Samoopravné kódy

Záverečná práca
Ukázky aplikací matematiky

ŽANETA SEMANIŠINOVÁ

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta
2017

Obsah

1 Úvod

Jedným z najaktuálnejších problémov súčasnosti je prenos informácií. Informácie vždy putujú cez nejaké fyzikálne prostredie, ktoré sa nazýva kanál. Vzhľadom k obrovskému množstvu prenášaných dát, aj pri pokročilých technológiách prenosu dochádza k šumu, ktorý skresľuje prenášanú informáciu. V princípe, v správe dochádza k chybám. Teória kódov sa zaoberá detekovaním a odstraňovaním chýb, ktoré pri prenose informácie kanálom vznikajú. [1]

Princíp teórie kódov je založený na tom, že k pôvodnej správe pripojíme kontrolné znaky, ktoré nemajú žiadnu obsahovú hodnotu pre prijímateľa, ale umožňujú nám overiť, či nebola správa poškodená. Inak povedané, ak prijmeme správu, ktorá svojou štruktúrou nezodpovedá systému kódovania, vieme detekovať chybu a prípadne aj zistiť, aká bola najpravdepodobnejšie pôvodná správa. Pri dobrom kódovaní vieme tak pri malom množstve chýb získať správnu správu bez potreby jej opätovného odoslania.

Už samotná detekcia chýb má v reálnom svete široké spektrum využitia, napríklad v rodných číslach alebo číslach kreditných kariet. Umožňuje odhaliť preklepy alebo falzifikáty. Korekcia chybne odoslaných správ sa používa v mobilnej komunikácii, CD prehrávačoch, čiarových kódoch alebo pri komunikácii s vesmírnymi sondami. [4]

Cieľom teórie kódovania je vymyslieť spôsob ako rýchlo prenášať veľké množstvo informácií s čo najmenej chybami. Zrejme ale nie je možné optimalizovať všetky aspekty tejto komunikácie. Tvorba vhodného kódu sleduje tieto ciele:

1. rýchle zakódovanie a dekódovanie správy
2. jednoduchý prenos správy
3. detekcia a oprava chýb vzniknutých pri prenose
4. maximálny prenos množstva informácie za jednotku času

Kľúčový je pritom zväčša práve tretí aspekt, ktorého optimalizácia väčšinou nie je v súlade so zvyšnými cieľmi, predovšetkým so štvrtým z nich. [3]

Keďže je cieľom preniesť čo najväčšie množstvo informácií (čiže symbolov, ktoré ju obsahujú) a k nim pripojiť čo najmenej kontrolných symbolov, chápeme **prenosovú rýchlosť** informácie pri použití daného kódu ako pomer informačných symbolov ku všetkým.

Teória Clauda Shannona z roku 1948 dosiahla presnejšie výsledky ohľadom toho, aké množstvo informácií s čo najviac opraveným chybami je možné preniesť za jednotku času. Shannon vo svojej teórii odvodil, že pre daný kanál s istou hladinou šumu existuje teoretická maximálna hodnota prenosovej rýchlosti, pri ktorej dokážeme opraviť všetky chyby. Toto číslo sa nazýva **kapacita kanálu** alebo **Shannonov limit**. [8]

Shannon navyše vo svojej teórii dokázal, že pri prenosovej rýchlosti nižšej ako kapacita daného kanálu vždy existuje kód, pri použití ktorého je pravdepodobnosť, že chybu nedokážeme opraviť, tak malá, ako potrebujeme (inak povedané, menšia než ľubovoľné kladné ε). Zaujímavé je tiež, že platí, že pri vyššej prenosovej rýchlosti nie je možné tento stav dosiahnuť. Dôkaz tejto vety však neukazuje žiadny spôsob, ako zostrojiť takto spoľahlivý kód, ktorý je použiteľný v praxi. Ukazuje však, že má zmysel sa ako konštrukciu takéhoto kódu pokúšať. [7][8]

2 Matematický model kódovania

2.1 Predpoklady

Ako je typické pre elektronické komunikačné kanály, za **slovo** budeme považovať sekvenciu núl a jednotiek vopred určenej dĺžky, pričom jeden znak slova nazývame **cifra** alebo **symbol**. **Kódom** budeme rozumieť množinu slov, pričom slová z tejto množiny nazývame **kódovými slovami**. Každé kódové slovo obsahuje dva druhy cifier – **informačné symboly**, ktoré obsahujú správu a **kontrolné symboly**, ktoré nenesú obsah, ale pomáhajú odhaľovať a opravovať chyby.

Budeme pracovať s predpokladom, že odosielateľ prostredníctvom komunikačného kanálu odoslal nejaké kódové slovo, ktoré je postupnosťou núl a jednotiek. Rovnako je na druhom konci kanálu prijatá postupnosť núl a jednotiek rovnakej dĺžky, ktorá sa ale nemusí zhodovať s odosielanou a nutne nepredstavuje kódové slovo.

Pre zjednodušenie tiež očakávame, že je rovnako pravdepodobné, že sa poškodí ktorákoľvek cifra nezávisle od jej hodnoty a umiestnenia, pričom tieto javy na jednotlivých cifrách sú navzájom nezávislé (to nie je veľmi typické pre chyby ako sú škrabance na CD a pod.).

Predpoklady boli prevzaté zo zdroja [1].

2.2 Jednoduché príklady samoopravných kódov

2.2.1 Paritný kód

Jednoduchý spôsob ako pristúpiť k zisťovaniu chýb je jednoducho vyslať jednu cifru navyše, ktorá nebude niesť informáciu, ale upovedomí prijímateľa o poškodení správy. Najzákladnejším kódom s touto vlastnosťou je tzv. **paritný kód**. Funguje na princípe, že sa ku každému slovu pripojí cifra 0 alebo 1 tak, aby obsahovalo vždy párny počet cifier 1. Napríklad správu 0101 vyšleme s kontrolným symbolom ako 01010 a správu 0100 vyšleme ako 01001. Pre slovo dĺžky n bude prvých $n-1$ symbolov predstavovať informačné symboly a posledný symbol bude kontrolný. Prenosová rýchlosť kódu je preto v tomto prípade $\frac{n-1}{n}$. [2]

Tento kód zrejme dokáže odhaliť 1 chybu, pretože ak by bol v prijatom slove bol počet jednotiek nepárny, pri prenose došlo k chybe. Zároveň si môžeme všimnúť, že chybu opraviť nevieme, pretože chyba na každom zo symbolov je rovnako pravdepodobná.

2.2.2 Opakovací kód

Asi najprirodzenejší spôsob, ako odhaliť chybu, poprípade aj nejakú opraviť, je vyslať bloky informácií opakovane (použijeme tzv. **opakovací** resp. ***n*-opakovací kód**, n závisí od počtu opakovaní). Napríklad správu 01 vyšleme ako 0101. Druhá polovica symbolov je kontrolná, rýchlosť prenosu je preto 1/2. Opäť dokážeme rozpoznať, či pri prenose správy došlo k jednej chybe, pretože sa informačná a kontrolná časť slova nebudú zhodovať. Chybný symbol však určiť nevieme. Napríklad, ak by sme prijali správu 0100, je rovnako pravdepodobné, že pôvodná správa bola 0000 ako, že bola 0101. [2]

Situácia sa zmení v prípade, že správu zopakujeme až dvakrát, teda znížime rýchlosť prenosu na 1/3 a vyšleme slovo 010101. Ak by prijímateľ dostal slovo 011101, vie, že došlo k chybe. Zároveň však môže predpokladať (keďže očakávame rozumnú spoľahlivosť kanála), že je pravdepodobnejšie, že chýb bolo čo najmenej. V tomto prípade bolo teda najpravdepodobnejšie odoslané slovo 010101. [2]

2.3 Algebraický prístup k práci so slovami

Pre zjednodušenie práce s niektorými algebraickými operáciami (typicky so sčítaním slov alebo pri práci s maticami) môžeme slová dĺžky n chápať ako vektory vektorového priestoru $\mathbb{Z}_2^n$. Špeciálne rozlišujeme **nulové slovo**, ktoré je vlastne nulovým vektorom o obsahujúcim len cifry 0 na všetkých pozíciách.

2.4 Hammingova vzdialenosť a váha

Pri opravovaní chybných slov na správne, je dôležité, ako veľmi sa potenciálne odosielané slová líšia od prijatého. Definujeme preto pojem **Hammingova vzdialenosť slov** (resp. len **vzdialenosť**) ako počet cifier, v ktorých sa dve slová líšia, značíme $d(v, u)$ pre dve slová rovnakej dĺžky v a u. Pojem **Hammingova váha slova** v (alebo len **váha**), potom vyjadruje vzdialenosť slova v od nulového slova, čo značíme $wt(v)$. Napríklad slovo 010111 má váhu 4 a jeho vzdialenosť od slova 010100 je 2.

Hammingova vzdialenosť predstavuje druh metriky, v tomto prípade je to metrika na slovách. To platí vďaka tomu, že spĺňa $d(v, v) = 0$, $d(u, v) = d(v, u)$ a $d(u, v)+d(v, w) \geq d(u, w)$ pre ľubovoľné slová u, v, w rovnakej dĺžky. Prvé dve vlastnosti platia zrejme z definície, posledná plynie z toho, že ak sa slová u a w líšia v i-tom znaku, potom sa v ňom líšia buď u a v alebo v a w. Preto platí, že ak i-tá pozícia prispieva k hodnote $d(u, w)$, tak prispieva aj k hodnote $d(u, v)$ resp. $d(v, w)$. [2]

Pri identifikovaní, v ktorých cifrách sa slová líšia, sa používa pojem **chybový vzor** (angl. error pattern). Tým je pre dané slová v a w slovo $u = v + w$, ktoré spĺňa, že má cifry 0 na pozíciách, kde sa slová zhodujú a 1 na pozíciách, kde nie. Ekvivalentne platí (z aritmetiky v telese $\mathbb{Z}_2$) $v = w + u$ a $w = v + u$. Všimneme si tiež, že $d(v, w) = wt(u)$. Napríklad chybový vzor pre slová 010111 a 010100 je 000011. [1]

Pri danom kóde je dôležité poznať minimálnu vzdialenosť medzi kódovými slovami. Definujeme preto **minimálnu vzdialenosť kódu** ako minimum zo vzdialeností medzi všetkými dvojicami kódových slov. Napríklad minimálna vzdialenosť paritného kódu alebo 2-opakovacieho kódu je 2, pri 3-opakovacom kóde je to až 3.

2.5 Najpravdepodobnejšie odoslané slovo

Pokiaľ by bol kanál, ktorým informáciu prenášame, dokonale spoľahlivý, nepotrebovali by sme samoopravné kódovanie. Keďže taký komunikačný kanál neexistuje, očakávame, že pravdepodobnosť, že sa odosielaná cifra pri prenose nepoškodí, je určená pravdepodobnosťou p. Môžeme predpokladať, že $1/2 < p < 1$, pretože kanál s $p = 1/2$ je nepoužiteľný pre praktické účely a zvyšné možnosti sa dajú previesť na tento prípad. Hodnotu p je spravidla pre konkrétny kanál náročné určiť, teória kódovania je však na jej presnej hodnote nezávislá.

Vzhľadom k uvedeným predpokladom, uvažujeme, že pravdepodobnosť, že odosielané slovo sa príjme ako nejaké iné slovo, závisí len od toho, v koľkých cifrách sa líšia. Pre dané slová v a w je tento počet $d(v, w) = wt(u)$, kde u je chybový vzor v a w. Pravdepodobnosť, že pri odosielaní slova v sme dostali slovo w, značíme $P(v, w)$, tak môžeme jednoducho vyjadriť ako

$$P(v, w) = p^{n-wt(u)}(1 - p)^{wt(u)}$$

kde n je dĺžka slov v a w.

Nech u, v sú dve kódové slová, w je prijaté slovo, $d_1 = d(u, w)$ a $d_2 = d(v, w)$. Nech $d_1 \geq d_2$. Potom platia nasledujúce úpravy (využívame, že $p > 1 - p$):

$$\left(\frac{p}{1-p}\right)^{d_2-d_1} \leq 1$$

$$p^{n-d_1}(1-p)^{d_1} \leq p^{n-d_2}(1-p)^{d_2}$$

$$P(u, w) \leq P(v, w)$$

Z toho plynie, že najpravdepodobnejšie odosielané slovo je to, ktoré má najnižšiu vzdialenosť k prijatému slovu zo všetkých kódových slov. Odvodenie bolo prevzaté zo zdroja [1].

2.6 Princíp kódovania a dekódovania

V pozícii odosielateľa vychádzame z toho, že máme zoznam všetkých možných správ dĺžky k, ktoré môžeme chcieť kanálom odoslať. Následne v závislosti od použitého kódovania pripojíme k tejto správe nejaké cifry, pričom vznikne slovo dĺžky n, ktoré by malo byť kódovým slovom. Toto slovo pošleme kanálom prijímateľovi.

V pozícii prijímateľa po prijatí slova dĺžky n vyhodnotíme, či je prijaté slovo kódovým slovom. Ak áno, je najpravdepodobnejšie, že slovo nebolo poškodené. V opačnom prípade vieme, že došlo k chybe. Všeobecnejšie, po prijatí slova vyhodnotíme, ktoré kódové slovo má od prijatého slova najnižšiu vzdialenosť. Toto kódové slovo vyhodnotíme (na základe záveru z časti 2.3) ako odoslané slovo. V prípade viacerých rovnako vzdialených slov môžeme podľa potreby rozhodnúť, či požadujeme opätovné odoslanie správy alebo z nich vyberieme náhodne.

Stojí za zdôraznenie, že týmto spôsobom sa nemusíme vždy dopracovať k správnej správe, dopracujeme sa len k najpravdepodobnejšej verzii. V praxi, pokiaľ majú prijaté a pravdepodobne odosielané slovo priveľkú vzdialenosť, môžeme požadovať opätovné zaslanie slova, aj ak sme chyby dokázali podľa našej teórie opraviť jednoznačne.

3 Ako fungujú samoopravné kódy

3.1 Detekcia chýb

V tejto časti ukážeme, ako závisí od vlastností kódu, koľko chýb vie odhaliť. Hovoríme, že kód odhalí (detekuje) chybu, všeobecnejšie, odhalí nejaký chybový vzor u, pokiaľ pre všetky kódové slová v, platí, že $v + u$ nie je kódovým slovom. Inak povedané, ak na zasielanom kódovom slove v nastanú chyby podľa vzoru u, prijímateľ odhalí, že prijaté slovo nie je kódové, čiže detekuje chybový vzor u. Ak kód chybový vzor odhaľuje, nemusí nutne chyby, ktoré vzor na slove spôsobil, lokalizovať a opravovať (čiže určiť o aký chybový vzor ide), určuje len, že k chybe došlo.

Množstvo chýb, ktoré kód dokáže odhaliť, závisí od jeho minimálnej vzdialenosti. Uvažujme kód, ktorého je minimálna vzdialenosť d. Uvažujme tiež nejaký nenulový chybový vzor u, ktorý má váhu najviac $d-1$ a kódové slovo v. Potom $d(v, v+u) = wt(v+v+u) = wt(u) < d$. Keďže minimálna vzdialenosť kódu je d, vieme, že $v+u$ nie je kódové slovo. Tento kód preto odhaľuje chybový vzor u.

Zároveň dokážeme zvoliť dvojicu slov v a w, ktorých vzdialenosť určuje minimálnu vzdialenosť kódu. Ich chybový vzor u má potom váhu d a zároveň ho kód neodhaľuje, pretože $w = v + u$ je kódové slovo.

Z týchto úvah plynie, že tento kód dokáže odhaliť všetky nenulové chybové vzory, ktorých váha je najviac $d-1$, čiže dokáže odhaliť až $d-1$ chýb. Navyše, existuje aspoň jeden chybový vzor váhy d, ktorý neodhalí.

Napríklad paritný kód s minimálnou vzdialenosťou 2 dokáže odhaliť jednu chybu a 3-opakovací kód s minimálnou vzdialenosťou 3 až dve, keďže aspoň jeden blok obsahujúci správu ostane stále zachovaný.

Informácie v tejto podkapitole boli čerpané zo zdroja [1].

3.2 Korekcia chýb

Počet chýb, ktoré vie kód opraviť, opäť závisí od jeho minimálnej vzdialenosti. Hovoríme, že kód opravuje chybový vzor u, ak pre všetky kódové slová v, platí, že slovo $v + u$ má menšiu vzdialenosť od slova v než od každého iného kódového slova. V princípe, pokiaľ sa na slove v pri prenose poškodia cifry, ktoré determinuje chybový vzor u a prijaté slovo bude stále najbližším slovom k odosielanému, budeme správne predpokladať, že toto slovo bolo odoslané, čiže dokážeme chyby opraviť.

Uvažujme preto kód s minimálnou vzdialenosťou d a dve rôzne kódové slová v a w. Uvažujme tiež chybový vzor u, splňujúci $wt(u) \leq (d-1)/2$. Z vlastností vzdialenosti slov a minimálnej vzdialenosti kódu vieme, že platí:

$$d(w, v+u) + d(v+u, v) \geq d(w, v) \geq d$$

Využijeme vzťah pre váhu chybového vzoru u a fakt, že $d(v+u, v) = wt(u)$.

$$d(w, v+u) + wt(u) \geq 2wt(u) + 1$$

$$d(w, v+u) \geq wt(u) + 1 > d(v, v+u)$$

Odtiaľ plynie, že pre ľubovoľný chybový vzor u splňujúci $wt(u) \leq (d-1)/2$ platí, že $v+u$ je bližšie k ľubovoľne zvolenému slovu v než k akémukoľvek inému kódovému slovu. Čiže kód so vzdialenosťou d opraví každý takýto chybový vzor, a teda opravuje $\lfloor (d-1)/2 \rfloor$ chýb.

Podobne ako v predošlom prípade, vieme skonštruovať chybový vzor u s váhou o 1 vyššou, ktorý už tento kód neopravuje.

Napríklad paritný kód má vzdialenosť 2 a preto nedokáže opraviť žiadnu chybu. Naopak 3-opakovací kód so vzdialenosťou 3 dokáže opraviť jednu chybu.

Informácie v tejto podkapitole boli čerpané zo zdroja [1].

3.3 Lineárne kódy

Lineárnym kódom nazývame taký kód (pripomeňme, že kód chápeme ako množinu slov), ktorý tvorí podpriestor vektorového priestoru $\mathbb{Z}_2^n$. Vzhľadom k telesu, nad ktorým pracujeme, vlastne požadujeme, aby kód obsahoval nulové slovo a bol uzavretý na sčítanie. Lineárne kódy je jednoduché zadať, stačí totiž uviesť nejakú ich bázu. Rovnako kódové slová sú jednoznačne určené prostredníctvom ich súradníc vzhľadom k zvolenej báze. V podstate všetky používané samoopravné kódy sú lineárne. [1]

3.3.1 Generujúca matica

Generujúcou maticou lineárneho kódu nazývame každú maticu, ktorej riadky tvoria bázu tohto kódu. Počet jej riadkov je rovný dimenzii daného lineárneho kódu (chápaného ako vektorový priestor), označme ju r. Pre potreby kódovania je užitočné používať generujúcu maticu v odstupňovanom tvare, ideálne v redukovanom odstupňovanom tvare.

Niektoré kódy majú vlastnosť, že prvé stĺpce ich generujúcej matice v redukovanom odstupňovanom tvare tvoria identickú maticu I_r. V takom prípade hovoríme, že lineárny kód má generujúcu maticu v **štandardnom tvare**. V prípade, že to tak nie je, existuje ekvivalentný lineárny kód s prehodeným poradím cifier v slovách, ktorý túto vlastnosť má. [1]

Kódovanie pri použití lineárneho kódu s generujúcou maticou v štandardnom tvare je potom veľmi prirodzený proces.Na začiatku má odosielateľ k dispozícii správu, ktorú chce poslať. Prípustné správy vlastne predstavujú všetky možné súradnice vzhľadom k zvolenej báze kódu. Takúto správu potom môžeme chápať ako riadkový vektor, ktorý má r zložiek, čo je rovnako ako počet riadkov generujúcej matice. Po prenásobení tohto riadkového vektoru generujúcou maticou dostaneme riadkový vektor, ktorý je kódovým slovom. Zároveň má prvých r symbolov zhodných s pôvodnou správou (vďaka štandardnému tvaru matice). Týchto prvých r symbolov predstavuje informačné cifry a zvyšné predstavujú kontrolné cifry. [1]

Napríklad si predstavme, že máme lineárny kód C s bázou $(10010, 01001, 00100)$. Kód má dimenziu 3, preto môžeme chcieť napríklad poslať správu 011. Použijeme pri tom generujúcu maticu, ktorá má v riadkoch bázické vektory. Zvolená generujúca matica je v štandardnom tvare. Správu zakódujeme vynásobením správy generujúcou maticou a výsledkom je správa 01101. Zachovali sme pri tom prvé tri informačné symboly správy a dodali dva kontrolné.

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

3.3.2 Kontrolná matica

Druhou významnou maticou pre lineárny kód je jeho **kontrolná matica**. Pod týmto názvom rozumieme maticu, ktorej jadrom je práve daný lineárny kód. Zároveň je to v istom

zmysle najmenšia taká matica, presnejšie, ak má kód dimenziu r a kódové slová dĺžku n, tak táto matica má $n - r$ riadkov. Jej význam je najmä v jednoduchom overení, či je prijaté slovo kódovým slovom – stačí ním prenásobiť kontrolnú maticu (v podobe stĺpcového vektoru) a overiť, či je výsledkom nulové slovo. V praxi sa lineárne kódy zadávajú najmä pomocou ich kontrolnej matice. [2]

Napríklad pri kóde C z časti 3.3.1 je $n = 5$ a $r = 3$ a jeho kontrolnou maticou je napríklad matica typu 2x5:

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

Ľahko overíme, že slovo 01101 je v jadre tejto matice, čiže je kódovým slovom a napríklad slovo 01111 nie je, čiže do kódu nepatrí.

3.3.3 Vzdielenosť lineárneho kódu

Pri lineárnom kóde vieme viac povedať aj o jeho minimálnej vzdialenosti. Uvažujme nejaký lineárny kód so vzdialenosťou d. Pre ľubovoľné nenulové kódové slovo v platí, že $wt(v) = d(v, o) \geq d$, kde o je nulové slovo. Zároveň existujú nejaké rôzne kódové slová a, b, ktoré spĺňajú $d(a, b) = d$. Keďže platí $d(a, b) = wt(a + b)$ a $a + b$ je z linearity kódu (nenulovým) kódovým slovom, tak existuje nenulové kódové slovo s váhou d. Z toho plynie, že vzdialenosť lineárneho kódu je rovná minimálnej váhe nenulových kódových slov. [1]

Vzdialenosť lineárneho kódu sa dá tiež zistiť priamo z kontrolnej matice. Platí totiž, že lineárny kód má vzdialenosť d práve vtedy, keď je ľubovoľná $(d - 1)$-prvková postupnosť stĺpcov kontrolnej matice lineárne nezávislá a existuje d-prvková postupnosť jej stĺpcov, ktorá je lineárne závislá. Dôkaz plynie z toho, že kód neobsahuje žiadne nenulové slovo s váhou najviac $d - 1$ a naopak obsahuje nejaké s váhou d. [2]

3.3.4 Korekcia chýb v lineárnom kóde

Opravovanie chýb pri použití lineárneho kódu sa dá značne zjednodušiť. Uvažujme kód C. Pri prijatí slova w, ktoré nie je kódovým slovom, vieme vytvoriť skupinu slov, ktoré môžu byť potenciálnymi chybovými vzormi. Využijeme, že hľadaný chybový vzor je súčtom odosielaného kódového slova a prijatého slova. Preto nájdeme všetky slová $v+w$, kde v je kódové slovo, čiže množinu slov $\{v+w : v \in C\}$. Táto množina sa nazýva **coset**[1] určený slovom w. Keďže predpokladáme výskyt vzoru s najnižšou váhou, z tejto skupiny vyberieme to slovo, ktoré ju má, a pomocou neho určíme odosielané slovo. [1]

V prípadoch, kedy je táto procedúra priveľmi zdĺhavá, môžeme využiť ďalšiu vlastnosť slov v rovnakom cosete. Tou je, že výsledkom násobenia kontrolnej matice vektormi z rovnakého cosetu je vždy rovnaký vektor. Ten sa nazýva **syndróm**. Miesto vytvárania celého cosetu tak môžeme prehľadať chybové vzory s nízkou váhou a vybrať z nich ten, ktorý má rovnaký syndróm ako prijaté slovo. [1]

3.4 Hammingove kódy

Richard W. Hamming je otcom prvých samoopravných kódov. Navrhol niekoľko kódov, ktoré kombinujú viacero paritných testov. Základný príklad Hammingovho kódu, ktorý sa

[1] výraz je prevzatý z angličtiny, nenašli sme vhodný slovenský ekvivalent v literatúre

dá pochopiteľne rozšíriť na väčšie rozmery, ukážeme na konštrukcii so štyrmi informačnými symbolmi.

3.4.1 Hammingov kód s ôsmimi symbolmi

Predpokladajme, že chceme odoslať text so štyrmi informačnými symbolmi a, b, c a d. Všetky symboly si pre ilustráciu budeme vkladať do matice typu 3x3, pričom tie informačné budú uložené na pozíciách (1,1), (1,2), (2,1) a (2,2). Zvyšné bunky využijeme na paritné testy. Do prázdnej bunky v prvom a druhom riadku vložíme čísla r_1 a r_2, tak aby bol v každom riadku párny počet cifier 1. Rovnako doplníme čísla s_1 a s_2 na prázdne miesta v prvom a druhom stĺpci. Poslednú bunku môžeme nechať prázdnu. Výsledkom je matica:

$$\begin{pmatrix} a & b & r_1 \\ c & d & r_2 \\ s_1 & s_2 & \end{pmatrix}$$

Použitie matice je len ilustračné, v praxi môžeme správu napríklad previesť po riadkoch na 8-znakové kódové slovo $abr_1cdr_2s_1s_2$. Zo vzťahov, ktoré maticu definujú, je zrejmé, že tento kód opravuje jednu chybu, pretože ak ku nej dôjde, chybný bude paritný test v príslušnom riadku a stĺpci, teda je presne určené, ktorý symbol je chybný.

Paritné testy, ktoré sme použili pri vypĺňaní matice sa dajú vďaka vlastnostiam počítania v telese $\mathbb{Z}_2$ zapísať nasledovnými vzťahmi:

$$a + b = r_1 \qquad c + d = r_2 \qquad a + c = s_1 \qquad b + d = s_2$$

Tieto vzťahy nám vlastne udávajú sústavu rovníc, resp. jej maticu, ktorej jadrom sú práve kódové slová. Inak povedané, vieme z nich zostaviť kontrolnú maticu, ktorou je

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

Tento Hammingov kód je lineárny (keďže je jadrom matice a teda podpriestorom $\mathbb{Z}_2^n$), preto z uvedeného vzťahu pre lineárne kódy vieme overiť, že jeho minimálna vzdialenosť je 3, teda dokáže odhaliť dve chyby a opraviť jednu. Navyše, z 8 potrebných cifier sú 4 informačné, takže prenosová rýchlosť je 1/2, čo je podstatne lepšie než pri 3-opakovacom kóde s rovnakou vzdialenosťou.

3.4.2 Hammingov kód so siedmimi symbolmi

Nasledovný kód definujeme priamo pomocou kontrolnej matice:

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

Kódové slová sú slová z jadra kontrolnej matice. Matica je v riadkovo odstupňovanom tvare, takže z nej vidíme, že dimenzia kódu je 4. Pre kódové slovo $x_1x_2x_3x_4x_5x_6x_7$ platí, že premenné x_4, x_5, x_6 a x_7 sú voľné, teda môžu predstavovať informačné symboly, zatiaľ čo zvyšné budú kontrolné. Z charakteru rovníc v kontrolnej matici vidíme, že ide o 3 paritné testy. Zo 7

symbolov v slove sú 4 informačné, čiže prenosová rýchlosť je ešte o niečo lepšia, konkrétne 4/7.

Zo vzťahu pre vzdialenosť lineárneho kódu vieme, že minimálna vzdialenosť uvedeného kódu je 3. Kód teda opravuje jednu chybu.

Vysvetlíme si mechanizmus, ako túto chybu opraviť. Využijeme pri tom, že kontrolná matica obsahuje v stĺpcoch všetky nenulové vektory v $\mathbb{Z}_2^3$. Predpokladajme, že sme prijali slovo $y = y_1y_2y_3y_4y_5y_6y_7$. Buď dostaneme vynásobením kontrolnej matice týmto slovom nulové slovo, a teda sme prijali kódové slovo, alebo dostaneme nenulový vektor s v $\mathbb{Z}_2^3$, ktorý je syndrómom slova y. Tento vektor je v niektorom stĺpci kontrolnej matice, povedzme v j-tom. Preto ho rovnako vieme získať vynásobením kontrolnej matice vektorom e_j (j-tým kanonickým vektorom v $\mathbb{Z}_2^7$). Výsledkom násobenia kontrolnej matice vektorom $y + e_j$ bude preto (z distributívnosti násobenia matíc) vektor $s + s$, ktorý je nulovým slovom. Slovo $y + e_j$ je preto kódovým slovom, ktoré sa navyše líši od y len v jedinom znaku. Z toho môžeme usúdiť, že je to hľadané kódové slovo, pričom poškodený bol j-tý symbol.

Napríklad ak by sme prijali slovo 1010101, tak jeho syndróm bude $(1, 1, 1)^T$. To je siedmy stĺpec kontrolnej matice, preto bol poškodený siedmy symbol a odosielané kódové slovo bolo 1010100. [2]

Uvedený kód patrí medzi perfektné kódy, pretože na počet chýb, ktoré opravuje, a počet informačných symbolov v ňom už nemôže obsahovať menej kontrolných symbolov. Presnejšie, kód s minimálnou vzdialenosťou d je **perfektný**, ak má každé slovo vzdialenosť od nejakého kódového slova najviac $d - 1$, čo tento kód spĺňa, ako sme videli na princípe jeho dekódovania. Konštrukcia Hammingových kódov, ktoré opravujú jednu chybu a sú perfektné, sa dá zovšeobecniť na všetky kódy so slovami dĺžky $n = 2^r - 1$, kde rxn sú rozmery kontrolnej matice, ktorá obsahuje v stĺpcoch všetky nenulové vektory v $\mathbb{Z}_2^r$.

Informácie v tejto podkapitole boli čerpané zo zdroja [2].

4 Význam samoopravných kódov

Dôsledky vzniku samoopravných kódov majú obrovský význam a dopad na reálny život či už v každodenne používaných technológiách alebo tých, ktoré slúžia na výskumnú činnosť. Ich typické použitie v bežnom živote je predovšetkým pri odosielaní dát po internetovej sieti alebo uchovávaní dát napríklad na CD nosičoch. [5] Vo vedeckej sfére je známa napríklad komunikácia s vesmírnymi sondami počas misií Voyager 1 a Voyager 2, prostredníctvom ktorej boli získané napríklad farebné fotografie planét Jupiter a Saturn. [6]

Používanie samoopravných kódov v praxi dokáže dramatickým spôsobom znížiť množstvo neodhalených resp. neopravených chýb pri odosielaní veľkého množstva dát dokonca aj pri použití pomerne spoľahlivého komunikačného kanálu. V prípade problematickejšej spoľahlivosti komunikačných kanálov spôsobenej vzdialenosťou alebo opotrebovaním je ich použitie prakticky nevyhnutné. Samoopravné kódy sú významnou súčasťou podstaty elektronickej komunikácie.

Zoznam použitej literatúry

[1] HOFFMAN, D. G., LEONARD, D.A., LINDNER, C.C., PHELPS, K.T., RODGER, C.A., WALL, J. R. 1991. *Coding theory: The Essentials*. 1. vyd. New York: Marcel Dekker, inc., 1991. ISBN 0-8247-8611-4

[2] http://www.karlin.mff.cuni.cz/~barto/LinAlg/skripta_la5.pdf

[3] http://www.karlin.mff.cuni.cz/~tuma/2002/NLinalg8.pdf

[4] http://mmm.zcu.cz/seminar2012/kaiser.pdf

[5] https://pdfs.semanticscholar.org/0da3/350929c6f790e5004cd4b8ffdb6b19354826.pdf

[6] https://en.wikipedia.org/wiki/Error_detection_and_correction

[7] http://news.mit.edu/2010/explained-shannon-0115

[8] https://en.wikipedia.org/wiki/Noisy-channel_coding_theorem