo.
  - Když změním dva znaky, dostanu jiné kódové slovo, a tak žádnou chybu nedetekuji.
  - Odpověď tedy je: **Protože dvě různá kódová slova jsou od sebe vzdálená alespoň „dvě chyby“.**

Definice

Pro dvě slova $u = u_1 u_2 \dots u_n$ *a* $v = v_1 v_2 \dots v_n$ *stejné délky* n *a nad stejnou abecedou definujeme* ***Hammingovu vzdálenost*** *jako*

$$d(u, v) = \text{ počet indexů } i \in \hat{n} \text{ takových, že } u_i \neq v_i.$$

Hammingova vzdálenost $d(u, v)$ vlastně vyjadřuje, kolik nejméně *chyb* musím udělat, abych ze slova u vyrobil slovo v.

Kód objevuje t chyb

Definice

*Pro kód K definujeme a značíme **minimální vzdálenost kódu** jako*

$$\mu(K) = \min\{d(u, v) \mid u, v \in K, u \neq v\}.$$

- Minimální vzdálenost paritního kódu je 2,
- Pro opakovací kód K^ℓ (kde vysílané slovo zopakujeme ℓkrát) je $\mu(K^\ell) = \ell$.

Definice

*Řekneme, že kód K **objevuje** t **chyb**, jestliže $\mu(K) > t$.*

Myšlenka: když v kódovém slově uděláme $t < \mu(K)$ chyb, nedostaneme nikdy jiné kódové slovo, a tak budeme vědět, že se stala chyba a přijaté slovo není to odeslané.

Kód opravuje chyby

Uvažujme opakovací kód K^3 (vysílané slovo opakujeme třikrát):

- Jelikož $\mu(K^3) = 3$, tento kód objevuje 2 chyby.
- Když se ale stane jenom jedna chyba, jsme dokonce schopni určit, jaké kódové slovo bylo na vstupu.
- Proč? Protože změním-li v jakémkoli kódovém slově jakýkoli znak, výsledné slovo bude od tohoto kódového slova vzdáleno 1 a od ostatních alespoň 2.

Definice

*Řekneme, že kód K **opravuje** t **chyb**, jestliže $\mu(K) > 2t$.*

Dohromady máme: Je-li $\mu(K) = n$, potom kód K objevuje $n - 1$ (a menší) chyby a opravuje $\lfloor \frac{n-1}{2} \rfloor$ (a menší) chyby.

Informační a kontrolní znaky

- Proč tedy nepoužívat K^ℓ pro nějaké velké ℓ? Protože jenom každý ℓ. znak „nese informaci“!
- Naopak u paritního kódu jenom jeden znak informaci nenese!

Definice

*Řekneme, že kód $K \subseteq \mathcal{A}^n$ (tj. předpokládáme, že kódová slova mají stejnou délku!) má k **informačních znaků** a $n - k$ **kontrolních znaků**, jestliže existuje bijekce $\varphi : \mathcal{A}^k \to K$.*
Lze-li navíc tuto bijekci zvolit tak, že pro každé $u \in \mathcal{A}^k$ existuje $v \in \mathcal{A}^{n-k}$ takové, že

$$\varphi(u) = uv \in K,$$

*říkáme, že K je **systematický kód**.*

Informační a kontrolní znaky: příklady

- Paritní kód $K \subseteq \mathbb{Z}_2^4$ má 3 informační a 1 kontrolní znak. Hledanou bijekci navíc můžeme volit tak, že pro $u_1 u_2 u_3 \in \mathbb{Z}_2^3$

$$\varphi(u_1 u_2 u_3) = u_1 u_2 u_3 v, \quad \text{kde } v = u_1 + u_2 + u_3 \pmod 2.$$

  Paritní kód je tedy i systematický.

- Opakovací kód $K^3 \subseteq Z_2^9$ (tří bitové slovo zopakujeme třikrát) má 3 informační a 6 kontrolních znaků. Hledanou bijekci navíc můžeme volit tak, že pro $u \in \mathbb{Z}_2^3$

$$\varphi(u) = uuu.$$

  Opakovací kód je tedy také systematický.

Systematické kódy a minimální vzdálenost

Věta (Singletonův odhad)

Mějme kód $K \subseteq \mathcal{B}^n$, potom

$$\#K \leq (\#\mathcal{B})^{n-\mu(K)+1},$$

kde $\#$ značí počet prvků v množině.
Speciálně: je-li $K \subseteq \mathcal{B}^n$ kód s k informačními znaky, platí pro jeho minimální vzdálenost

$$\mu(K) \leq n - k + 1.$$

Kódy: shrnutí

Hledáme tedy kód $K \subseteq \mathcal{A}^n$ s násl. vlastnostmi:

- Aby měl co nejvíce informačních znaků,
- a zároveň co nejvyšší minimální vzdálenost $\mu(K)$ (kódové slova byla v $\mathcal{A}^n$ „rozprostřená“ rovnoměrně).
- Mělo by být snadné kódovat (k vysílanému slovu najít příslušné kódové slovo),
- a zároveň i „dekódovat“:
  - k přijatému (bezchybnému) kódovému slovu najít původní vysílané slovo,
  - ke kódovému slovu s chybou najít nejbližší kódové slovo (existuje-li právě jedno).

Jednoduchý „framework“, který výše popsané umožňuje, jsou **Lineární kódy**

Hlavní body

Definice

*Buď $\mathcal{A}$ konečné těleso, K nazveme **lineární** (n, k)-**kód**, jestliže je K podprostor $\mathcal{A}^n$ dimenze k.*
*Nechť $k \in \{1, 2, \ldots, n-1\}$. Matici $G_K \in \mathcal{A}^{k,n}$, jejíž řádky tvoří bázi K, nazýváme **generující maticí** K, matici $H_K \in \mathcal{A}^{n-k,n}$ takovou, že K je množina řešení soustavy $H_K \mathbb{x} = \theta$, nazýváme **kontrolní maticí** K.*

Paritní kód $K \subseteq \mathbb{Z}_2^4$ je lineární $(4, 3)$-kód. Jeho kontrolní matice je

$$H_K = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

a generující je např.

$$G_K = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Lineární kódy: základní vlastnosti

Několik pozorování o lineárním (n, k)-kódu K:

- Lineární (n, k)-kódy pro $k \in \{0, n\}$ jsou triviální (rozmyslete si, jak by vypadaly H_K, G_K).
- Jestli je $u \in \mathcal{A}^n$ kódové slovo poznám snadno pomocí kontrolní matice, neboť

$$u \in K \Leftrightarrow H_K \cdot u = \theta.$$

- Platí $H_K G_K^T = \Theta$ a podobně $G_K H_K^T = \Theta$, jedna matice tedy určuje tu druhou!
- Je-li $\mathcal{A} = \mathbb{Z}_p$, je počet kódových slov v K roven p^k
- Každému prvku $(\alpha_1, \ldots, \alpha_k) \in \mathcal{A}^k$ umím přiřadit jednoznačně kódové slovo jako lineární kombinaci báze K s koeficienty $(\alpha_1, \ldots, \alpha_k)$.
- Kódové slovo z předch. bodu je rovno výsledku násobení matic

$$\left(\begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_k \end{pmatrix} \cdot G_K\right)^T.$$

Systematické lineární kódy

Věta

Lineární (n, k)-kód má k informačních a $n - k$ kontrolních bitů. Navíc platí, že je systematický právě tehdy, když generující matici ***lze volit*** *ve tvaru*

$$G_K = \begin{pmatrix} \mathbb{E}_k & \mathbb{A} \end{pmatrix},$$

kde $\mathbb{E}_k$ je jednotková matice typu $k \times k$ a $\mathbb{A}$ je nějaká matice typu $k \times n - k$.

Objevování/opravování chyb (1/2)

Definice

Buď $\mathcal{A}$ konečné těleso. Pro $u = u_1 u_2 \dots u_n \in \mathcal{A}^n$ definujeme ***Hammingovu váhu*** *jako*

$$\|u\| = \text{ počet } i \in \hat{n} \text{ takových, že } u_i \neq 0,$$

tj. jako počet nenulových znaků ve slově u.

Pozorování:

- Pro lib. $u, v \in \mathcal{A}^n$ platí
$$d(u, v) = \|u - v\|.$$
- Z toho pro lin. kód K plyne
$$\mu(K) = \min\{\|u\| \mid u \in K, u \neq \theta\},$$
tedy minimální vzdálenost kódu je rovna minimální váze nenulového kódového slova.

Objevování/opravování chyb (1/2)

Věta

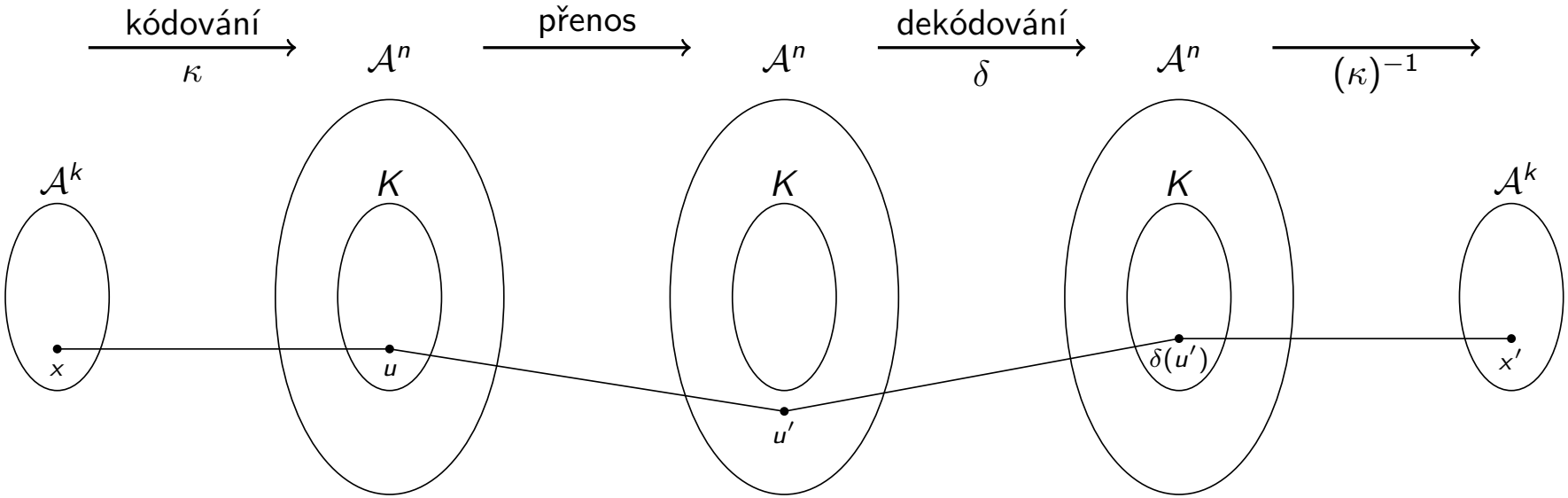
Lineární kód K objevuje, resp. opravuje t-chyby právě tehdy, když je soubor ***libovolných*** *t, resp. $2t$ sloupců v jeho kontrolní matici LN.*

Hlavní body

1. Kódování: základní pojmy a příklady
2. Lineární kódy
3. Dekódování

Dekódování: o co nám jde

- Předpokládejme, že chceme poslat slovo $x \in \mathcal{A}^k$.
- Zakódujeme jej na slovo $\kappa(x) = u \in K$, kde $K \subseteq \mathcal{A}^n$ je lineární (n, k)-kód.
- Slovo u pošleme a přijmeme slovo u', které se od u může lišit o chybu ϵ (tj. $u' = u + \epsilon$).
- Dekódování pro nás bude zobrazení δ, které slovu u' přiřadí nějaké kódové slovo (ideálně odeslané kódové slovo u).



Dekódování: definice

Definice

Buď K lineární kód. Dekódování je libovolné zobrazení $\delta : \mathcal{A}^n \to K$ takové, že pro každé kódové slovo $\alpha \in K$ platí

$$\delta(\alpha) = \alpha.$$

Pozorování: Pokud lineární kód K opravuje t-chyby a definujeme

$$\delta(u') := \text{„nejbližší kódové slovo k } u'\text{“},$$

potom pro libovolné slovo $\alpha \in K$ a chybu $\epsilon \in \mathcal{A}^n$, $\|\epsilon\| \leq t$ platí

$$\delta(\alpha + \epsilon) = \alpha.$$

Příklad dekódování (1 ze 1)

Jak definovat dekódování, které libovolnému u' vrací „nejbližší kódové slov“ si ukážeme na příkladu lineárního $(5, 2)$-kódu K s generující maticí

$$G_K = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Čtyři kódová slova: 00000,01101,11010,10111. Vidíme, že $\mu(K) = 3$ (kód objevuje 2 chyby a opravuje 1 chybu)

- Buď $u' = 11100$. Hledáme $u \in K$ tak, že $d(u, u')$ je nejmenší.
- Najdeme $u = 11010$, chyba $u' - u = 00110$ a vzdálenost $d(u, u') = 2$, „nejmenší chybě“ 00110 budeme říkat **pivot slova** u' a značit $\pi_{u'}$.
- Všimněme si, že slova
  $00110 = 00000 + 00110, 01011 = 01101 + 00110, 10111 + 00110 = 10001$
  mají stejný pivot.

Příklad dekódování (1 ze 2)

Najdeme-li ke každému slovu ze $\mathbb{Z}_2^5$ nejbližší kódové slovo (je-li jich více, vyberu si) a pivot, můžeme je zapsat do tabulky, kde první řádek jsou kódová slova, první sloupec pivoty. Slovo u' takové, že $u' = u + \pi_{u'}, u \in K$, zapíšeme do sloupce pod u a do řádku vedle $\pi_{u'}$:

	pivot			
K:	00000	01101	11010	10111
	00001	01100	11011	10110
	00010	01111	11000	10101
	00100	01001	11110	10011
	01000	00101	10010	11111
	10000	11101	01010	00111
	00110	01011	11100	10001
	00011	01110	11001	10100

Tabulka pro dekódování.

Standardní dekódování

Obecná konstrukce: Buď K lineární (n, k)-kód nad abecedou $\mathcal{A}$ velikosti q a uvažujme variety se zaměřením K (tj. ve tvaru $x + K$ pro nějaké $x \in \mathcal{A}^n$).

- Každé slovo x z $\mathcal{A}^n$ leží v nějaké takové varietě (třeba v $x + K$).
- Pokud W_1 a W_2 jsou variety se zaměřením K a jejich průnik $W_1 \cap W_2$ je neprázdný, tak potom $W_1 = W_2$.
- Každá varieta se zaměřením K má $(\#\mathcal{A})^k$ slov.

Proto existují variety $W_i \in \mathcal{A}^n$ se zaměřením K, kde $i \in \{1, \ldots, q^{n-k}\}$, takové, že

1. $\mathcal{A}^n = \bigcup_{i=1}^{q^{n-k}} W_i$,
2. $W_i \cap W_j = \emptyset$, pro $i \neq j$.

Jinými slovy, prostor $\mathcal{A}^n$ lze disjunktně rozložit na variety se zaměřením K. Takových variet je q^{n-k}.

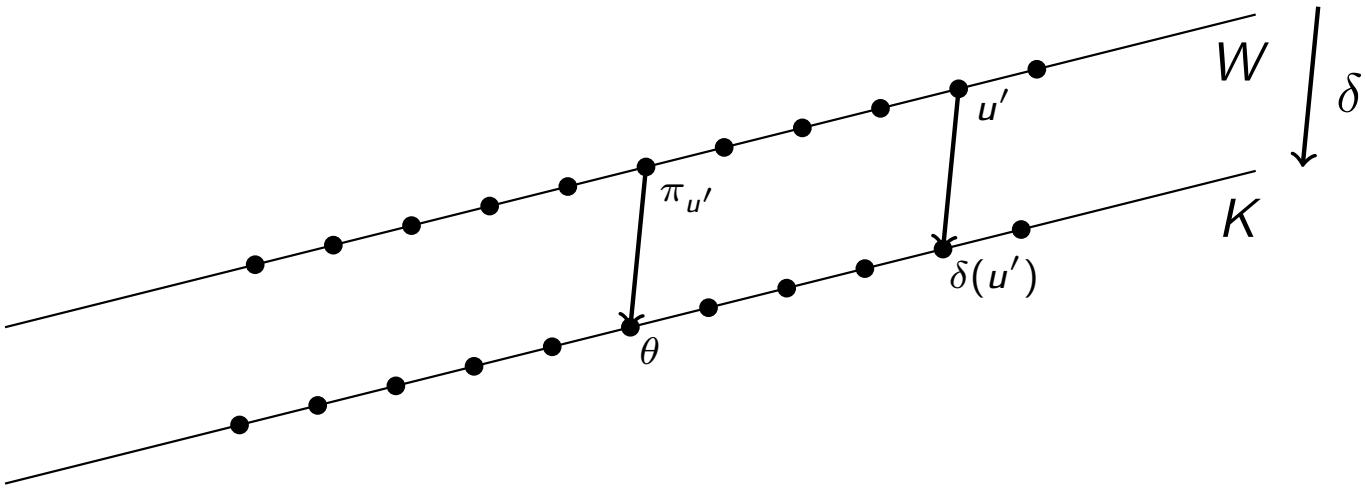
Standardní dekódování

Definice

Buď K lineární (n, k)-kód. V každé varietě se zaměřením K zvolíme ***pivot****, tj. slovo nejmenší Hammingovy váhy. Označme $\pi_{u'}$ pivot variety, ve které leží u' (tj. variety $u' + K$).*
Potom pro slova $u' \in \mathcal{A}^n$ definujeme ***standardní dekódování*** *δ vztahem:*

$$\delta(u') := u' - \pi_{u'}.$$

Schéma pro dekódování na varietě $W = x + K$:



Standardní dekódování

Tabulka pro dekódování:

	pivot						
K:	θ	α_2	α_3	$\dots$	α	$\dots$	α_p
	β_1	β_2	β_3	$\dots$		$\dots$	β_p
	γ_1	γ_2	γ_3	$\dots$		$\dots$	γ_p
	$\vdots$	$\vdots$	$\vdots$				$\vdots$
	σ			$\dots$	τ	$\dots$	
	$\vdots$	$\vdots$	$\vdots$				$\vdots$

- V prvním sloupci jsou pivoty,
- $\beta_i = \alpha_i + \beta_1$, $\gamma_i = \alpha_i + \gamma_1$,
- přijaté τ najdeme v tabulce a dekódujeme jako slovo α nacházející se ve stejném sloupci v prvním řádku,
- $\delta(\tau) = \alpha = \tau - \sigma = \tau - \pi_\tau$,
- tabulka má $p = q^k$ sloupců a q^{n-k} řádků, kde $q = \#\mathcal{A}$.

Syndromy

Definice

Buď K lineární kód. ***Syndromem*** *slova $u' \in \mathcal{A}^n$ nazýváme slovo $\sigma_{u'} := H_K \cdot u'$.*

Pozorování:

1. $\sigma_{u'} = \theta \iff u' \in K$
2. $\sigma_{u'} = \sigma_{v'} \iff u' + K = v' + K$
3. Buď $u \in K$ a $u' = u + \epsilon$, pak $\sigma_{u'} = \sigma_\epsilon$. Tedy syndrom přijatého slova u' splývá se syndromem příslušného chybového slova ϵ.

Syndromová tabulka pro standardní dekódování:

pivot	$\epsilon_1 = \theta$	ϵ_2	$\dots$	ϵ_r
syndrom	$\sigma_1 = \theta$	σ_2	$\dots$	σ_r

Dekodér přijme u', spočítá syndrom $\sigma_{u'}$, najde $i \in \hat{r}$ tak, že $\sigma_i = \sigma_{u'}$ a dekóduje

$$\delta(u') = u' - \epsilon_i.$$

Příklad

Uvažujme ještě jednou předchozí příklad s binárním lineárním $(5, 2)$-kódem K definovaným generující maticí

$$G_K = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Spočítáme

$$H_K = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Syndromová tabulka:

pivot	00000	00001	00010	00100	01000	10000	00110	00011
syndrom	000	110	011	100	010	001	111	101

Přijme-li dekodér např. slovo $u' = 11111$, spočítá $\sigma_{u'} = H_K \cdot 11111 = 010$.
V pátém sloupci syndromové tabulky najde $\epsilon = 01000$ a dekóduje:

$$\delta(11111) = 11111 - 01000 = 10111.$$