Diskrétní matematika – 7. týden
Lineární kódy

Lukáš Vokřínek

Masarykova univerzita
Fakulta informatiky

podzim 2024

Obsah přednášky

Doporučené zdroje

- Jan Slovák, Martin Panák, Michal Bulant
  Matematika drsně a svižně, *e*-text na
  `www.math.muni.cz/Matematika_drsne_svizne`.

Doporučené zdroje

- Jan Slovák, Martin Panák, Michal Bulant **Matematika drsně a svižně**, *e*-text na `www.math.muni.cz/Matematika_drsne_svizne`.
- W. J. Gilbert, W. K. Nicholson, Modern algebra with applications, 2nd ed. John Wiley and Sons (Pure and applied mathematics) ISBN 0-471-41451-4

Plán přednášky

Při přenosu informace zpravidla dochází k její deformaci. Budeme pro jednoduchost pracovat s modelem, kdy jednotlivé částečky informace jsou buď nuly nebo jedničky (tj. prvky v $\mathbb{Z}_2$) a přenášíme slova o k bitech.

Při přenosu informace zpravidla dochází k její deformaci. Budeme pro jednoduchost pracovat s modelem, kdy jednotlivé částečky informace jsou buď nuly nebo jedničky (tj. prvky v $\mathbb{Z}_2$) a přenášíme slova o k bitech.
Přenosové chyby chceme

1. rozpoznávat
2. opravovat

a za tím účelem přidáváme dodatečných $n - k$ bitů informace pro pevně zvolené $n > k$. Mluvíme pak o (n, k)-*kódu*.

Při přenosu informace zpravidla dochází k její deformaci. Budeme pro jednoduchost pracovat s modelem, kdy jednotlivé částečky informace jsou buď nuly nebo jedničky (tj. prvky v $\mathbb{Z}_2$) a přenášíme slova o k bitech.
Přenosové chyby chceme

1. rozpoznávat
2. opravovat

a za tím účelem přidáváme dodatečných $n - k$ bitů informace pro pevně zvolené $n > k$. Mluvíme pak o (n, k)-*kódu*.
Všech slov o k bitech je 2^k a každé z nich má jednoznačně určovat jedno **kódové slovo** z 2^n možných. Máme tedy ještě

$$2^n - 2^k = 2^k(2^{n-k} - 1)$$

slov, které jsou chybové. Lze tedy tušit, že pro veliké k nám i malý počet přidaných bitů dává hodně redundantní informace.

Úplně jednoduchým příkladem je **kód kontrolující paritu**. Kódové slovo o $k + 1$ bitech je určené tak, aby přidáním prvního bitu byl zaručen sudý počet jedniček ve slově.

Úplně jednoduchým příkladem je **kód kontrolující paritu**. Kódové slovo o $k + 1$ bitech je určené tak, aby přidáním prvního bitu byl zaručen sudý počet jedniček ve slově.
Pokud při přenosu dojde k lichému počtu chyb, přijdeme na to.
Dvě různá kódová slova se při tomto kódu vždy liší alespoň ve dvou pozicích, chybové slovo se ale od dvou různých kódových slov liší pouze v pozici jedné. Nemůžeme proto umět chyby opravovat ani kdybychom věděli, že došlo k právě jedné.

Úplně jednoduchým příkladem je **kód kontrolující paritu**. Kódové slovo o $k + 1$ bitech je určené tak, aby přidáním prvního bitu byl zaručen sudý počet jedniček ve slově.
Pokud při přenosu dojde k lichému počtu chyb, přijdeme na to.
Dvě různá kódová slova se při tomto kódu vždy liší alespoň ve dvou pozicích, chybové slovo se ale od dvou různých kódových slov liší pouze v pozici jedné. Nemůžeme proto umět chyby opravovat ani kdybychom věděli, že došlo k právě jedné.
Navíc neumíme detekovat tak obvyklé chyby, jako je záměna dvou sousedních hodnot ve slově.

Definice

Hammingova vzdálenost dvou slov je rovna počtu bitů, ve kterých se liší.

Definice

Hammingova vzdálenost dvou slov je rovna počtu bitů, ve kterých se liší.

Věta

1. *Kód odhaluje r a méně chyb právě, když je Hammingova vzdálenost kódových slov alespoň $r + 1$.*
2. *Kód opravuje r a méně chyb právě, když je Hammingova vzdálenost kódových slov alespoň $2r + 1$.*

Plán přednášky

Definice

Lineární kód je injektivní lineární zobrazení $g : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^n$. Matice G typu n/k reprezentující toto zobrazení v standardních bazích se nazývá generující **matice kódu**.

Definice

Lineární kód je injektivní lineární zobrazení $g : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^n$. Matice G typu n/k reprezentující toto zobrazení v standardních bazích se nazývá generující **matice kódu**.

Pro každé slovo u je

$$v = G \cdot u$$

příslušné kódové slovo. Pro jednoduchost budeme předpokládat, že kód přidává dodatečnou informaci a pro konkrétnost tedy, že v má blokový tvar $v = \binom{Pu}{u}$, kde Pu je ona dodatečná informace a u je původní vektor. Proto matice G bude mít blokový tvar

$$G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix}.$$

Věta

Je-li $g : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^n$ *lineární kód s (blokově zapsanou) maticí*

$$G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix},$$

potom zobrazení $h : (\mathbb{Z}_2)^n \to (\mathbb{Z}_2)^{n-k}$ *s maticí*

$$H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix}$$

má následující vlastnost: slovo v *je kódové, právě když* $H \cdot v = 0$.

Věta

Je-li $g : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^n$ *lineární kód s (blokově zapsanou) maticí*

$$G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix},$$

potom zobrazení $h : (\mathbb{Z}_2)^n \to (\mathbb{Z}_2)^{n-k}$ *s maticí*

$$H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix}$$

má následující vlastnost: slovo v *je kódové, právě když* $H \cdot v = 0$.

Důkaz.

Slovo $v = \binom{x}{y}$ je kódové, právě když $v = Gy = \binom{Py}{y}$, protože v takovém případě jsme schopni původní slovo y dekódovat z informačních bitů. Slovo v je tedy kódové, právě když $x = Py$, což je přesně podmínka $H\binom{x}{y} = 0$. □

Matici H z věty se říká *matice kontroly parity* přílušného (n, k)–kódu, součinu Hv říkáme *syndrom* slova v.

Matici H z věty se říká *matice kontroly parity* přílušného (n, k)–kódu, součinu Hv říkáme *syndrom* slova v.

Význam syndromu

Pokud $Hv \neq 0$, hledáme co nejbližší kódové slovo, tj. co nejmenší e tak, aby $H(v + e) = 0$. Přitom přičtení jedničky na i-tém místě způsobí změnu hodnoty přesně o i-tý sloupec kontrolní matice H. Chceme tedy syndrom Hv vynulovat přičtením co nejmenšího počtu sloupců matice H.

Matici H z věty se říká *matice kontroly parity* přílušného (n, k)–kódu, součinu Hv říkáme *syndrom* slova v.

Význam syndromu

Pokud $Hv \neq 0$, hledáme co nejbližší kódové slovo, tj. co nejmenší e tak, aby $H(v + e) = 0$. Přitom přičtení jedničky na i-tém místě způsobí změnu hodnoty přesně o i-tý sloupec kontrolní matice H. Chceme tedy syndrom Hv vynulovat přičtením co nejmenšího počtu sloupců matice H.
V levé submatici máme právě sloupce s jednou jedničkou. Kdybychom využívali pouze tyto sloupce (opravovali pouze kontrolní bity), potřebujeme opravit přesně tolik bitů, kolik obsahuje syndrom Hv jedniček, navíc přesně na stejných pozicích.

Naopak, nechť h je lineární zobrazení, jehož matice je tvaru $H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix}$. Pak můžeme sestrojit lineární kód s maticí $G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix}$ a podle věty pak H bude matice kontroly parity. Jsme tedy schopni lineární kód zadat jeho (lineární) kontrolou parity.

Naopak, nechť h je lineární zobrazení, jehož matice je tvaru $H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix}$. Pak můžeme sestrojit lineární kód s maticí $G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix}$ a podle věty pak H bude matice kontroly parity. Jsme tedy schopni lineární kód zadat jeho (lineární) kontrolou parity.

Příklad

Jednoduše to lze ilustrovat na klasické kontrole parity, kde $h(x_0, \ldots, x_k) = x_0 + \cdots + x_k$ je zjevně lineární s maticí

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix},$$

kýženého tvaru. Nemusíme tedy specifikovat matici kódu, tu lze z matice kontroly parity odvodit.

Naopak, nechť h je lineární zobrazení, jehož matice je tvaru $H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix}$. Pak můžeme sestrojit lineární kód s maticí $G = \begin{pmatrix} P \\ \mathbb{I}_k \end{pmatrix}$ a podle věty pak H bude matice kontroly parity. Jsme tedy schopni lineární kód zadat jeho (lineární) kontrolou parity.

Příklad

Jednoduše to lze ilustrovat na klasické kontrole parity, kde $h(x_0, \ldots, x_k) = x_0 + \cdots + x_k$ je zjevně lineární s maticí

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix},$$

kýženého tvaru. Nemusíme tedy specifikovat matici kódu, tu lze z matice kontroly parity odvodit.

V další části uvedeme konstrukci polynomiálních kódů skrze jejich matici kontroly parity.

Plán přednášky

Jak konstruovat kódová slova, abychom je snadno rozpoznali? Kontrolu parity jsme už viděli, další triviální možnost je prosté opakování bitů – např. $(3, 1)$–kód bere jednotlivé bity a posílá je třikrát po sobě.

Jak konstruovat kódová slova, abychom je snadno rozpoznali?
Kontrolu parity jsme už viděli, další triviální možnost je prosté opakování bitů – např. $(3, 1)$–kód bere jednotlivé bity a posílá je třikrát po sobě.
Docela systematickou cestou je využití dělitelnosti polynomů.
Zpráva $b_0 b_1 \dots b_{k-1}$ je reprezentována jako polynom
$m(x) = b_0 + b_1 x + \cdots + b_{k-1} x^{k-1} \in \mathbb{Z}_2[x]$.

Jak konstruovat kódová slova, abychom je snadno rozpoznali?
Kontrolu parity jsme už viděli, další triviální možnost je prosté opakování bitů – např. $(3, 1)$–kód bere jednotlivé bity a posílá je třikrát po sobě.
Docela systematickou cestou je využití dělitelnosti polynomů.
Zpráva $b_0 b_1 \dots b_{k-1}$ je reprezentována jako polynom $m(x) = b_0 + b_1 x + \cdots + b_{k-1} x^{k-1} \in \mathbb{Z}_2[x]$.

Definice

Nechť $p(x) = a_0 + \cdots + a_{n-k} x^{n-k} \in \mathbb{Z}_2[x]$ je polynom s $a_0 = 1$, $a_{n-k} = 1$. *Polynomiální kód generovaný polynomem* $p(x)$ je lineární (n, k)–kód, jehož slova jsou polynomy stupně menšího než n dělitelné $p(x)$ a jehož kontrola parity je lineární zobrazení h posílající polynom na jeho zbytek po dělení $p(x)$.

Zobrazení h je opravdu lineární (zbytek součtu je součet zbytků). Polynomy stupně menšího než $n - k$ jsou automaticky zbytkem po dělení $p(x)$, takže je na nich h identita a matice H je tedy tvaru

$$H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix},$$

jak je potřeba.

Zobrazení h je opravdu lineární (zbytek součtu je součet zbytků). Polynomy stupně menšího než $n - k$ jsou automaticky zbytkem po dělení $p(x)$, takže je na nich h identita a matice H je tedy tvaru

$$H = \begin{pmatrix} \mathbb{I}_{n-k} & P \end{pmatrix},$$

jak je potřeba.

Sestavení matic

Nastíníme nyní, jak matici P sestavit. Její první sloupec je zbytek x^{n-k} a sestává se tedy z koeficientů $p(x)$ stupně menšího než $n - k$. Další sloupec je zbytkem $x^{n-k+1} = x \cdot x^{n-k}$ a získáme jej tedy z předchozího sloupce vynásobením x, přičemž případný výskyt x^{n-k} nahradíme jeho zbytkem, tedy prvním sloupcem P. Konkrétně tedy sloupec posuneme dolů a případná jednička se při přetečení nahradí přičtením prvního sloupce. Takto postupujeme i pro další sloupce – posouváme předchozí, při přetečení přičítáme první!

Příklad

1. Polynom $p(x) = 1 + x$ generuje $(n, n-1)$–kód kontroly parity pro všechna $n \geq 3$.
2. Polynom $p(x) = 1 + x + x^2$ generuje $(3, 1)$–kód opakování bitů.

První tvrzení plyne z toho, že $1 + x$ dělí polynom $v(x)$ tehdy a jen tehdy, když $v(1) = 0$ a to nastane tehdy, když je ve $v(x)$ sudý počet nenulových koeficientů. Druhé je zřejmé.

Matice příslušná k polynomu $p(x) = 1 + x + x^3$ a jím určenému $(7, 4)$–kódu je

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

Přenos slova $v \in \mathbb{Z}_2[x]$ dopadne příjmem polynomu

$$u(x) = v(x) + e(x)$$

kde $e(x)$ je tzv. **chybový polynom** reprezentující vektor chyby přenosu.

Přenos slova $v \in \mathbb{Z}_2[x]$ dopadne příjmem polynomu

$$u(x) = v(x) + e(x)$$

kde $e(x)$ je tzv. **chybový polynom** reprezentující vektor chyby přenosu.

Analýza

Chyba je rozpoznatelná pouze, když generátor kódu $p(x)$ nedělí $e(x)$. Máme proto zájem o polynomy, které nevystupují jako dělitelé zbytečně často. Připomeňme, že matice kontroly parity obsahuje ve sloupcích zbytky po dělení x^i polynomem $p(x)$. Pokud chceme, aby kód rozpoznal jednoduché chyby, nesmí matice kontroly parity obsahovat žádný nulový sloupec – to totiž odpovídá $p(x) \mid x^i$. Pokud chceme, aby kód rozpoznal dvojité chyby, nesmí matice kontroly parity obsahovat žádný sloupec dvakrát – to totiž odpovídá $p(x) \mid x^i + x^j$. Při počtu řádků $m = n - k$, tak může P obsahovat maximálně $n = 2^m - 1$ sloupců.

Definice

Ireducibilní polynom $p(x) \in \mathbb{Z}_2[x]$ stupně m se nazývá *primitivní*, jestliže $p(x) \nmid (1 + x^\ell)$ pro $\ell < 2^m - 1$, a teprve $p(x) \mid (1 + x^\ell)$ pro $\ell = 2^m - 1$.

Definice

Ireducibilní polynom $p(x) \in \mathbb{Z}_2[x]$ stupně m se nazývá *primitivní*, jestliže $p(x) \nmid (1 + x^\ell)$ pro $\ell < 2^m - 1$, a teprve $p(x) \mid (1 + x^\ell)$ pro $\ell = 2^m - 1$.

Věta

Je-li $p(x)$ primitivní polynom stupně m, pak pro všechna $n \leq 2^m - 1$ rozpoznává příslušný $(n, n - m)$–kód všechny jednoduché a dvojité chyby.

Definice

Ireducibilní polynom $p(x) \in \mathbb{Z}_2[x]$ stupně m se nazývá *primitivní*, jestliže $p(x) \nmid (1 + x^\ell)$ pro $\ell < 2^m - 1$, a teprve $p(x) \mid (1 + x^\ell)$ pro $\ell = 2^m - 1$.

Věta

Je-li $p(x)$ primitivní polynom stupně m, pak pro všechna $n \leq 2^m - 1$ rozpoznává příslušný $(n, n - m)$–kód všechny jednoduché a dvojité chyby.

Důsledek

Je-li $q(x)$ primitivní polynom stupně m, pak pro všechna $n \leq 2^m - 1$ rozpoznává $(n, n - m - 1)$–kód generovaný polynomem $p(x) = q(x)(1 + x)$ všechny dvojité chyby a všechna slova s lichým počtem chyb.

Tabulka dává o informace o výsledcích předchozích dvou vět pro několik polynomů:

Tabulka dává o informace o výsledcích předchozích dvou vět pro několik polynomů:

primitivní polynom	kontrolní bity	délka slova
$1 + x + x^2$	2	3
$1 + x + x^3$	3	7
$1 + x + x^4$	4	15
$1 + x^2 + x^5$	5	31
$1 + x + x^6$	6	63
$1 + x^3 + x^7$	7	127
$1 + x^2 + x^3 + x^4 + x^8$	8	255
$1 + x^4 + x^9$	9	511
$1 + x^3 + x^{10}$	10	1023

Nástroje pro konstrukci primitivních polynomů dává teorie konečných polí. Souvisí s tzv. primitivními prvky v Galoisových polích $G(2^m)$.

Nástroje pro konstrukci primitivních polynomů dává teorie konečných polí. Souvisí s tzv. primitivními prvky v Galoisových polích $G(2^m)$.
Ze stejné teorie lze také dovodit příjemnou realizaci dělení se zbytkem (tj.) ověřování, zda je přijaté slovo kódové, pomocí zpožďovacích registrů. Jde o jednoduchý obvod s tolika prvky, kolik je stupeň polynomu.