Kódování

Jan Paseka

12. května 1999

Předmluva

Teorie kódování je interdisciplinární teorie, která v sobě spojuje metody a postupy informatiky, matematiky a spojovací techniky. Teorie kódování přitom stále více a více nachází bezprostřední aplikaci v praxi vlivem nových technologických změn.

To, co je na teorii kódování tak strhující, je z jedné strany těsná souvislost teorie s praxí a ze strany druhé pak mnohostranost používaných matematických metod.

Úlohou teorie kódování je tvorba postupů a metod, které nám zajistí bezpečný přenos zpráv komunikačním systémem. Z důvodů technické realizovatelnosti se zprávy převedou nejprve do řady znaků nad nějakou konečnou abecedou (nejlépe nad konečným tělesem). Tato řada znaků se pak rozloží do bloků pokud možno stejné délky k. Kódovací zařízení nám pak utvoří z každého bloku délky k blok délky n,kde $n \geq k$. Redundance získaná v případě kdy $n > k$ slouží později k rozpoznání a případné opravě pokud možno co nejvíce přenosových chyb. Přenos bloků délky n pomocí spojovacího systému, které reprezentují kódované zprávy a které se jako celek označují blokové kódy délky n, si lze představit buď prostorově (přes satelit, telefonem, televizí, rádiem atd.) nebo také v čase (CD, gramodeska, magnetofonová páska atd.). Podíl k/n se nazývá míra informace blokového kódu a reprezentuje množství energie potřebné k přenosu kódovaných zpráv.

V rušeném spojovém kanálu se mohou při přenosu kódovaných zpráv vyskytnout chyby dvojího typu. Nejprve je myslitelné, že některé z vysílaných zpráv nedojdou vůbec k příjemci nebo že je příjemce obdrží neúplné. Druhou možností je, že se mohou vyskytnout rovněž přenosové chyby, tj. vyslaný znak 0 se např. přijme jako 1; v teorii kódování se zabýváme zejména s druhým případem.

Pro opravu eventuálně se vyskytujících se přenosových chyb jsou rozhodující dvě veličiny:

- míra opravitelnosti chyb, která nám udává v každé kódované zprávě podíl opravitelných chyb, a

- komplexita dekóderu, který má za úlohu pro přijatou kódovanou zprávu zjistit vyslanou zprávu.

Hlavním cílem teorie kódování je tvorba kódu s pokud možno co největší mírou informace a s co možná největší mírou opravitelnosti chyb při současně co možná

nejmenší komplexitě dekódéru.

Shannonova věta o kapacitě kanálu nám zaručuje existenci blokových kódů s mírou informace libovolně blízce pod kapacitou kanálu, tzn. s mírou informace, která je tak vysoká jak nám to používaný kanál vůbec dovolí a s libovolně velkou mírou opravitelnosti chyb. Nekonstruktivní charakter této skutečnosti byl zrodem teorie kódování.

V mnoha případech je však časová náročnost pro dekódování kódu tak velká, že neúplné využití kapacity kanálu má mnohem menší důležitost než příliš komplikovaný dekódovací postup. Z tohoto důvodu se v teorii kódování zkoumají značně kódy s relativně jednoduchým realizovatelným dekódovacím algoritmem.

Pro určení vlastností opravujících se chyb daného kódu se ukázala důležitá dodatečná znalost jeho struktury. Proto se v teorii kódování zkoumají blokové kódy opatřené dodatečnou algebraickou strukturou, u kterých lze doufat, že budou mít v praxi použitelné teoretické vlastnosti.

Lineární kódy reprezentují jistou třídu blokových kódů a jsou opatřeny dodatečnou algebraickou strukturou – strukturou vektorového prostoru. Pak lineární kód nad konečným tělesem K je reprezentován jako k-rozměrný podprostor n-rozměrného vektorového prostoru nad K. Strukturu lineárních kódů lze pak analyzovat prostředky a metodami lineární algebry. K nejznámějším příkladům praktického použití lineárních kódů patří

- binární Reed-Mullerovy kódy – vesmírná sonda „Mariner“ použila binární Reed-Mullerův kód prvního řádu délky 32 pro přenos datového materiálu fotodokumentace planety Mars, a rovněž

- Reed-Solomonovy kódy – např. se používají pro ukládání opticky kódovaných zvukových signálů na CD dva lineární kódy, které byly odvozeny zkrácením Reed-Solomonova kódu délky 255 nad tělesem GF(2^8).

Tento učební text je založen na monografii „Codes and Cryptography“ D. Welshe. Zároveň využívá texty českých autorů jako je např. „Kódování“ Jiřího Adámka. Text je zpřístupněn všem studentům (a uživatelům INTERNETu) pomocí anonymního FTP přístupu na počítač ftp.muni.cz. Adresář /pub/math.muni.cz je jeden z disků počítače king.math.muni.cz. Tento adresář obsahuje adresáře lectures, papers, software. Na počítači v adresáři lectures je zřízen adresář pro ukládání materiálu pro přednášky/semináře.

Obsah

Kapitola 1

Entropie = neurčitost = nejistota = informace

1 Nejistota

Uvažme následující tvrzení.

A Výsledek běhu mezi dvěma si rovnými závodníky je méně nejistý než výsledek běhu mezi šesti si rovnými závodníky.

B Výsledek rulety je více nejistý než vrh kostkou.

C Výsledek vrhu ideální kostkou je více nejistý než výsledek vrhu falešnou kostkou.

Zřejmě s výše uvedenými tvrzeními lze okamžitě souhlasit. Obtížné však bude definovat, co je to vlastně nejistota. Podívejme se na dvě různé náhodné veličiny X a Y. Nechť

$$P(X=0)=p, \quad P(X=1)=1-p,$$

a

$$P(Y=100)=p, \quad P(Y=200)=1-p,$$

přičmž $0<p<1$.

Zřejmě by nám definice nejistoty měla zajistit, že X a Y jsou stejně nejisté. Tedy nejistota X a tedy i Y by měla být funkcí *pouze* pravděpodobnosti p. Tato vlastnost nejistoty musí být rozšiřitelná i na náhodné proměnné, které nabývají více než dvou hodnot. Tedy:

> Nejistota náhodné proměnné Z, která nabývá hodnoty a_i s pravděpodobností p_i, $(1 \le i \le n)$, je funkcí *pouze* pravděpodobností $p_1, \dots, p_n$.

Proto značíme takovouto funkci jako $H(p_1, \dots, p_n)$, přičemž předpokládáme splnění následujících přirozených podmínek:

(A1) $H(p_1, \dots, p_n)$ je maximální, když $p_1 = p_2 = \ldots = p_n = 1/n$.

(A2) Pro každou permutaci π na $\{1, \dots, n\}$ platí

$$H(p_1, \dots, p_n) = H(p_{\pi(1)}, \dots, p_{\pi(n)}).$$

Tedy H je symetrická funkce svých argumentů tj. její výsledek nezávisí na pořadí.

(A3) $H(p_1, \dots, p_n) \ge 0$ a rovnost nastává právě tehdy, když $p_i = 1$ pro nějaké i. Nejistota má tedy vždy nezápornou hodnotu a je nulová právě tehdy, když je jakákoliv náhoda vyloučena.

(A4)

$$H(p_1, \dots, p_n, 0) = H(p_1, \dots, p_n).$$

Nejistota vrhu šestibokou ideální kostkou je tatáž jako nejistota vrhu sedmibokou kostkou, u které je nemožné aby padla 7 ale ostatní případy jsou si rovnocenné.

(A5)

$$H(1/n, \dots, 1/n) \le H(1/n+1, \dots, 1/n+1).$$

Výsledek běhu mezi dvěma závodníky je méně nejistý než výsledek běhu mezi více závodníky.

(A6) $H(p_1, \dots, p_n)$ je spojitá funkce svých parametrů. Malé změny na vstupu dají malé změny na výstupu.

(A7) Jsou-li $m, n \in \mathbf{N}$, pak

$$H(1/m \cdot n, \dots, 1/m \cdot n) = H(1/m, \dots, 1/m) + H(1/n, \dots, 1/n).$$

Tato podmínka říká, že nejistota vrhu $m \cdot n-$stranné kostky je obsažena ve vrhu m-stranné kostky následovaná vrhem n-stranné kostky, a je rovna součtu individuálních nejistot.

(A8) Nechť $p = p_1 + \ldots + p_m$ a $q = q_1 + \ldots + q_n$, p_i, q_j jsou neznámé. Jsou-li p a q kladná čísla, $p+q=1$, pak platí

$$H(p_1, ..., p_m, q_1, \dots, q_n) = H(p,q) + p \cdot H(p_1/p, \dots, p_m/p) + q \cdot H(q_1/q, \dots, q_n/q).$$

Představme si, že máme $n+m$ uchazečů na místo v konkurzu - z toho je m mužů a n žen, s pravděpodobnostmi p_i, q_j vítězství v konkurzu. Pak nejistota výsledku konkurzu je nejistota, že vyhraje muž nebo žena plus vážený součet, nejistot výhry mezi muži a ženami.

Věta 1.1 *Buď $H(p_1,\dots,p_n)$ funkce definovaná pro každé přirozené číslo n a pro všechny hodnoty $p_1,\dots,p_n$ tak, že $p_i \geq 0$ a*

$$\sum_{i=1}^{n} p_i = 1.$$

Pokud H splňuje axiómy (A1)-(A8), pak platí

$$H(p_1,p_2,\dots,p_n) = -\lambda \sum_k p_k \cdot \log p_k, \tag{1.1}$$

kde λ je libovolná kladná konstanta a sumuje se přes všechna k taková,že $p_k > 0$.

Důkaz. Nechť H splňuje axiómy (A1)-(A8). Definujme

(1) $g(n) = H(1/n,\dots,1/n)$ pro $n \in \mathbf{N}$. Z (A7) pak

$$g(n^k) = g(n) + g(n^{k-1})$$

a tedy

(2) $g(n^k) = k \cdot g(n)$. Buď dále $r,s \in \mathbf{N} - \{1\}, n \in \mathbf{N}$ a $m = m(n,r,s) \in \mathbf{N}$ tak, že

(3) $r^m \leq s^n \leq r^{m+1}$; pak dle (2) a monotonie g (dle (A5)) máme

$$g(r^m) \leq g(s^n) \leq g(r^{m+1}),$$

tedy

$$m \cdot g(r) \leq n \cdot g(s) \leq (m+1) \cdot g(r).$$

Z (3) pak máme

$$m \cdot \ln(r) \leq n \cdot \ln(s) \leq (m+1) \cdot \ln(r)$$

a tedy

$$\left| \frac{g(s)}{g(r)} - \frac{\ln(s)}{\ln(r)} \right| \leq \frac{2}{n}.$$

Protože n bylo libovolnou přirozené číslo, je

$$\frac{g(s)}{\ln(s)} = \frac{g(r)}{\ln(r)} = A,$$

kde A je nějaká konstanta. Tedy

(5)

$$g(s) = A \cdot \ln(s).$$

Buď $0 < p < 1$ racionální, $p = t/n$, $t,n \in \mathbf{N}$. Položme $q = (n-t)/n$. Z (A8) pak

$$g(n) = H(\frac{1}{n},\dots,\frac{1}{n}) = H(\frac{t}{n},\frac{n-t}{n}) + \frac{t}{n} \cdot g(t) + \frac{n-t}{n} \cdot g(n-t).$$

Z (5) pak jednoduchou úpravou

$$H(\frac{t}{n},\frac{n-t}{n}) = A \cdot (\frac{t}{n}) \cdot \ln\frac{t}{n} + A \cdot (\frac{n-t}{n}) \cdot \ln\frac{n-t}{n}.$$

Zejména pak

(6)

$$H(p,1-p) = A \cdot p \cdot \ln p + A \cdot (1-p) \cdot \ln(1-p),$$

a to pro každé racionální číslo p mezi 0 a 1. Ze spojitosti H platí (6) pro všechna $0 < p < 1$. Dokažme, že pro každé $N \in \mathbf{N}$ platí

(7)

$$H(p_1,\dots,p_N) = A \cdot \sum_{i=1}^{N} p_i \cdot \ln p_i,$$

přičemž $p_i > 0$ a $p_1 + \ldots + p_N = 1$, a to indukcí podle N. Z (6) víme, že (7) platí pro $N = 2$. Předpokládejme, že (7) platí pro N a uvažujme $H(p_1,\dots,p_{N+1})$. Položme $p = p_1 + \ldots + p_N, q = p_{N+1}$, a použijme (A8). Máme pak

$$\begin{aligned} H(p_1,\dots,p_{N+1}) &= H(p,q) + p \cdot H(\frac{p_1}{p},\dots,\frac{p_N}{p}) + q \cdot H(1) \\ &= A \cdot p \cdot \ln p + A \cdot q \cdot \ln q + p \cdot A \cdot \sum_{i=1}^{N} \frac{p_i}{p} \ln\frac{p_i}{p}, \end{aligned}$$

z indukčního předpokladu. Upravíme-li poslední rovnost na tvar

$$H(p_1,\dots,p_{N+1}) = A \cdot p \cdot \ln p + A \cdot p_{N+1} \cdot \ln p_{N+1} + A \cdot \sum_{i=1}^{N} p_i \cdot (\ln p_i - \ln p),$$

a vzpomeneme-li si, že $\sum_{i=1}^{N} p_i = p$, obdržíme hledanou rovnost

$$H(p_1,\dots,p_{N+1}) = A \cdot \sum_{i=1}^{N+1} p_i \cdot \ln p_i.$$

Na základě výše uvedené věty pak definujeme

Definice. Buď X náhodná proměnná s konečným oborem hodnot s odpovídajícími pravděpodobnostmi $p_1, p_2, \ldots, p_n$. Pak definujeme *nejistotu* neboli *entropii* náhodné veličiny X jako

$$H(X) = -\sum_k p_k \cdot \log_2 p_k, \tag{1.2}$$

kde suma se bere pouze přes ta k, pro která je $p_k > 0$.

Poznámka 1.2 *Nadále budeme vždy (bez újmy na obecnosti) předpokládat, že pro všechny členy pravé strany* (1.2) *jsou pravděpodobnosti p_k nenulové.*

Poznámka 1.3 *Podmínky (A1)-(A8) odpovídají axiomům pro entropii navrženým Shannonem.*

Cvičení 1.4

1. *Který dostih má větší entropii: handikap, ve kterém je sedm žokejů, tři z nich vyhrají s pravděpodobností $\frac{1}{6}$ a čtyři z nich s pravděpodobností $\frac{1}{8}$ nebo dostih, v němž musí být vítěz prodán za předem stanovenou cenu a ve kterém je 8 žokejů s dvěma koni s pravděpodobností výhry $\frac{1}{4}$ a šest koní s pravděpodobností výhry $\frac{1}{12}$?*
2. *Ověřte, že výše definovaná funkce entropie splňuje podmínky (A1)-(A8).*

2 Entropie a její vlastnosti

Řekli jsme, že pro náhodnou proměnnou X s konečným oborem hodnot a s pravděpodobnostmi $p_1, \ldots, p_n$ tak, že

$$\sum p_i = 1 \text{ a } p_i > 0 (1 \leq i \leq n),$$

definujeme entropii X jako

$$H(X) = -\sum_{k=1}^{n} p_k \cdot \log_2 p_k.$$

Analogicky pak pro náhodný vektor $\mathbf{X}$, který nabývá pouze konečně mnoha hodnot $\mathbf{u}_1, \ldots, \mathbf{u}_m$, defimujeme jeho *entropii* náhodného velktoru jako

$$H(\mathbf{X}) = -\sum_{k=1}^{m} p(\mathbf{u}_k) \cdot \log_2 p(\mathbf{u}_k). \tag{1.3}$$

Je-li například $\mathbf{X}$ 2-dimenzionální náhodný vektor, $\mathbf{X} = (U, V)$ s

$$p_{ij} = P(U = a_i, V = b_j),$$

budeme často psát

$$H(\mathbf{X}) = H(U, V) = -\sum p_{ij} \cdot \log_2 p_{ij}.$$

Zcela obecně, jsou-li $X_1, \ldots, X_m$ náhodné proměnné tak, že každá z nich nabývá pouze konečně mnoha hodnot, lze pak považovat $\mathbf{X} = (X_1, \ldots, X_m)$ za náhodný vektor a definovat souhrnou entropii $X_1, \ldots, X_m$ jako

$$H(X_1, \ldots, X_m) = H(\mathbf{X}) = -\sum_{(x_1, \ldots, x_m)} p(x_1, \ldots, x_m) \cdot \log_2 p(x_1, \ldots, x_m), \tag{1.4}$$

kde $p(x_1, \ldots, x_m) = P(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m)$. Snadno se ověří, že:

$$H(\mathbf{X}) = 0 \text{ právě tehdy, když } \mathbf{X} \text{ je konstantní.} \tag{1.5}$$

Horní hranice pro H je určena následující větou:

Věta 2.1 *Pro každé přirozené číslo n máme*

$$H(p_1, \ldots, p_n) \leq \log_2 n,$$

přičemž rovnost nastává právě tehdy, když $p_1 = p_2 = \ldots = p_n = n^{-1}$.

Důkaz. Zřejmě

$$\log_2 x = \log_2 \mathrm{e} \log_\mathrm{e} x.$$

Protože logaritmus je konvexní funkce tj. leží celá pod tečnou, máme

$$\log_\mathrm{e} x \leq x - 1,$$

přičemž rovnost nastává právě tehdy, když $x = 1$. Tedy, je-li $(q_1, \ldots, q_n)$ libovolné pravděpodobnostní rozdělení, pak máme

$$\log_\mathrm{e}(q_k/p_k) \leq (q_k/p_k) - 1,$$

s rovností právě tehdy, když $q_k = p_k$. Tudíž,

$$\sum p_i \cdot \log_\mathrm{e}(q_i/p_i) \leq \sum q_k - \sum p_k = 0,$$

a z toho pak

$$\sum p_i \cdot \log_2 q_i \leq \sum p_i \cdot \log_2 p_i.$$

Položíme-li $q_i = 1/n$, obdržíme dosazením

$$H(p_1, \ldots, p_n) = -\sum p_i \cdot \log_2 p_i \leq \log_2 n,$$

což se mělo dokázat. Zbytek tvrzení o rovnosti plyne bezprostředně z důkazu.

Poznamenejme, že jsme dokázali velmi užitečnou nerovnost a to:

Lemma 2.2 *Je-li $(p_i : 1 \leq i \leq n)$ dané pravděpodobnostní rozdělení, pak minimum funkce*

$$G(q_1, \ldots, q_n) = -\sum p_i \cdot \log_2 q_i$$

přes všechna pravděpodobnostní rozdělení $(q_1, \ldots, q_n)$ nastává, pokud $q_k = p_k \quad (1 \leq k \leq n)$.

Věta 2.3 *Jsou-li X a Y náhodné proměnné s konečným oborem hodnot, platí pak*

$$H(X,Y) \leq H(X) + H(Y),$$

přičemž rovnost nastává tehdy a jen tehdy, když X a Y jsou nezávislé.

Důkaz. Předpokládejme, že

$$r_i = P(X = a_i)\ (1 \leq i \leq m), \quad s_j = P(Y = b_j)\ (1 \leq j \leq n),$$

$$t_{ij} = P(X = a_i, Y = b_j)\ (1 \leq i \leq m, 1 \leq j \leq n).$$

Pak

$$\begin{aligned} H(X) + H(Y) &= -\left(\sum_i r_i \cdot \log_2 r_i + \sum_j s_j \cdot \log_2 s_j\right) \\ &= -\left(\sum_{i,j} t_{ij} \cdot \log_2 r_i + \sum_{j,i} t_{ij} \cdot \log_2 s_j\right), \end{aligned}$$

protože

$$\sum_j t_{ij} = r_i, \sum_i t_{ij} = s_j.$$

Odtud pak

$$\begin{aligned} H(X) + H(Y) &= -\sum_{i,j} t_{ij} \cdot \log_2(r_i \cdot s_j) \\ &\geq -\sum_{i,j} t_{ij} \cdot \log_2(t_{ij}) = H(X,Y) \end{aligned}$$

dle předchozího lemmatu. Rovnost nastane právě tehdy, když

$$r_i \cdot s_j = t_{ij}.$$

Ale to je právě podmínka nezávislosti X a Y.

Jednoduchým rozšířením této metody lze dokázat:

$$H(X_1, \ldots, X_n) \leq H(X_1) + \ldots + H(X_n), \tag{1.6}$$

přičemž rovnost nastává právě tehdy, když $X_1, \ldots, X_n$ jsou navzájem nezávislé;

$$H(\mathbf{U}, \mathbf{V}) \leq H(\mathbf{U}) + H(\mathbf{V}) \tag{1.7}$$

pro každou dvojici náhodných vektorů $\mathbf{U}, \mathbf{V}$, přičemž rovnost nastává právě tehdy, když $\mathbf{U}$ a $\mathbf{V}$ jsou nezávislé náhodné vektory.

Důkazy (jež lze dokázat přesně stejným způsobem jako větu 1.2 z předchozího lemmatu) jsou ponechány čtenáři.

Cvičení 2.4

1. *Dvě ideální kostky jsou vrženy; X označuje hodnotu získanou první kostkou, Y hodnotu získanou druhou kostkou. Dokažte, že $H(X,Y) = H(X) + H(Y)$. Dokažte, že je-li $Z = X + Y$, pak*

$$H(Z) < H(X,Y).$$

2. *Dokažte, že pro každou náhodnou proměnnou X,*

$$H(X, X^2) = H(X).$$

3. *Dokažte, že pro každou posloupnost náhodných proměnných $(X_i : 1 \leq i < \infty)$,*

$$H(X_1, \ldots, X_n) \leq H(X_1, \ldots, X_{n+1}).$$

3 Podmíněná entropie

Předpokládejme, že X je náhodná proměnná na pravděpodobnostním prostoru Ω a A je událost z Ω. Nabývá-li X konečné množiny hodnot $\{a_i : 1 \leq i \leq m\}$, je přirozené definovat *podmíněnou entropii* náhodné proměnné X určenou událostí A jako

$$H(X|A) = -\sum_{k=1}^{m} P(X = a_k|A) \log P(X = a_k|A).$$

Úplně stejně, je-li Y jiná náhodná proměnná nabývající hodnot b_k $(1 \leq k \leq m)$, definujeme *podmíněnou entropii* náhodné proměnné X určenou náhodnou proměnnou Y jako

$$H(X|Y) = \sum_j H(X|Y = b_j) P(Y = b_j).$$

Považujeme $H(X|Y)$ za entropii náhodné proměnné X určenou jistou hodnotou Y zprůměrovanou přes všechny hodnoty, jichž může Y nabývat.

Zcela triviální důsledky definic jsou:

$$H(X|X) = 0, \tag{1.8}$$

$$H(X|Y) = H(X) \text{ jsou-li } X \text{ a } Y \text{ nezávislé.} \tag{1.9}$$

Příklad 3.1 *Buď X náhodná proměnná získaná vrháním ideální kostky. Buď dále Y jiná náhodná proměnná určená týmtéž experimentem, přičemž Y se rovná 1, je-li vržená hodnota lichá a 0 v ostatních případech. Protože kostka je ideální,*

$$H(X) = \log 6, H(Y) = \log 2,$$

a

$$H(X|Y) = \log 3.$$

Jsou-li $\mathbf{U}$ a $\mathbf{V}$ náhodné vektory, přirozeně rozšíříme definici podmíněné entropie následovně

$$H(\mathbf{U}|\mathbf{V}) = \sum_j H(\mathbf{U}|\mathbf{V} = \mathbf{v}_j)P(\mathbf{V} = \mathbf{v}_j), \tag{1.10}$$

přičemž se sčítá, jako obvykle, přes (konečný) obor hodnot $\mathbf{v}_j$ tak, že odpovídající pravděpodobnost je kladná.

Jako první příklad, jakým způsobem entropie $H(\mathbf{U}|\mathbf{V})$ měří nejistotu o $\mathbf{U}$ obsaženou ve $\mathbf{V}$, dokážeme:

$$H(\mathbf{U}|\mathbf{V}) = 0 \text{ právě tehdy, když } \mathbf{U} = g(\mathbf{V}) \text{ pro nějakou funkci } g. \tag{1.11}$$

Důkaz. Pravá strana z definice podmíněné entropie je součet konečného počtu nezáporných veličin. Tudíž, aby byla nulová, potřebujeme $H(\mathbf{U}|\mathbf{V} = \mathbf{v}_j) = 0$ pro všechna j. Ale opět každá z těchto nezáporných veličin je nulová právě tehdy, když $\mathbf{U}$ je jednoznačně určená $\mathbf{V}$.

Poněkud více nám dává následující výsledek, který matematicky vyjadřuje ideu, že naše definice podmíněné entropie X při daném Y korektně měří zbývající nejistotu.

Věta 3.2 *Pro každou dvojici náhodných proměnných X a Y, které nabývají pouze konečné množiny hodnot, platí*

$$H(X, Y) = H(Y) + H(X|Y).$$

Důkaz. Bez ztráty na obecnosti lze předpokládat, že X a Y nabývají pouze celočíselných hodnot a, kde to bude nutné, že $p_{ij} = P(X = i, Y = j)$. Nyní

$$\begin{aligned}
H(X,Y) &= -\sum_i \sum_j P(X=i, Y=j) \log P(X=i, Y=j) \\
&= -\sum_i \sum_j P(X=i, Y=j) \log P(X=i|Y=j)P(Y=j) \\
&= -\sum_i \sum_j p_{ij} \log P(X=i|Y=j) - \sum_i \sum_j p_{ij} \log P(Y=j) \\
&= -\sum_i \sum_j P(X=i|Y=j)P(Y=j) \log P(X=i|Y=j) + H(Y) \\
&= -\sum_j P(Y=j) \sum_i P(X=i|Y=j) \log P(X=i|Y=j) + H(Y) \\
&= \sum_j P(Y=j) H(X|Y=j) + H(Y) \\
&= H(X|Y) + H(Y), \text{ což bylo třeba dokázat}
\end{aligned}$$

Věta 3.3 *Jsou-li $\mathbf{U}$ a $\mathbf{V}$ náhodné vektory, které nabývají pouze konečné množiny hodnot, platí*

$$H(\mathbf{U}, \mathbf{V}) = H(\mathbf{V}) + H(\mathbf{U}|\mathbf{V}).$$

Důkaz. Procházíme důkazem předchozí věty, ale namísto X a Y nabývajících pouze celočíselných hodnot i a j máme $\mathbf{U}$ a $\mathbf{V}$ nabývajících hodnot $\mathbf{u}_i$ a $\mathbf{v}_j$, kde $\mathbf{u}_i$ a $\mathbf{v}_j$ jsou zadané vektory.

Následující výsledek je bezprostřední důsledek.

Důsledek 3.4 *Pro každou dvojici $\mathbf{X}$ a $\mathbf{Y}$ náhodných vektorů je $H(\mathbf{X}|\mathbf{Y}) \leq H(\mathbf{X})$ a rovnost nastává právě tehdy, když $\mathbf{X}$ a $\mathbf{Y}$ jsou nezávislé.*

Důkaz.

$$H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{X}, \mathbf{Y}) - H(\mathbf{Y}).$$

Ale $H(\mathbf{X}, \mathbf{Y}) \leq H(\mathbf{X}) + H(\mathbf{Y})$, s rovností právě tehdy, když $\mathbf{X}$ a $\mathbf{Y}$ jsou nezávislé.

Cvičení 3.5

1. *Ukažte, že pro každou náhodnou proměnnou X platí*

$$H(X^2|X) = 0,$$

ale uveďte příklad, že $H(X|X^2)$ není vždy nulová.

2. *Náhodná proměnná X nabývá celočíselných hodnot $1, \ldots, 2N$ se stejnou pravděpodobností. Náhodná proměnná Y je definovaná $Y = 0$, je-li X sudá, ale $Y = 1$, je-li X lichá. Ukažte, že*

$$H(X|Y) = H(X) - 1,$$

ale že $H(Y|X) = 0$.

4 Informace

Zdá se, že R.V.L. Hartley byl v r. 1928 první, kdo se pokusil přiřadit kvantitativní míru k pojmu informace. Racionální příčinu za tímto pokusem můžeme částečně popsat následovně.

Předpokládejme, že E_1 a E_2 jsou dvě události v pravděpodobnostním prostoru Ω spojené jistým experimentem a předpokládejme, že funkce I je naše míra informace. Mají-li E_1 a E_2 pravděpodobnosti p_1 a p_2, pak můžeme argumentovat tím, že každá přirozená míra obsahu informace by měla splňovat

$$I(p_1 p_2) = I(p_1) + I(p_2)$$

na základě toho, že, pro dvě nezávislé realizace experimentu, informace, pro kterou výsledky těchto experimentů dopadnou jako E_1 následováno E_2, by měla být součtem informací získaných provedením těchto experimentů zvlášť.

Připustíme-li, že výše uvedená rovnost má jistou platnost, a přejeme-li si mít naši míru nezápornou a spojitou v p, což jsou oba přirozené předpoklady, zbývá nám s malou alternativou *definovat informaci* I události E kladné pravděpodobnosti jako

$$I(E) = -\log_2 P(E),$$

přičemž jsme vybrali 2 jako základ našich logaritmů, abychom zachovali soulad s moderní konvencí. (Hartley původně použil logaritmy o základu 10.) Platí totiž následující tvrzení

Věta 4.1 *Funkce $I(p)$, definovaná pro všechna $0 < p \leq 1$, splňuje podmínky $I(p) \geq 0$, pro všechna $0 < p \leq 1$, $I(p \cdot q) = I(p) + I(q)$ pro všechny $0 < p, q \leq 1$ takové, že p a q jsou pravděpodobnosti navzájem nezávislých jevů, a podmínku spojitosti vzhledem k p právě tehdy, když je tvaru*

$$I(p) = -\lambda log_2 p,$$

kde λ je kladná konstanta.

Důkaz. Ponecháme za cvičení ukázat, že každá funkce výše uvedeného tvaru splňuje všechny tři podmínky.

Abychom dokázali obrácené tvrzení, připomeňme, že z vlastnosti $I(p \cdot q) = I(p) + I(q)$ máme $I(p^n) = nI(p)$ pro všechna kladná přirozená čísla n. Speciálně tedy platí

$$I(p^{\frac{1}{n}}) = \frac{1}{n} I(p).$$

Odtud pak máme

$$I(p^{\frac{n}{m}}) = nI(p^{\frac{1}{m}} = \frac{n}{m} I(p),$$

tj. platí $I(p^q) = qI(p)$ pro všechna kladná racionální čísla q. Ze spojitosti umocňování a funkce I máme, že pro všechna kladná reálná čísla r musí platit

$$I(p^r) = rI(p).$$

Buď nyní p libovolné, pevně zvolené číslo, $0 < p < 1$. Protože každé číslo q, $0 < q < 1$ lze psát ve tvaru $q = p^{\log_p q}$, máme pak

$$I(q) = I(p^{\log_p q}) = I(p) \log_p q = I(p) \frac{\log_2 q}{\log_2 p} = -\lambda \log_2 q,$$

pro vhodnou kladnou konstantu $\lambda = -\frac{I(p)}{\log_2 p}$. Ze spojitosti funkce I plyne $I(1) = 0$.

Příklad 4.2 *Předpokládejme, že máme zdroj, který emituje řetězec binárních číslic 0 a 1, každou se stejnou pravděpodobností a nezávisle pro po sobě jdoucích číslicích. Buď E událost, že prvních n číslic jsou střídavě nuly a jedničky. Pak evidentně*

$$I(E) = -\log_2 \frac{1}{2^n} = n$$

a totéž platí pro každou předepsanou posloupnost číslic délky n.

Tedy "informačně-teoretická" jednotka informace, totiž *bit*, odpovídá přirozeně využití slova "bit", které znamená binární číslo v současné počítačové terminologii.

Rozšíříme pojem informace na to, abychom pokryli náhodné proměnné a vektory následovně. Předpokládejme, že $\mathbf{U}$ je náhodný vektor, který nabývá hodnoty $\mathbf{u}_1, \ldots, \mathbf{u}_m$, s pravděpodobnostmi $p_1, \ldots, p_m$. Pak každá z elementárních událostí $\mathbf{U} = \mathbf{u}_k$ $(1 \leq k \leq m)$ obsahuje sdruženou informaci rovnou $-\log_2 P(p_k)$ a poznamenejme, že entropie vektoru $\mathbf{U}$ je určena vztahem

$$H(\mathbf{U}) = -\sum p_k \log_2 P(p_k) = \sum p_k I(\mathbf{U} = \mathbf{u}_k),$$

tedy $H(\mathbf{U})$ má přirozenou interpretaci jako střední hodnota informace sdružené s elementárními událostmi určenými $\mathbf{U}$.

Obecněji, jsou-li $\mathbf{U}$ a $\mathbf{V}$ dva náhodné vektory, definujeme *informaci o* $\mathbf{U}$ *poskytnutou* $\mathbf{V}$ jako číslo

$$I(\mathbf{U}|\mathbf{V}) = H(\mathbf{U}) - H(\mathbf{U}|\mathbf{V}).$$

Jinak řečeno, $I(\mathbf{U}|\mathbf{V})$ vyjadřuje množství nejistoty o $\mathbf{U}$ odstraněné $\mathbf{V}$.

Zřejmě platí, že

$$I(\mathbf{U}|\mathbf{U}) = H(\mathbf{U}),$$

$$I(\mathbf{U}|\mathbf{V}) = 0 \text{ právě tehdy, když } \mathbf{U} \text{ a } \mathbf{V} \text{ jsou nezávislé.}$$

Důkaz. Výsledek plyne bezprostředně z dřívější poznámky, že $H(\mathbf{U}) = H(\mathbf{U}|\mathbf{V})$ právě tehdy, když $\mathbf{U}$ a $\mathbf{V}$ jsou nezávislé.

Poněkud udivující symetrie v I je následující výsledek, který zřejmě nemá intuitivní vysvětlení.

$$\begin{aligned} I(\mathbf{U}|\mathbf{V}) &= H(\mathbf{U}) - H(\mathbf{U}|\mathbf{V}) \\ &= H(\mathbf{U}) - [H(\mathbf{U},\mathbf{V}) - H(\mathbf{V})] \\ &= H(\mathbf{U}) + H(\mathbf{V}) - H(\mathbf{U},\mathbf{V}) \\ &= I(\mathbf{V}|\mathbf{U}). \end{aligned}$$

Cvičení 4.3

1. *Co má větší informační obsah: posloupnost deseti písmen nad abecedou o 26 písmenech nebo posloupnost 26 číslic z množiny* $\{0, 1, \ldots, 9\}$*? [Předpokládejte, že všechny posloupnosti mají stejnou pravděpodobnost.]*

2. *Ideální kostka je vržena. Ukažte, že informace o hodnotě kostky daná znalostí, že se jedná o nesložené číslo, má velikost* $\log_2 \frac{3}{2}$.

5 Závěr

Shrneme-li předchozí odstavce, ukázali jsme, že nejistota a informace jsou tytéž veličiny a odstranění nejistoty je rovno podání informace. Obě veličiny jsou měřitelné matematickým pojmem entropie, který je jednoznačně definován (až na multiplikativní konstantu) veličinou

$$H = -\lambda \sum p_i \cdot \log p_i.$$

Konvence si žádá, aby logaritmy byly brány o základu 2, v kterémžto případě je jednotka entropie *bit*.

Problémy 1.1

1. *Diskžokej má slovník o kapacitě 10 000 slov a pronáší 1000 slov náhodně (opakování je dovoleno). Ukažte, že informační obsah jeho 1000 slov je mnohonásobně menší než obrazovky televizního přijímače o 500 řádcích a 600 sloupcích, přičemž každý pixel nabývá jednu z 16 úrovní jasu.*

2. *Jsou-li X a Y diskrétní náhodné proměnné, které nabývají pouze konečného počtu hodnot, ukažte, že*
$$H(X+Y|X) = H(Y|X).$$
Ukažte, že
$$H(g(X,Y)|X) = H(Y|X)$$
neplatí obecně pro $g : \mathbf{R}^2 \to \mathbf{R}$.

3. *Je-li $(X_i : 1 \le i < \infty)$ posloupnost náhodných proměnných a Y je nějaká jiná náhodná proměnná, dokažte, že*
$$H(X_1, \ldots, X_n|Y) \le H(X_1, \ldots, X_{n+1}|Y)$$
pro každé přirozené číslo n.

4. *Statistický přehled ženatých dvojic ukazuje, že 70% mužů mělo tmavé vlasy, 25% žen bylo blondýnek a že 80% blondýnek si bere tmavovlasé muže. Kolik informace o barvě mužových vlasů je sděleno barvou vlasů jeho ženy?*

5. *Jsou-li* $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ *náhodné vektory, přičemž každý z nich nabývá pouze konečně mnoha hodnot, dokažte, že*
$$H(\mathbf{Y}|\mathbf{X}) + H(\mathbf{Z}|\mathbf{X}) \le H(\mathbf{Y},\mathbf{Z}|\mathbf{X}).$$

6. *Dokažte, že pro každý náhodný vektor* $\mathbf{Y}$ *a pro každou množinu náhodných proměnných $X_1, \ldots, X_{n+1}$ platí*
$$H(\mathbf{Y}|X_1, \ldots, X_n) \ge H(\mathbf{Y}|X_1, \ldots, X_{n+1}).$$

7. *Jsou-li X a Y dvě náhodné proměnné a f a g jsou libovolné dvě funkce, dokažte, že*
$$H(f(X), g(Y)) \le H(X,Y).$$

8. *Náhodná proměnná X má binomiální rozdělení s parametry n a p a platí, pro* $1 \le k \le n$,
$$P(X=k) = \binom{n}{k} p^k q^{n-k},$$
kde $0 < p < 1$ a $q = 1 - p$.

Dokažte, že
$$H(X) \le -n(p\log p + q\log q).$$

9. *Náhodná veličina X má geometrické rozložení a nabývá celočíselných hodnot $k = 0, 1, 2, \ldots$ s pravděpodobnostmi*

$$p_k = P(X = k) = pq^k,$$

kde $0 < p$ a $p + q = 1$. Ukažte, že rozšíříme-li pojem entropie a definujeme-li

$$H(X) = -\lambda \sum_{k=1}^{\infty} p_k \cdot \log p_k,$$

kdykoliv pravá strana konverguje, pak zejména

$$H(X) = -(p\log p + q\log q)/p.$$

10. *Nazvěme dvě náhodné proměnné X a Y* ekvivalentní, *jestliže $H(X|Y) = 0$ a $H(Y|X) = 0$. Dokažte, že jsou-li X a Y ekvivalentní a Z a Y jsou ekvivalentní, jsou i Z a X jsou ekvivalentní.*

11. *Definujme* vzdálenost *mezi dvěma náhodnými proměnnými X a Y jako*

$$d(X, Y) = H(X|Y) + H(Y|X).$$

Dokažte, že pro všechny tři náhodné proměnné X, Y, Z platí

$$d(X, Y) + d(Y, Z) \geq d(X, Z).$$

12. *Předpokládejte, že X je náhodná proměnná nabývající hodnot $v_1, \ldots, v_n$. Ukažte, že je-li $E(X) = \mu$ a X je náhodná proměnná s maximální entropií vzhledem k těmto omezením, pak*

$$p_j = P(X = v_j) = A\mathrm{e}^{-\alpha v_j},$$

kde A a α jsou konstanty určené vztahy $E(X) = \mu$ a $\sum p_j = 1$.

Poznámka: hořejší příklad je ilustrací *principu maximální entropie*: jedná se o rozšíření Laplaceova principu nedostatečné příčiny; tento je často užíván ve statistické mechanice, vytváření obrazů a podobně jako je princip pro vybrání a priori distribuce vzhledem k různým omezením. Viz např. Guiasu a Shenitzer (1985).

Řešené problémy 1.1 6. Máme ukázat, že

$$H(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) \leq H(\mathbf{X}|\mathbf{Y}).$$

Ta ale platí právě tehdy, když $H(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) - H(\mathbf{Y}, \mathbf{Z}) \leq H(\mathbf{X}|\mathbf{Y}) \iff H(\mathbf{X}, \mathbf{Z}|\mathbf{Y}) - H(\mathbf{Z}|\mathbf{Y}) \leq H(\mathbf{X}|\mathbf{Y}) \iff H(\mathbf{X}, \mathbf{Z}|\mathbf{Y}) \leq H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Z}|\mathbf{Y})$.

Stačí tedy ověřit, že

$$\begin{aligned}
H(\mathbf{X}, \mathbf{Z}|\mathbf{Y}) &= \textstyle\sum_j H(\mathbf{X}, \mathbf{Z}|\mathbf{Y} = \mathbf{b}_j) P(\mathbf{Y} = \mathbf{b}_j) \\
&\leq \textstyle\sum_j H(\mathbf{X}|\mathbf{Y} = \mathbf{b}_j) P(\mathbf{Y} = \mathbf{b}_j) + \sum_j H(\mathbf{Z}|\mathbf{Y} = \mathbf{b}_j) P(\mathbf{Y} = \mathbf{b}_j) \\
&= H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Z}|\mathbf{Y}).
\end{aligned}$$

Definujme náhodný vektor $(\mathbf{X}', \mathbf{Z}')$ předpisem

$$P\left((\mathbf{X}', \mathbf{Z}') = (\mathbf{a}_i, \mathbf{c}_l)\right) = \frac{P\left((\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_l)\right)}{P(\mathbf{Y} = \mathbf{b}_j)}.$$

Pak náhodné vektory $\mathbf{X}'$ a $\mathbf{Z}'$ jsou definované předpisy

$$P\left(\mathbf{X}' = \mathbf{a}_i\right) = \frac{P\left((\mathbf{X}, \mathbf{Y}) = (\mathbf{a}_i, \mathbf{b}_j)\right)}{P(\mathbf{Y} = \mathbf{b}_j)} \text{ a } P\left(\mathbf{Z}' = \mathbf{c}_l\right) = \frac{P\left((\mathbf{Y}, \mathbf{Z}) = (\mathbf{b}_j, \mathbf{c}_l)\right)}{P(\mathbf{Y} = \mathbf{b}_j)}.$$

Pak nutně $H(\mathbf{X}', \mathbf{Z}') \leq H(\mathbf{X}') + H(\mathbf{Z}')$, což nám sumací přes j dává $H(\mathbf{X}, \mathbf{Z}|\mathbf{Y}) \leq H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Z}|\mathbf{Y})$.

11. Máme ukázat, že

$$d(X, Y) + d(Y, Z) \geq d(X, Z).$$

Platí:

$$\begin{aligned}
d(X, Y) + d(Y, Z) &= H(X|Y) + H(Y|X) + H(Y|Z) + H(Z|Y) \\
&= H(X) - H(Y) + 2H(Y|X) + H(Y) + H(Z) + 2H(Z|Y) \\
&= H(X|Z) - H(Z|X) + 2H(Y|X) + 2H(Z|Y) \\
&\geq d(X, Z) = H(X|Z) + H(Z|X).
\end{aligned}$$

To je ale ekvivalentní s nerovností

$$H(Z|X) \leq H(Z|Y) + H(Y|X).$$

Jejím rozepsáním obdržíme $H(Z, X) - H(X) \leq H(Z, Y) - H(Y) + H(Y, X) - H(X)$, a tedy $H(Z, X) \leq H(Z, Y) - H(Y) + H(Y, X) = H(Y, Z) + H(X|Y)$,

Máme pak

$$H(Z, X) \leq H(Z, Y, X) = H(X|Y, Z) + H(Y, Z) \leq H(X|Y) + H(Y, Z). \blacksquare$$

Kapitola 2

Věta o kódování bez šumu pro zdroje bez paměti

1 Zdroje bez paměti

V této kapitole dokážeme první a snadnější ze dvou Shannonových hlavních vět pro nejjednodušší třídu zdrojů.

Stručný oxfordský slovník definuje *zdroj* jako pramen, vrchol zřídla, ze kterého proudí výstupy. Ve své plné obecnosti, je to přesně to, co uvažujeme v teorii informace, ačkoliv typicky považujeme zdroj za proud symbolů jisté konečné abecedy. Zdroj má obvykle nějaký náhodný mechanismus, který je založen na statistice situace, která je modelovaná. Tento náhodný mechanismus může být poměrně dost komplikovaný, ale my se budeme pro okamžik soustředit na následující opravdu speciální a jednoduchý příklad. Značí-li X_i i-tý symbol vytvořený zdrojem, dohodneme se pak, že, pro každý symbol a_j, pravděpodobnost

$$P(X_i = a_j) = p_j$$

je nezávislá na i a tedy je nezávislá na všech minulých nebo v budoucnosti vyslaných symbolech. Jinak řečeno, $X_1, X_2, \ldots$ je právě posloupnost identicky distribuovaných, nezávislých náhodných veličin. Takovýto zdroj nazveme *zdrojem s nulovou pamětí* nebo *zdrojem bez paměti* a jeho entropie H je definována jako

$$H = -\sum p_j \log p_j$$

kde sčítáme přes množinu j takových, že $p_j > 0$.

Cvičení 1.1

1. *Je-li $\mathcal{S}$ zdroj bez paměti s abecedou Σ,* **rozšíření řádu** *n S je zdroj bez paměti $S^{(n)}$ s abecedou $\Sigma^{(n)}$ skládající se ze všech řetězců délky n symbolů ze Σ tak, že*

pravděpodobnost každého řetězce σ je určena pravděpodobností, že je to řetězec prvních n symbolů vyslaných S. Dokažte, že $S^{(n)}$ má entropii

$$H(S^{(n)}) = nH(S).$$

2 Prefixové a jednoznačně dekódovatelné kódy

Hlavní problém řešený v této kapitole je následující. Předpokládejme, že máme zdroj bez paměti $\mathcal{S}$, který vysílá symboly z abecedy $W = \{w_1, \ldots, w_n\}$ s pravděpodobnostmi $\{p_1, \ldots, p_n\}$. Z pedagogických důvodů budeme prvky W nazývat *zdrojová slova* a ptát se na následující otázku. Je-li Σ abeceda D symbolů, jak můžeme zakódovat zdrojová slova w_i pomocí symbolů z Σ, abychom dostali co možná nejekonomičtější zakódování?

Příklad 2.1 *Předpokládejme, že zdroj $\mathcal{S}$ vysílá čtyři zdrojová slova a, b, c, d s pravděpodobnostmi*

$$p_a = 0.9, \quad p_b = 0.05, \quad p_c = p_d = 0.025.$$

Srovnáme-li pak zakódování

$$a \rightsquigarrow 0, \quad b \rightsquigarrow 111, \quad c \rightsquigarrow 110, \quad d \rightsquigarrow 101,$$

a

$$a \rightsquigarrow 00, \quad b \rightsquigarrow 01, \quad c \rightsquigarrow 10, \quad d \rightsquigarrow 11,$$

je evidentně průměrná délka zakódovaného zdroje 1.2 v prvním kódu a 2 v druhém kódu.

Formálněji, *kódování* nebo *kód* je zobrazení f z $\{w_1, \ldots, w_n\}$ do Σ^*, kde Σ^* označuje soubor konečných řetězců symbolů z Σ. *Zpráva* je každý konečný řetězec zdrojových slov a, je-li

$$m = w_{i_1} \ldots w_{i_k}$$

a je-li f kódování, pak *rozšíření* f k W^* je definováno obvyklým způsobem pomocí zřetězení

$$f(m) = f(w_{i_1}) \ldots f(w_{i_k}).$$

Kódování f je *jednoznačně dekódovatelné*, jestliže každý konečný řetězec z Σ^* je obraz nejvýše jedné zprávy. Řetězce $f(w_i)$ se nazývají *kódová slova* a přirozená čísla

$|f(w_i)|$ jsou *slovní délky* kódování f. *Průměrná délka* $\langle f \rangle$ kódování f je definovaná jako

$$\langle f \rangle = \sum_{i=1}^{m} p_i |f(w_i)|.$$

Kódování f se nazývá *bezprostředně dekódovatelné* nebo *prefixové*, jestliže neexistují různé w_i a w_j tak, že $f(w_i)$ je prefix $f(w_j)$. Zde používáme, jak lze očekávat, *prefix* v obvyklém smyslu, že pokud $x, y \in \Sigma^*$, pak x je prefix y, jestliže existuje $z \in \Sigma^*$ tak, že $xz = z$.

Prefixová kódování jsou jednoznačně dekódovatelná. Skutečně, mají silnější vlastnost, že prefixový kód může být dekódován 'on line' bez pohledu do budoucnosti.

Příklad 2.2 Předpokládejme, že $\Sigma = \{0, 1\}$ a máme čtyři zdrojová slova $w_1, .., w_4$. Prefixové kódování je

$$f(w_1) = 0, \quad f(w_2) = 10, \quad f(w_3) = 110, \quad f(w_4) = 1110.$$

Například zprávu 01101001010010 lze dekódovat jako $w_1 w_3 w_2 w_1 w_2 w_2 w_1 w_2$. (Toto je příklad toho, co je známo jako *čárkové* kódování, protože evidentně používáme nulu, abychom signalizovali konec slova.)

Ne každé jednoznačně dekódovatelné kódování je prefixové.

Příklad 2.3 Předpokládejme, že $W = \{w_1, w_2\}, \Sigma = \{0, 1\}$ a kódování g je definováno jako

$g(w_1) = 0, \quad g(w_2) = 01.$

Toto kódování není evidentně prefixové, ale lze snadno ověřit, že je jednoznačně dekódovatelné, pokud budeme postupovat zpět z konce zprávy.

Je zřejmé, že jednoznačně dekódovatelná kódování jsou o mnoho obtížnější pojem, než prefixová kódování. Překvapivě ukážeme, že můžeme omezit pozornost na prefixová kódování v našem hledání jednoznačně dekódovatelných kódování, která mají minimální průměrnou délku.

Poznámka 2.4 *Ačkoliv jsme definovali kódování jako zobrazení, často ho identifikujeme se souborem C kódových slov.*

Cvičení 2.5

1. *Ukažte, že pro každé přirozené číslo m existuje prefixové kódování nad $\{0, 1\}$, které má slova všech délek v množině $\{1, \ldots, m\}$.*

3 Kraftova a McMillanova nerovnosti

V tomto odstavci dokážeme dvě základní nerovnosti, které ospravedlňují naši dřívější poznámku, že můžeme v podstatě zapomenout na pojem jednoznačné dekódovatelnosti a omezit pozornost na prefixová kódování.

Nejdříve vyslovme nerovnosti.

KRAFTOVA NEROVNOST

Je-li Σ abeceda mohutnosti D a W obsahuje N slov, pak nutná a dostatečná podmínka, že existuje prefixové kódování $f : W \to \Sigma^*$ se slovními délkami $l_1, \ldots, l_N$ je, že platí

$$\sum_{i=1}^{N} D^{-l_i} \leq 1. \tag{2.1}$$

McMILLANOVA NEROVNOST

Je-li Σ abeceda mohutnosti D a W obsahuje N slov, pak nutná a dostatečná podmínka, že existuje jednoznačně dekódovatelné kódování $f : W \to \Sigma^*$ se slovními délkami $l_1, \ldots, l_N$ je, že platí (2.1).

Kombinací těchto dvou nerovností dostaneme:

Věta 3.1 *Jednoznačně dekódovatelné kódování s předepsanou délkou slov existuje právě tehdy, když existuje prefixový kód se stejnou délkou slov.*

DŮKAZ KRAFTOVY NEROVNOSTI

Předpokládejme, že množina $\{l_1, \ldots, l_N\}$ splňuje

$$\sum_{i=1}^{N} D^{-l_i} \leq 1.$$

Přepišme nerovnost do tvaru

$$\sum_{j=1}^{l} n_j D^{-j} \leq 1,$$

kde n_j je počet l_i rovných j a $l = \max l_i$.

Přepišme tuto nerovnost opět do tvaru

$$n_l \leq D^l - n_1 D^{l-1} - \ldots - n_{l-1} D. \tag{2.2}$$

Protože n_j jsou všechna nezáporná, postupně dostaneme z (2.2) nerovnosti

$$
\begin{aligned}
n_{l-1} &\leq D^{l-1} - n_1 D^{l-2} - \ldots - n_{l-2} D,\\
n_{l-2} &\leq D^{l-2} - n_1 D^{l-3} - \ldots - n_{l-3} D,\\
\\
n_3 &\leq D^3 - n_1 D^2 - \ldots - n_2 D,\\
n_2 &\leq D^2 - n_1 D,\\
n_1 &\leq D.
\end{aligned}
\tag{2.3}
$$

Tyto nerovnosti jsou klíč ke konstrukci kódování s danou délkou slov.

Nejdříve vyberme n_1 slov délky 1, přičemž použijeme různá písmena z Σ. Zbývá nám $D - n_1$ nepoužitých symbolů a můžeme vytvořit $(D - n_1)D$ slov délky 2 přidáním písmena ke každému z těchto symbolů.

Vyberme našich n_2 délky 2 libovolně z těchto slov a zbývá nám pak $D^2 - n_1 D - n_2$ prefixů délky 2.

Tyto lze užít pro vytvoření $(D^2 - n_1 D - n_2)D$ slov délky 3, ze kterých můžeme vybrat n_3 libovolně atd. Pokračujeme-li tímto způsobem, pokaždé je zachována vlastnost, že žádné slovo není prefixem jiného.

V každém případě zjistíme, že nerovnosti (2.3) nám dovolí provést tento výběr. Tedy skončíme s prefixovým kódováním s předepsanou délkami kódování.∎

Dokázali jsme, že numerická podmínka (2.1) je dostatečná pro existenci prefixového kódování. Ačkoliv Kraft rovněž dokázal i nutnost podmínky (2.1), jedná se o bezprostřední důsledek McMillanovy nerovnosti, kterou v dalším dokážeme. Podaný důkaz je o mnoho jednodušší, než McMillanův původní důkaz a patří Karushovi (1961).

DŮKAZ McMILLANOVY NEROVNOSTI

Předpokládejme, že máme jednoznačně dekódovatelné kódování C s délkami slov $l_1, \ldots, l_N$.

Pokud $l = \max l_i$, pak, pro každé kladné celé číslo r, máme

$$\left(D^{-l_1} + \ldots + D^{-l_N}\right)^r = \sum_{i=1}^{rl} b_i D^{-i}, \tag{2.4}$$

kde b_i je nezáporné celé číslo. Ale celá čísla b_i jsou právě počet možností, kolika způsoby lze řetězec délky i z symbolů abecedy Σ utvořit konkatenací r slov z délek vybraných z množiny $\{l_1, \ldots, l_N\}$.

Jelikož je kódování C jednoznačně dekódovatelné, každý řetězec délky i tvořený z kódových slov musí odpovídat nejvýše jedné posloupnosti kódových slov. Musíme tedy mít

$$b_i \leq D^i \qquad (1 \leq i \leq rl).$$

Dosadíme-li do (2.4), obdržíme

$$\left(D^{-l_1} + \ldots + D^{-l_N}\right)^r \leq lr.$$

Proto

$$\sum_{j=1}^{l} n_j D^{-j} \leq l^{1/r} r^{1/r},$$

a protože r bylo libovolné kladné celé číslo, dostáváme limitním přechodem $r \to \infty$ na pravé straně McMillanovu nerovnost.∎

Cvičení 3.2

1. *Jaký je maximální počet slov binárního prefixového kódování, ve kterém je maximální délka slova 7?*

4 Věta o kódování bez šumu pro zdroje bez paměti

Uvažujme nyní následující situaci. Mějme zdroj $\mathcal{S}$ bez paměti, který vysílá slova $w_1, \ldots, w_m$ s pravděpodobnostmi $p_1, \ldots, p_m$, každé vyslané slovo je vybráno nezávisle na všech jiných slovech. Náš problém je: je-li dán takovýto zdroj společně s abecedou Σ, najděte jednoznačně dekódovatelné kódování, jež má minimální průměrnou délku slov. Takovéto kódování nazýváme *kompaktní.*

Heuristický přístup k tomuto problému by mohl být následující. Zdroj $\mathcal{S}$ má entropii

$$H = -\sum p_i \log p_i.$$

Maximální entropie v abecedě o D písmenech je $\log D$. Tedy počet symbolů abecedy potřebný v průměru na zakódování slova ze zdroje by měl být asi $H/\log D$.

Tuto hrubou ideu nyní zprecizujeme.

Věta 4.1 *Má-li zdroj bez paměti entropii H, pak každé jednoznačně dekódovatelné kódování pro tento zdroj v abecedě o D symbolech musí mít délku alespoň $H/\log D$. Navíc existuje takové jednoznačně dekódovatelné kódování, které má průměrnou délku slov menší nebo rovnu $1 + H/\log D$.*

Důkaz. Předpokládejme, že máme jednoznačně dekódovatelné kódování C s délkami slov $l_1, \ldots, l_N$. Předpokládejme dále, že pravděpodobnosti emitovaných slov odpovídajících těmto délkám jsou $p_1, \ldots, p_m$. Tedy

$$H = -\sum p_i \log p_i$$

a průměrná délka kódování C je dána

$$l(C) = \sum p_i l_i.$$

Z Kraft–McMillanovy nerovnosti víme, že

$$G = \sum_{j=1}^{l} n_j D^{-j} \leq 1.$$

Definujme q_i $(1 \leq i \leq N)$ jako

$$q_i = D^{-l_i}/G,$$

tedy je $(q_1, \ldots, q_N)$ pravděpodobnostní rozdělení. Aplikujme lemma 1.2.2 a obdržíme

$$H = -\sum p_i \log p_i \leq -\sum p_i \log q_i.$$

Ale

$$\log q_i = -l_i \log D - \log G.$$

Tedy

$$H \leq \left(\sum p_i l_i\right) \log D + \left(\sum p_i\right) \log G.$$

Ale $G \leq 1$ z Kraft-McMillanovy nerovnosti a tedy, jak je požadováno,

$$H \leq l\,(C) \log D.$$

Abychom dokázali horní hranici, vybereme naše délky slov $l_1, \ldots, l_N$ podle pravidla, že pro všechna i je délka l_i minimální přirozené číslo splňující

$$p_i^{-1} \leq D^{l_i}. \tag{2.5}$$

Ale, protože $p_1 + \ldots + p_N = 1$, toto implikuje

$$\sum_{i=1}^{N} D^{-l_i} \leq 1,$$

tedy víme, že existuje jednoznačně dekódovatelné (ve skutečnosti prefixové) kódování s těmito slovními délkami.

Ale, protože 2.5 je ekvivalentní s

$$l_i \log D \geq -\log p_i$$

a l_i je minimální vzhledem k této vlastnosti, víme, že

$$l_i < 1 - \log p_i / \log D.$$

Víme tedy, že

$$l(C) = \sum p_i l_i < 1 + H/\log D.$$

∎

Cvičení 4.2

1. *Zakódujeme-li n stejně pravděpodobných slov nad binární abecedou, věta o kódování bez šumu tvrdí, že průměrná délka slova $l(C)$ každého kompaktního jednoznačně dekódovatelného kódování splňuje*

   $$\log_2 n \leq l(C),$$

   pro které hodnoty n platí rovnost?

2. *Srovnejme hranice věty o kódování bez šumu s délkou kompaktního zakódování $2^k - 1$ stejně pravděpodobných slov nad $\{0, 1\}$.*

5 Konstruování kompaktních kódování

Předpokládejme, že máme dán zdroj $\mathcal{S}$ bez paměti ze slov $w_1, \ldots, w_N$ s pravděpodobnostmi $p_1, \ldots, p_N$ a že si přejeme najít kompaktní kódování C pro $\mathcal{S}$ nad abecedou Σ. Z věty o kódování bez šumu víme, že průměrná délka musí splňovat ohraničení

$$H/\log D \leq l(C) \leq 1 + H/\log D, \tag{2.6}$$

ale snadno se vidí, že dolní hranice lze dosáhnout pouze, když p_i jsou jisté celočíselné mocniny D. Z Kraft-McMillanovy nerovnosti máme:
Jestliže existuje kompaktní jednoznačně dekódovatelné kódování o průměrné délce l, pak existuje kompaktní prefixové kódování o průměrné délce l.

Můžeme se tedy omezit na prefixová kódování. Nyní popíšeme metodu navrhnutou Huffmanem v roce 1952 pro konstruování kompaktníhu prefixového kódování pro výše uvedený zdroj $\mathcal{S}$ v případě, že Σ je binární abeceda. Nejprve dokážeme některé vlastnosti kompaktního prefixového kódování C nad $\Sigma = \{0, 1\}$. Budeme užívat $l(w)$ k označení délky slova w v C.

Lemma 5.1 *Kompaktní kódování pro zdroj s právě dvěma slovy w_1 a w_2 je*

$$w_1 \to 0, \qquad w_2 \to 1.$$

Důkaz. Zřejmé.

Lemma 5.2 *Je-li C prefixové a kompaktní kódování a $p_i > p_j$, pak $l(w_i) \le l(w_j)$.*

Důkaz. Pokud tomu tak není, vytvořme nový kód C' z C záměnou zakódování w_i a w_j. Pak průměrná délka je zmenšena a stále máme prefixové kódování.

Lemma 5.3 *Je-li C prefixové a kompaktní kódování, pak mezi všemi kódovými slovy v C maximální délky musí existovat alespoň dvě lišící se pouze v poslední číslici.*

Důkaz. Předpokládejme, že tomu tak není; pak můžeme odebrat poslední číslici z těchto všech kódových slov maximální délky a stále máme prefixové kódování, což je spor s kompaktností C.

HUFFMANŮV KÓDOVACÍ ALGORITMUS

Beze ztráty obecnosti můžeme předpokládat, že zdroj $\mathcal{S}$ má svůj systém zdrojových slov $\{w_1, \ldots, w_N\}$ uspořádaných tak, že pravděpodobnosti p_i vyslání w_i splňují

$$p_1 \ge p_2 \ge \ldots \ge p_N.$$

Huffmanova procedura konstruuje rekurzivně posloupnost zdrojů $S_0, S_1, \ldots, S_{N-2}$ tak, že $S_0 = S$ a S_k je získáno z S_{k-1} ztotožněním dvou nejméně pravděpodobných symbolů z S_{k-1} s jediným symbolem σ z S_k. Pravděpodobnost, že symbol σ je vyslán z S_k je brána jako součet pravděpodobností jeho dvou vytvářejících symbolů v S_{k-1}.

Tedy S_1 je získán z S_0 ztotožněním w_N a w_{N-1} do jednoho symbolu w_{N-1} vyskytujícího se s pravděpodobností $p_N + p_{N-1}$. V každém stavu máme zdroj s o jeden méně symboly, až po $N-2$ takových redukcích dospějeme k zdroji S_{N-2}, který má pouze dva symboly. Přechod mezi S_{j-1} a S_j lze nejlépe vidět na následujícím obrázku.

Zakódování	Pravděpodobnost	Slovo	Slovo	Pravděpodobnost	Zakódování
σ_1	q_1	v_1	u_1	q_1	σ_1
σ_2	q_2	v_2	u_2	q_2	σ_2
σ_{k-1}	q_{k-1}	v_{k-1}	u_{k-1}	q_{k-1}	σ_{k-1}
σ_{k+1}	q_k	v_k	u_k	$q_t + q_{t+1}$	σ_k
σ_{k+2}	q_{k+1}	v_{k+1}	u_{k+1}	q_k	σ_{k+1}
σ_t	q_{t-1}	v_{t-1}			
$(\sigma_k, 0)$	q_t	v_t	u_t	q_{t-1}	σ_t
$(\sigma_k, 1)$	q_{t+1}	v_{t+1}			
	S_{j-1}			S_j	

Je-li dáno zakódování $\sigma_1, \ldots, \sigma_t$ zdroje S_j, Huffmanova procedura pro nalezení zakódování zdroje S_{j-1} je následující velmi snadné pravidlo.

Předpokládejme pravděpodobnosti $q_1 \ge q_2 \ge \ldots \ge q_{t+1}$ slov z S_{j-1} jsou takové, že slovo vytvořené z v_t a v_{t+1} je slovo u_k z S_j. Pak by Huffmanovo zakódování S_{j-1} mělo být, jak je ukázáno v levém sloupci předchozí tabulky. Formálně, mělo by být zadáno pravidlem

$$\begin{aligned} v_i \to \sigma_i \qquad (1 \le i \le k-1), v_i &\to \sigma_{i+1} \qquad (k \le i \le t-1), \\ v_t \to (\sigma_k, 0), &\qquad v_{t+1} \to (\sigma_k, 1). \end{aligned}$$

Zpětným zpracováním nastartujeme naši zakódovací proceduru zakódováním dvou slov z S_{N-2} s dvěma kódovými slovy 0 a 1; pak S_{N-3} bude mít tři kódová slova atd. a budeme pokračovat ve výše uvedené proceduře, až dosáhneme Huffmanova kódu pro $S = S_0$.

Budeme ilustrovat tuto metodu na skutečně malém příkladě.

Příklad 5.4 *Předpokládejme, že $\mathcal{S}$ je zdroj s pěti zdrojovými slovy a pravděpodobnostmi (viz níže). Pak vývoj Huffmanova zakódování lze považovat za procházení šipek dopředu a pak zakódování zpátky.*

W	*P*	*C*	*W*	*P*	*C*	*W*	*P*	*C*	*W*	*P*	*C*
w_1	*0.5*	*1*	v_1	*0.5*	*1*	u_1	*0.5*	*1*	x_1	*0.5*	*0*
w_2	*0.2*	*01*	v_2	*0.2*	*01*	u_2	*0.3*	*00*	x_2	*0.5*	*1*
w_3	*0.15*	*001*	v_3	*0.15*	*000*	u_3	*0.2*	*01*			
w_4	*0.1*	*0000*	v_4	*0.15*	*001*						
w_5	*0.05*	*0001*									

W: slovo, P: pravděpodobnost C: kódování

Výsledné zakódování

$$w_1 \to 1, w_2 \to 01, w_3 \to 001, w_4 \to 0000, w_5 \to 0001$$

má průměrnou délku 1.95 na zdrojové slovo. ▌

Poznámka 5.5 *Minimálně dvakrát v horním příkladu jsme schopni provést výběr, protože dvě slova mají stejné pravděpodobnosti. Pokud tento případ nastane, dostaneme různá zakódování.*

DŮKAZ, ŽE HUFFMANŮV ALGORITMUS JE KOREKTNÍ

Z lemmatu 1 víme, že zakódování S_{N-2} dvěma symboly 0 a 1 je optimální. Důkaz bude tedy úplný, jestliže budeme schopni dokázat, že kompaktnost je zachovávána při přechodu ze zdroje S_j ke zdroji S_{j-1}.

Předpokládejme tedy, že S_j je kompaktně zakódován a že $l_1, \ldots, l_t$ jsou délky slov $\sigma_1, \ldots, \sigma_t$, ale že Huffmanovo zakódování C_{j-1} zdroje S_{j-1} není kompaktní. Existuje tedy prefixové kompaktní kódování C zdroje S_{j-1} tak, že

$$l(C) < l(C_{j-1}).$$

Dle lemmatu 3 můžeme přeuspořádat kódová slova kódování C maximální délky tak, že má-li C kódová slova $v'_1, \ldots, v'_{t+1}$ s délkami $l'_1, \ldots, l'_{t+1}$, pak $l'_1 \leq l'_2 \ldots \leq l'_{t+1}$ a $v'_t = (\sigma, 0), v'_{t+1} = (\sigma, 1)$, kde σ je nějaké kódové slovo z Σ^*.

Nechť kódování C' zdroje S_j sestává ze slov

$$v'_1, v'_2, \ldots, v'_{k-1}, \sigma, v'_k, \ldots, v'_{t+1};$$

pak C' je prefixové zakódování zdroje S_j a má průměrnou délku

$$\begin{aligned} l(C') &= p_1 l'_1 + \ldots + p_{k-1} l'_{k-1} + p'_k|\sigma| + +p_k l'_k + p_{k+1} l'_{k+1} + \ldots + p_t l'_{t-1} = \\ &= l(C) - p'_k. \end{aligned}$$

Přesněji $p'_k = p_t + p_{t+1}$, $l'_t = l'_{t+1}$, $|\sigma| = l'_t - 1 = l'_{t+1} - 1$ a proto

$$l(C') = l(C) - |\sigma|p'_k + l'_t p_t + l'_{t+1} p_{t+1} = l(C) - |\sigma|p'_k + l'_t p'_k = l(C) - p'_k(l'_t - |\sigma|) = l(C) - p'_k.$$

Ale zároveň

$$l(C_j) = l(C_{j-1}) - |\sigma|p'_k + l'_t p_t + l'_{t+1} p_{t+1} = l(C_{j-1}) - p'_k.$$

Tudíž, jestliže $l(C) < l(C_{j-1})$, pak

$$l(C') < l(C_j),$$

což je spor s kompaktností C_j. ▌

HUFFMANOVA KÓDOVÁNÍ NAD NEBINÁRNÍMI ABECEDAMI

Předpokládejme, že pracujeme namísto s binární abecedou $\{0, 1\}$ s abecedou Σ o r symbolech.

Budeme postupovat stejným způsobem. Stejně jako v binárním případě, začněme s $S = S_0$ (původní zdroj) a budeme konstruovat posloupnost zdrojů S_0, S_1, ..., S_t až skončíme u zdroje S_t obsahujícím právě r symbolů. Tento zdroj má kompaktní zakódování (jednoduše máme bijekci mezi S_t a abecedou Σ). Musíme aplikovat následující dva body:

1. Jak budem konstruovat S_{j+1} z S_j, ztotožníme ne 2, ale r nejméně pravděpodobných symbolů z S_j do jednoho symbolu z S_{j+1}. Tedy S_{j+1} má o $r-1$ symbolů méně než S_j.

2. Budeme potřebovat pro závěrečný zdroj S_t, abychom měli právě r symbolů. Abychom toho dosáhli, je nutno začít se zdrojem $\mathcal{S}$ s právě $r + t(r-1)$ symboly. Protože obecně je málo pravděpodobné, že $\mathcal{S}$ bude mít právě tento počet slov, uměle přidáme k $\mathcal{S}$ množinu S', která bude disjunktní s $\mathcal{S}$, a slova v ní obsažená budou mít nulovou pravděpodobnost. Pak klademe $S_0 = S \cup S'$ a máme $|S'| = r + t(r-1) - |S|$.

Cvičení 5.6

1. *Jaká je průměrná délka slova kompaktního kódování nad $\{0, 1\}$, jestliže máme 5 stejně pravděpodobných zdrojových slov?*

2. *Najděte kompaktní kódování nad $\{0, 1\}$ pro zdroj vysílající slova $w_1, \ldots, w_6$ s pravděpodobnostmi*

$$P(w_1) = \frac{1}{3}, P(w_2) = \frac{1}{4}, P(w_3) = \frac{1}{6}, P(w_4) = P(w_5) = P(w_6) = \frac{1}{12}$$

a porovnejte jejich průměrnou délku s horní a dolní hranicí dle věty o kódování bez šumu pro zdroje bez paměti.

3. *Najděte kompaktní kódování nad $\{0, 1, 2\}$ pro zdroj z předchozího příkladu.*

Problémy 2.1 *1. Ve hře na šachovnici má jeden z hráčů (A) uhádnout, kam jeho protivník umístil královnu. Hráči A je povoleno šest otázek, které musí být pravdivě zodpovězeny odpovědí ano/ne. Dokažte, že existuje strategie, při které může hráč A vždy vyhrát tuto hru, ale že nelze zajistit výhru, pokud má povoleno pouze pět otázek.*

2. Je-li hra z předchozího příkladu hraná na šachovnici o rozměrech $n \times n$, kolik otázek potřebuje hráč A, aby bezpečně vyhrál.

3. *Najděte průměrnou délku optimálního (kompaktního) jednoznačně dekódovatelného binárního kódu pro zdroj bez paměti, který vysílá šest slov s pravděpodobnostmi*
$$0.25, 0.10, 0.15, 0.05, 0.20, 0.25.$$
Analyzováním Huffmanova algoritmu ukažme, že zdroj bez paměti vysílá N slov a jestliže $l_1, \ldots, l_N$ jsou délky kódových slov je optimální kódování nad binární abecedou, pak $l_1 + \ldots + l_N \leq \frac{1}{2}(N^2 + N - 2)$.

4. *Máte k dispozici rovnovážnou váhu a devět zdánlivě identických mincí. Bylo Vám sděleno, že jedna mince je různá od zbývajících stejných mincí. Máte za úkol najít, o kterou minci se jedná a zda je těžší nebo lehčí. Navrhněte strategii o nejvýše 3 váženích pro řešení tohoto problému. Abychom obecně vyřešili tentýž problém pro n mincí v k váženích, je nutno, aby platilo, že $k \log 3 \geq \log 2n$. Dokažte.*

5. *V Huffmanově algoritmu pro binární abecedu aplikovaném na N zdrojových slov jsou délky slov pro konečné optimální zakódování $l_1, \ldots, l_N$. Dokažte, že*
$$l_1 + \ldots + l_N \geq N \log_2 N.$$

6. *Mějme dva hráče, hráče A a hráče B. Tito hráči hrají hru s náhodnou kostkou, která má n stran a nabývá hodnot $1, \ldots, n$ s pravděpodobnostmi $p_1, p_2, \ldots, p_n$.*
Hráč B vrhá kostkou za závěsem a hráč A má zjistit hodnotu po vržení kostky co možná nejrychleji. Hráč A se může ptát hráče B a ten mu musí pravdivě odpovídat buď ano nebo ne. Ukažte, že průměrný počet otázek pro úspěšnou strategii hráče A musí být alespoň entropie $H = -\sum p_j \log p_j$.

7. *Ukažte, že optimální zakódování zdroje s N stejně pravděpodobnými zdrojovými slovy v abecedě o D písmenech, je*
$$max\{(D^{r+1} - N - b)/(D-1), N\}$$
slov délky r, kde b je určeno vztahem
$$N + b = D + k(D-1), \quad 0 \leq b < D - 1,$$
a r je největší přirozené číslo tak, že
$$D^r \leq N + b.$$

8. *Dokažte, že následující metoda nám dává jednoznačně dekódovatelný kód pro zdro $\mathcal{S}$.*

Předpokládejme, že $\mathcal{S}$ má N zdrojových slov $s_1, s_2, \ldots, s_N$ a p_i je pravděpodobnost, že je vysláno slovo s_i, přičemž platí $p_i \geq p_{i+1}$. Nechť navíc
$$a_1 = 0, \quad a_2 = p_1, \quad a_3 = p_1 + p_2, \ldots, \quad a_N = p_1 + p_2 + \ldots + p_{N-1}.$$
Nechť m_i $(1 \leq i \leq N)$ je definováno jako nejmenší takové přirozené číslo m_i splňující $2^{-m_i} \leq p_i$.
Je-li pak a_j^* binární rozvoj čísla a_j na m_j desetinných míst, je kódování
$$s_j \mapsto a_j^*, \quad 1 \leq j \leq N$$
jednoznašně dekódovatelné pro zdroj $\mathcal{S}$. Ukažte, že se nejedná o optimální zakódování, ale že průměrná délka $\hat{l}$ kódování splňuje
$$H(S) \leq \hat{l} \leq H(S) + 1.$$

9. *Jsou-li $l_1, \ldots, l_n$ délky slov binárního Huffmanova kódování zdrojových slov majících pravděpodobnost $p_1, \ldots, p_n$, pak* redundance *kódování je definována jako*
$$r = \sum_{k=1}^{n} p_k l_k - H(p_1, \ldots, p_n).$$
Ukažte, že platí
$$r \leq p_{max} + \log[2(\log e)/e] = p_{max} + 0.086,$$
kde $p_{max} = \max_i p_i$.

Kapitola 3

Komunikace kanály se šumem

1 Komunikační systém

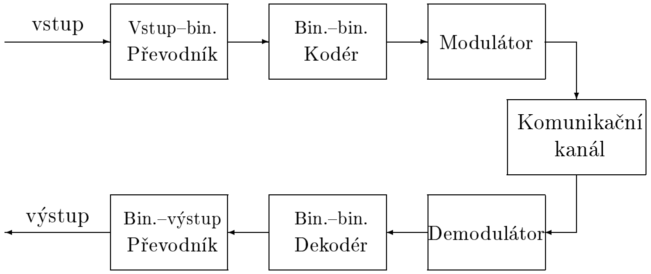
Komunikační systém je mechamismus, který zprostředkovává přenos informace od zdroje zprávy až k zařízení, které tuto informaci zpracovává. Obecně sestává z kodéru, sdělovacího (komunikačního) kanálu a následně dekodéru.

Obrázek 3.1: Blokový diagram obecného sdělovacího systému.

Zdrojová zpráva je obvykle v podobě sekvence binárních nebo desítkových číslic, nebo sekvence abecedních znaků převedených do technicky zpracovatelné podoby. Kódovací zařízení převádí tyto zprávy na signály kompatibilní se vstupem kanálu – obvykle se jedná o elektrické signály, které mají jistá omezení na velikost napětí, šířku pásma a délku trvání impulzu. Takto upravené pak vstupují do sdělovacího kanálu a jsou vystaveny šumu (tj. možnosti vzniku chyby). Výstup z kanálu vstupuje do dekodéru, jehož funkcí je rozhodnout, jakou podobu měla původní zdrojová zpráva, a tu pak přivést na výstup celého sdělovacího systému.

Většina komunikačních kanálů má konečnou kapacitu přenosu informace, tj. míru schopnosti přenášet informaci měřenou v bitech za sekundu nebo bitech na symbol. Díky vynikající teoretické práci Shannona (r. 1948) se dá ukázat, že pokud je průměrná rychlost přenosu informace menší než kapacita kanálu, je možné vybrat množinu signálů (kódových slov) takovou, že pravděpodobnost výskytu chyby při dekódování bude libovolně malá. Nicméně jakkoliv je tato teorie mocná, výsledek nevypovídá nic o tom, jak tyto signály volit, ani zda je možné je pomocí současných technických prostředků konstruovat.

Používání samoopravných kódů je pokusem výše uvedené dva problémy obejít. Ovšem celý přenos informace od zdroje až po zpracování není tak jednoduchý, jak znázorňuje (obr.3.1). Sestává z komplexnějšího sdělovacího systému; jednu takovou možnost nabízí (obr.3.2).



Obrázek 3.2: Konkrétní sdělovací systém.

Kodéry (převodníky) převádí znaky jedné abecedy na znaky abecedy jiné. Obvykle mají obě abecedy poměrně malou mohutnost – typický převod může být z desítkové do dvojkové soustavy. Modulátor na vstupu přijímá jednotlivé znaky a ke každému znaku vytváří proudový impuls, který vstupuje do kanálu. Tato operace s každým znakem zvlášť je omezením při přenosu informace a způsobí tak ztrátu kapacity kanálu.

Demodulátor provádí inverzní operaci. Ke každému obdrženému impulsu hledá znak tak, aby pravděpodobnost přenosové chyby byla co nejmenší. A opět jako při modulaci, i zde individuální modulace způsobuje ztrátu kapacity.

2 Diskrétní kanál bez paměti

Ve svém nejširším smyslu lze komunikační kanál považovat za černou skříňku, která akceptuje řetězce symbolů ze vstupní abecedy Σ_1 a vysílá řetězce symbolů z výstupní abecedy Σ_2.

Můžeme zřejmě tvrdit jen málo o takovéto struktuře. Omezme pozornost na *diskrétní kanál bez paměti*, který je charakterizován vstupní abecedou $\Sigma_1 = \{a_1, \ldots, a_m\}$, výstupní abecedou $\Sigma_2 = \{b_1, \ldots, b_n\}$ a *maticí P kanálu*

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & \cdots & p_{1n-1} & p_{1n} \\ p_{21} & p_{22} & \cdots & \cdots & p_{2n-1} & p_{2n} \\ \vdots & \vdots & \ldots & \ldots & \vdots & \vdots \\ p_{m-11} & p_{m-12} & \cdots & \cdots & p_{m-1n-1} & p_{m-1n} \\ p_{m1} & p_{m2} & \cdots & \cdots & p_{mn-1} & p_{mn} \end{pmatrix}.$$

Způsob používání kanálu je následující: každá posloupnost $(u_1, u_2, \ldots, u_N)$ symbolů ze vstupní abecedy Σ_1 na vstupu se převede na posloupnost $(v_1, v_2, \ldots, v_N)$ téže délky symbolů z výstupní abecedy Σ_2 na výstup tak, že

$$P(v_k = b_j | u_k = a_i) = p_{ij} \quad (1 \le i \le m, 1 \le j \le n),$$

a to nezávisle pro každé k.

Implicitně je ve výše uvedeném obsaženo, že pro každé i, $1 \le i \le m$ platí

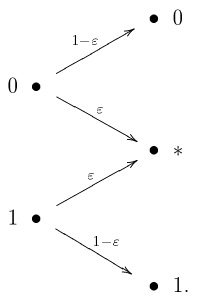
$$\sum_j p_{ij} = 1.$$

Matice P s nezápornými hodnotami taková, že součet prvků v každém řádku je roven 1, se nazývá *stochastická matice*; v teorii náhodných procesů mluvíme o matici přechodu markovského řetězce. Je často užitečné reprezentovat kanál pomocí diagramu, jako je tomu např v násl. příkladu.

Příklad 2.1 Binární vypouštěcí kanál *má vstupní abecedu* $\Sigma_1 = \{0, 1\}$, *výstupní abecedou* $\Sigma_2 = \{0, 1, *\}$ *a* maticí P kanálu

$$P = \begin{pmatrix} 1-\varepsilon & 0 & \varepsilon \\ 0 & 1-\varepsilon & \varepsilon \end{pmatrix}.$$

Diagram odpovídající tomuto kanálu má tvar

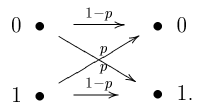


To odpovídá situaci, pro kterou má každý symbol pravděpodobnost ε, že se špatně přenese a to na $*$. Ale jak 1 tak 0 nelze navzájem zaměnit.

Příklad 2.2 *Nejpoužívanější kanál v tomto modelu komunikace je* binární symetrický kanál *má vstupní abecedu* $\Sigma_1 = \{0, 1\}$, *výstupní abecedou* $\Sigma_1 = \{0, 1\}$ *a* maticí P kanálu

$$P = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}.$$

Diagram odpovídající tomuto kanálu má tvar



Jinak řečeno, to odpovídá situaci, pro kterou má každý symbol x pravděpodobnost p, že se špatně přenese a to na $1 - x$. Často budeme psát $q = 1 - p$ bez dalšího komentáře.

Rozšíření diskrétních kanálů bez paměti

Uvažme diskrétní kanál bez paměti se vstupní abecedu Σ_1, výstupní abecedou Σ_2 a *maticí P kanálu* . r*-té rozšíření* tohoto kanálu je diskrétní kanál bez paměti se vstupní abecedu $\Sigma_1^{(r)}$, výstupní abecedou $\Sigma_2^{(r)}$ a *maticí $P^{(r)}$ kanálu* , která je definována následovně:

Souřadnice (i, j) matice $P^{(r)}$ odpovídající vstupu

$$\sigma_i = \alpha_1 \alpha_2 \ldots \alpha_r$$

s $\alpha_k \in \Sigma_1$, a výstupu

$$\tau_j = \beta_1 \beta_2 \ldots \beta_r$$

s $\beta_k \in \Sigma_2$, je

$$(P^{(r)})_{ij} = p(\beta_1|\alpha_1) p(\beta_2|\alpha_2) \ldots p(\beta_r|\alpha_r),$$

kde $p(\beta_k|\alpha_k)$ je pravděpodobnost, že v kanálu s maticí P je obdržen symbol β_k za předpokladu odeslání α_k.

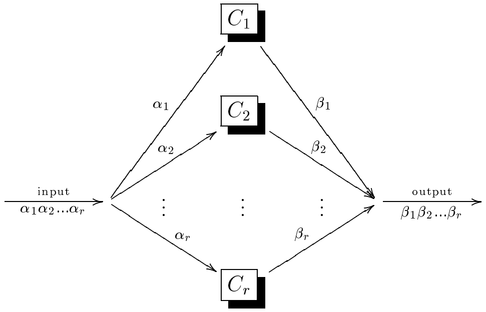
Příklad 2.3 *Druhé rozšíření* binárního symetrického kanálu *s* maticí P kanálu

$$P = \begin{pmatrix} q & p \\ p & q \end{pmatrix}$$

má tvar

$$P^2 = \begin{pmatrix} q^2 & qp & pq & p^2 \\ qp & q^2 & p^2 & pq \\ pq & p^2 & q^2 & qp \\ p^2 & pq & qp & q^2 \end{pmatrix} = \begin{pmatrix} qP & pP \\ pP & qP \end{pmatrix}.$$

Alternativní způsob jak můžeme přemýšlet o r-té extenzi je, že kanál C považujeme za r kopií C operujících nezávisle a paralelně dle níže uvedeného.

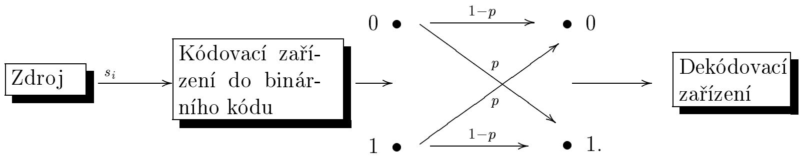


Cvičení 2.4 *1. Zpráva sestávající z N binárních číslic je přenesena binárním symetrickým kanálem mající pravděpodobnost chyby přenosu p. Ukažte, že očekávaný počet chyb je Np.*

3 Spojení zdroje s kanálem

Uvažme následující situaci: máme zdroj bez paměti $\mathcal{S}$, který vysílá symboly (zdrojová slova) $s_1, \ldots, s_N$ s pravděpodobnostmi $p_1, \ldots, p_N$.

Zdroj je spojen s binárním symetrickým kanálem s pravděpodobností chyby následovně:



Budeme předpokládat, že zakódování do binárního kódu proběhne bez šumu a že způsob zakódování je znám dekódovacímu zařízení.

Předpokládejme pro jednoduchost, že $N = 8$; za symboly můžeme považovat např. písmena abecedy, různé měny nebo cokoliv jiného. Efektivní (tj. kompaktní) zakódování ve smyslu předchozího odstavce je pak

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
000	100	010	001	110	101	011	111.

Zpráva je řetězec zdrojových slov s_i, která jsou postupně zakódovaná, přenesená a pak dekódovaná. Tudíž, dle výše uvedeného kódovacího schématu, je pravděpodobnost, že jisté slovo je správně přeneseno, rovna q^3. Pravděpodobnost, že zpráva o n slovech je korektně přenesena, je q^{3n}.

Lze to provést lépe? Odpověď je samozřejmě ano, jinak bychom se vůbec neptali. Zajímavé otázky jsou (a) jak moc lépe a (b) na čí náklady?

Příklad 3.1 *Uvažme výše uvedený příklad s osmi stejně pravděpodobnými zdrojovými slovy a předpokládejme, že použijeme zdvojené zakódování následovně:*

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
000000	100100	010010	001001	110110	101101	011011	111111.

Pokud dekódovací zařízení přijme pravidlo, že bude pouze dekódovat v případě, že první tři symboly a druhé tři symboly jsou totožné, a jinak „zavolá o pomoc“, pravděpodobnost, že se vyskytne chyba a zůstane neobjevena, se drasticky redukuje. Zajisté budeme platit podstatnou cenu tím, že jsme snížili poměr přenosu faktorem 2. navíc se jedná o čistě detekční systém, který nebude využitelný v případě, že se dekódovací zařízení nemůže kontaktovat s kódovacím zařízením a požádat ho o znovuposlání slova, u kterého byla detekována chyba.

Zbývající část této kapitoly je věnována způsobu získání dostatečné míry přenosu kanálu se šumem bez příliš velkého prodloužení zprávy.

Cvičení 3.2 *Jednoduchý způsob* detekování *nejvýše jedné chyby je použít zařízení přidávajícího* kontrolu parity*, abychom měli zajištěno, že součet čísel v přenášeném slově je sudý. Tedy kontrola parity z výše uvedeného příkladu má tvar*

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0000	1001	0101	0011	1100	1010	0110	1111.

Ukažte, že jestliže přeneseme kód s kontrolou parity binárním symetrickým kanálem, pravděpodobnost, že není objevena chyba, je rovna $6p^2(1-p)^2 + p^4$, kde p je pravděpodobnost výskytu chyby při přenosu kanálem.

4 Kódování a dekódovací pravidla

Buď dán kanál bez paměti se vstupní abecedou $\Sigma_1 = \{a_1, \ldots, a_m\}$ a výstupní abecedou $\Sigma_2 = \{b_1, \ldots, b_k\}$. *Kód délky n* je libovolný systém $\mathcal{C}$ různých posloupností délky n symbolů ze Σ_1. Prvky z $\mathcal{C}$ se nazývají *kódová slova*. Je-li dán kód délky n s kódovými slovy $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N$, *dekódovací pravidlo* je libovolný rozklad množiny možných obdržených posloupností do disjunktních množin $R_1, R_2, \ldots, R_N$ se zřejmou interpretací toho, že je-li obdržená posloupnost $\mathbf{y}$ prvkem množiny R_j, je $\mathbf{y}$ dekódováno jako kódové slovo $\mathbf{c}_j$.

Formálně vzato, předpokládáme-li, že kód $\mathcal{C}$ neobsahuje např. symbol "?" jakožto kódové slovo, *rozhodovací (dekódovací) pravidlo* pro kód $\mathcal{C}$ je funkce $f : \Sigma_2^n \to \mathcal{C} \cup \{?\}$. Aplikaci dekódovacího pravidla nazýváme *dekódování*. Je-li $\mathbf{y}$ (obdržené) slovo v Σ_2^n, pak rozhodovací pravidlo *dekóduje* $\mathbf{y}$ jakožto kódové slovo $f(\mathbf{y})$ nebo v opačném případě nahlásí *dekódovací chybu*, jestliže $f(\mathbf{y}) =?$.

Výběr dekódovacího pravidla je podstatný k úspěchu každého komunikačního systému. Jako extrémní příklad je snadné zkonstruovat dekódovací pravidlo, které zcela zničí bezchybnost kanálu bez šumu.

Příklad 4.1 *Předpokládejme, že máme zdroj s právě dvěma zdrojovými slovy s_1 a s_2, který můžeme zakódovat pro přenos binárním symetrickým kanálem jako*

$$s_1 \mapsto 000 = \mathbf{c}_1, \qquad s_2 \mapsto 111 = \mathbf{c}_2.$$

Máme pak osm možných obdržených zpráv. Možné dekódovací pravidlo by mohlo být dekódovat zprávu jako s_1, pokud obsahuje více nul než jedniček. Méně citlivé pravidlo by mohlo být dekódovat zprávu jako s_1, pouze když obdržená zpráva byla 000. A priori, každé z těchto pravidel má stejnou váhu, i s pravidlem: dekódujte každé obdržené slovo jako s_1!

Naší snahou bude najít dekódovací pravidlo, které maximalizuje pravděpodobnost správného dekódování tj. pravděpodobnost, že $\mathbf{x} = f(\mathbf{y})$ je opravdu to kódové slovo $\mathbf{c}$, které bylo odesláno. Poznamenejme, že příjemce nemá žádnou možnost zjistit, zdali dekódovací proces opravdu dekódoval správně.

Pravděpodobnost správného dekódování lze vyjádřit mnoha způsoby. Použijeme-li například formuli úplné pravděpodobnosti, obdržíme následující dva vztahy:

$$P(\text{správné dekódování}) = \sum_{\mathbf{c}\in\mathcal{C}} P(\text{správné dekódování}|\mathbf{c}\text{ odesláno})P(\mathbf{c}\text{ odesláno}), \tag{3.1}$$

vztahujeme-li podmínku na množinu kódových slov resp.

$$P(\text{správné dekódování}) = \sum_{\mathbf{y}\in\Sigma_2^n} P(\text{správné dekódování}|\mathbf{y}\text{ obdrženo})P(\mathbf{y}\text{ obdrženo}), \tag{3.2}$$

vztahujeme-li podmínku na množinu slov ze Σ_2^n.

Poznamenejme, že vztah (3.1) explicitně obsahuje pravděpodobnosti $P(\mathbf{c}$ odesláno$)$, že různá kódová slova byla poslána pomocí kanálu. Tyto pravděpodobnosti nejsou nic jiného než pravděpodobnosti zdroje $\mathcal{C}$. Mluvíme pak o *vstupním rozdělení kanálu*. Přitom (3.2) rovněž obsahuje vstupní rozdělení, protože pravděpodobnost, že dané slovo $\mathbf{y}$ je obdrženo, obvykle závisí na tom, které kódové slovo bylo odesláno.

Nechť f je dekódovací pravidlo pro kód $\mathcal{C}$. Je-li odesláno kódové slovo $\mathbf{c}$, pak správné dekódování nastane právě tehdy, když $f(\mathbf{y}) = \mathbf{c}$ pro obdržené slovo $\mathbf{y}$. Platí tedy

$$P(\text{správné dekódování}|\mathbf{c}\text{ odesláno}) = \sum_{\mathbf{y}, f(\mathbf{y})=\mathbf{c}} P(\mathbf{y}\text{ obdrženo}|\mathbf{c}\text{ odesláno}). \tag{3.3}$$

Provedeme-li substituci do (3.1), obdržíme

$$P(\text{správné dekódování}) = \sum_{\mathbf{c}\in\mathcal{C}, \mathbf{y}, f(\mathbf{y})=\mathbf{c}} P(\mathbf{y}\text{ obdrženo}|\mathbf{c}\text{ odesláno})P(\mathbf{c}\text{ odesláno}). \tag{3.4}$$

Tato dvojnásobná suma není však vždy zcela příhodná. Přitom vztah (3.2) nám podává vhodnější návod, jak obdržet dobré dekódovací pravidlo. Podle dekódovacího pravidla f je obdržené slovo $\mathbf{y}$ dekódováno správně, jestliže odesláné slovo bylo $f(\mathbf{y})$. Platí tedy

$$P(\text{správné dekódování}|\mathbf{y}\text{ obdrženo}) = P(f(\mathbf{y})\text{odesláno}|\mathbf{y}\text{ obdrženo}) \tag{3.5}$$

a přitom se ve výše uvedeném výrazu nevyskytuje žádná suma. Dosaďme do vztahu (3.2). Pak máme

$$P(\text{správné dekódování}) = \sum_{\mathbf{y}\in\Sigma_2^n} P(f(\mathbf{y})\text{ odesláno}|\mathbf{y}\text{ obdrženo})P(\mathbf{y}\text{ obdrženo}). \tag{3.6}$$

Pravděpodobnost správného kódování lze maximalizovat tím, že budeme postupovat podle takového dekódovacího pravidla, které maximalizuje každou z podmíněných pravděpodobností

$$P(f(\mathbf{y})\text{ odesláno}|\mathbf{y}\text{ obdrženo}).$$

Jinak řečeno, za předpokladu, že jsme obdrželi $\mathbf{y}$, rozhodneme se tak, že kódové slovo, které bylo posláno, je to nejpravděpodobnější, které mohlo být odesláno. To jde konkrétně zajistit tak, že se procházíme zpětnými kanálovými pravděpodobnostmi

$$P(\mathbf{c}_1 \text{ odesláno}|\mathbf{y} \text{ obdrženo}), \ldots, P(\mathbf{c}_N \text{ odesláno}|\mathbf{y} \text{ obdrženo})$$

a vybereme kódové slovo $\mathbf{c}_i$ s největší pravděpodobností.

Toto pravidlo se nazývá pravidlo *ideálního pozorovatele* neboli *pravidlo minimální chyby*. Nicméně, přepis těchto podmíněných pravděpodobností nám ukazuje, že tyto podmíněné pravděpodobnosti nelze použít bez znalosti pravděpodobností výskytu kódových slov $\mathbf{c}_j$. Máme totiž podle Bayesovy věty:

$$P(\mathbf{c} \text{ odesláno}|\mathbf{y} \text{ obdrženo}) = \frac{P(\mathbf{y} \text{ obdrženo}|\mathbf{c} \text{ odesláno})P(\mathbf{c} \text{ odesláno})}{\sum_{k=1}^{N} P(\mathbf{y} \text{ obdrženo}|\mathbf{c}_k \text{ odesláno})P(\mathbf{c}_k \text{ odesláno})}. \quad (3.7)$$

V praxi je to vážná nevýhoda. Totiž, abychom určili dekódovací funkci, musíme znát s jakou pravděpodobností jsou kódová slova posílána pomocí kanálu tj. musíme znát jistou informaci o zprávě, což není zrovna vždy možné.

Toto, společně se skutečností, že není snadné toto pravidlo aplikovat v případě, kdy máme velký počet kódových slov, opravňuje užití následujícího pravidla nazývaného *pravidlem maximální pravděpodobnosti* (maximum-likelihood (ML)). Toto pravidlo dekóduje každý obdržený vektor $\mathbf{y}$ do kódového slova $\mathbf{c}_j$ tak, že maximalizuje

$$P(\mathbf{y} \text{ obdrženo}|\mathbf{c}_j \text{ odesláno}). \quad (3.8)$$

Pro ty, kteří jsou obeznámeni s odhady maximální pravděpodobnosti ve statistice, je analogie zřejmá.

Za předpokladu nedostatku informace o pravděpodobnostech různých kódových slov máme následující:

Mají-li kódová slova stejnou pravděpodobnost, pak pravidlo maximální pravděpodobnosti splývá s pravidlem ideálního pozorovatele. (3.9)

Důkaz je snadný. Totiž platí $P(\mathbf{c} \text{ odesláno}) = \frac{1}{N}$. Tedy platí dle (3.7)

$$P(\mathbf{c} \text{ odesláno}|\mathbf{y} \text{ obdrženo}) = \frac{P(\mathbf{y} \text{ obdrženo}|\mathbf{c} \text{ odesláno})}{\sum_{k=1}^{N} P(\mathbf{y} \text{ obdrženo}|\mathbf{c}_k \text{ odesláno})}. \quad (3.10)$$

Odtud pak máme, že maximum na pravé straně obdržíme právě tehdy, když budeme mít maximum na levé straně. ▌

Hammingova vzdálenost

V hlavní části této přednášky budeme pracovat s binárním symetrickým kanálem. Pro tento kanál má pravidlo maximální pravděpodobnosti obzvlášť snadnou implementaci.

Nechť V_n označuje množinu všech posloupností délky n složených z nul a jedniček a, pokud to bude nutné, považujme V_n za vektorový n-dimenzionální prostor nad tělesem celých čísel modulo 2. Jsou-li $\mathbf{x}$ a $\mathbf{y}$ vektory z V_n, definujme *Hammingovu vzdálenost* $\mathrm{d}(\mathbf{x}, \mathbf{y})$ mezi $\mathbf{x}$ a $\mathbf{y}$ jako počet míst, ve kterých se $\mathbf{x}$ a $\mathbf{y}$ liší.

Pro binární symetrický kanál je přirozeným dekódovacím pravidlem pravidlo *minimální vzdálenosti*, totiž: dekódujme každý obdržený vektor $\mathbf{y}$ do kódového slova $\mathbf{c}_j$, že které má kódové slovo $\mathbf{c}$, jež má minimální Hammigovu vzdálenost od $\mathbf{y}$: pokud je vícero takových slov, vybereme $\mathbf{c}_j$ libovolně.

$$P(\mathbf{y} \text{ obdrženo}|\mathbf{c}_j \text{ odesláno}). \quad (3.11)$$

Následující snadný výsledek nám tvrdí:

Věta 4.2 *Pro binární symetrický kanál s pravděpodobností chyby $p \le \frac{1}{2}$ je dekódovací pravidlo minimální vzdálenosti ekvivalentní k pravidlu maximální pravděpodobnosti.*

Důkaz. Pro všechny vektory $\mathbf{x}$ a $\mathbf{y}$ z V_n s vlastností $d(\mathbf{x}, \mathbf{y}) = d$ platí

$$P(\mathbf{y} \text{ obdrženo}|\mathbf{x} \text{ odesláno}) = p^d q^{n-d}.$$

Pokud $p < \frac{1}{2}$, tento výraz nabývá maxima, je-li d minimální. To ale zřejmě stačí k tomu, že pevné slovo $\mathbf{y}$ dekódujeme jako to kódové slovo, které má nejmenší vzdálenost od slova $\mathbf{y}$. Obráceně, vezmeme-li jako rozkódování pevného slova $\mathbf{y}$ kódové slovo minimální vzdálenosti, je výše uvedená pravděpodobnost maximální. ▌

Cvičení 4.3

1. *Nechť kód sestává ze čtyř kódových slov* $\mathbf{c}_1 = 1000$, $\mathbf{c}_2 = 0110$, $\mathbf{c}_3 = 0001$ *a* $\mathbf{c}_4 = 1111$. *Pravděpodobnosti výskytu těchto kódových slov jsou dány jako*

$$P(\mathbf{c}_1) = P(\mathbf{c}_2) = \frac{1}{3}, \quad P(\mathbf{c}_3) = P(\mathbf{c}_4) = \frac{1}{6}$$

Používáte-li pro přenos binární symetrický kanál s pravděpodobností chyby $\frac{1}{10}$ *a obdržíte na výstup vektor* 1001, *jak by jste se rozhodoval při*

(a) *použití pravidla ideálního pozorovatele,*

(b) *použitím pravidla maximální pravděpodobnosti?*

2. *Dokažte tvrzení 3.9 tj. že v případě, že všechna kódová slova stejnou pravděpodobnost, pravidlo maximální pravděpodobnosti splývá s pravidlem ideálního pozorovatele.*

5 Kapacita kanálu

Jak už napovídá jméno, kapacita komunikačního kanálu je míra jeho schopnosti přenášet informaci. Formální definice je motivována níže uvedeným:

Předpokládejme, že máme diskrétní kanál bez paměti se vstupní abecedou $\Sigma_1 = \{a_1, \ldots, a_m\}$, výstupní abecedou $\Sigma_2 = \{b_1, \ldots, b_n\}$ a maticí P kanálu

$$P = [p_{ij}] = P(\mathbf{b}_j \text{ obdrženo}|\mathbf{a}_i \text{ odesláno}).$$

Přidáme-li k tomuto kanálu zdroj $\mathcal{S}$ bez paměti, který vysílá symboly $a_1, \ldots, a_m$ s pravděpodobnostmi $p_1, \ldots, p_m$, pak výstup kanálu můžeme považovat za zdroj $\mathcal{J}$ bez paměti, který vysílá symboly $b_1, \ldots, b_n$ s pravděpodobnostmi $q_1, \ldots, q_n$, kde

$$\begin{aligned} q_j &= \textstyle\sum_{i=1}^m P(\mathbf{b}_j \text{ obdrženo}|\mathbf{a}_i \text{ odesláno}) P(\mathbf{a}_i \text{ odesláno}) \\ &= \textstyle\sum_{i=1}^m p_i p_{ij}. \end{aligned}$$

Informace o $\mathcal{S}$ podaná pomací $\mathcal{J}$, definovaná v kapitole 1, je rovna

$$I(\mathcal{S}|\mathcal{J}) = H(\mathcal{S}) - H(\mathcal{S}|\mathcal{J}) = H(\mathcal{S}) + H(\mathcal{J}) - H(\mathcal{S}, \mathcal{J})$$

a je to funkce, která závisí pouze na pravděpodobnostním rozdělení $p_1, \ldots, p_m$, a matici kanálu P. Je proto přirozené definovat *kapacitu* C kanálu jako

$$C = \sup I(\mathcal{S}|\mathcal{J}), \tag{3.12}$$

kde supremum je bráno přes všechny zdroje bez paměti $\mathcal{S}$, nebo, ještě přesněji, nad všemi možnými rozděleními pravděpodobností $(p_1, \ldots, p_n)$.

Nejdříve si připomeňme, že C je dobře definováno v tom smyslu, že pouze hledáme supremum funkce $f(\mathbf{p})$, kde f je spojitá funkce na uzavřené a ohraničené podmnožině množiny $\mathbf{R}^m$ a dle základní věty analýzy má f maximum v nějakém bodě. Můžeme tedy 3.12 přepsat jako

$$C = \max I(\mathcal{S}|\mathcal{J}), \tag{3.13}$$

Dále si uvědomme, že C je kvantitativní veličina určená pouze maticí kanálu P. Můžeme ji zhruba považovat za konduktanci odporu v teorii elektrických obvodů. Její jednotky jsou pak jednotky informace nebo entropie, totiž „bity za sekundu“ nebo „bity na symbol“ v závislosti na kontextu.

Ukažme příklad, jak lze najít kapacitu kanálu.

Věta 5.1 *Kapacita binárního symetrického kanálu s pravděpodobností chyby přenosu p je určena vztahem*

$$C(p) = 1 + p \log p + q \log q, \tag{3.14}$$

kde $q = 1 - p$.

Důkaz. K usnadnění označení předpokládejme, že zdroj $\mathcal{S}$ emituje 0 s pravděpodobností α a 1 s pravděpodobností $\beta = 1 - \alpha$. Pak výstup $\mathcal{J}$ má rozdělení

$$0 \text{ s pravděpodobností } \alpha q + \beta p, \quad 1 \text{ s pravděpodobností } \beta q + \alpha p.$$

Je tedy $H(\mathcal{S}, \mathcal{J})$ právě entropie rozdělení $(\alpha q, \alpha p, \beta q, \beta q)$. Po jednoduché úpravě

$$\begin{aligned} I(\mathcal{S}|\mathcal{J}) &= p \log p + q \log q - (\alpha q + \beta p) \log(\alpha q + \beta p) \\ &\quad - (\alpha p + \beta q) \log(\alpha p + \beta q) \end{aligned}$$

.

Derivujme dle α. Pak obdržíme, že $I(\mathcal{S}|\mathcal{J})$ má maximum v případě, že $\alpha = \frac{1}{2}$ a obdržíme pak 3.14.∎

Poznamenejme, že kapacita má očekávané vlastnosti – $C(p)$ je monotonní funkce p, $0 \le p \le \frac{1}{2}$, a

$$C(0) = 1, \qquad C(\frac{1}{2}) = 0,$$

což odpovídá intuici, že, pokud $p = \frac{1}{2}$, kanál se stane dokonalým rušičem, ale že, pokud $p = 0$, máme dokonalý přenos.

Zjištění kapacity obecných kanálů je netriviální záležitost. V případě, že kanál nemá nějakou speciální vlastnost nebo není odvozen z kanálu, jehož kapacita je známa, jediný způsob, jak můžeme vypočítat kapacitu, je vyřešení problému optimalizace s omezeními, a to zejména metodou Lagrangeových multiplikátorů.

Příkladem první z těchto technik je následující výsledek.

Věta 5.2 *Má-li kanál $\mathcal{S}$ bez paměti kapacitu C, má jeho r-té rozšíření $\mathcal{S}^{(r)}$ kapacitu rC.*

Důkaz. Označme jako $C^{(r)}$ kapacitu $r-$tého rozšíření tak, že

$$C^{(r)} = \sup_{\mathbf{X}} H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}), \tag{3.15}$$

kde $\mathbf{X} = (X_1, \ldots, X_r)$ a $\mathbf{Y} = (Y_1, \ldots, Y_r)$ jsou vstupní a výstupní dvojice. Máme ale

$$H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}). \tag{3.16}$$

a

$$H(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{x}} p(\mathbf{x}) H(\mathbf{Y}|\mathbf{X} = \mathbf{x}).$$

Protože se jedná o kanál bez paměti, máme

$$H(\mathbf{Y}|\mathbf{X} = \mathbf{x}) = \sum_i H(Y_i|\mathbf{X} = \mathbf{x}) = \sum_i H(Y_i|X_i = x_i).$$

Zejména
$$\begin{aligned}H(\mathbf{Y}|\mathbf{X}) &= \textstyle\sum_{\mathbf{x}} p(\mathbf{x}) H(Y_i|X_i = x_i)\\ &= \textstyle\sum_i \sum_u H(Y_i|X_i = u)\cdot P(X_i = u).\end{aligned}$$

Tedy
$$H(\mathbf{Y}|\mathbf{X}) = \sum_i^r H(Y_i|X_i). \tag{3.17}$$

Obecně platí
$$H(\mathbf{Y}) \leq H(Y_1) + \ldots + H(Y_r),$$

a tedy celkem $C^{(r)} \leq rC$. Připomeňme, že rovnost nastává právě tehdy, když $Y_1, \ldots, Y_r$ jsou nezávislá. Toho lze dosáhnout tím, že zvolíme $X_1, \ldots, X_r$ jako nezávislé a vybráním rozdělení, při kterém bylo dosaženo kapacity C kanálu. ▮

Cvičení 5.3 *1. Vypočtěte kapacitu binárního vypouštěcího kanálu s pravděpodobností chyby ε.*

2. Uvažujeme-li kanál bez paměti s maticí
$$\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 1\end{bmatrix},$$
ukažte, že kapacity lze dosáhnout více než jedním rozdělením na vstupu. Ukažte, že rozšířením 2. řádu můžeme dosáhnout kapacity pomocí rozdělení na vstupu, kteé není součinem rozdělení na vstupu původního kanálu.

(Feinstein, 1958)

6 Věta o kódování se šumem

Již dříve jsme viděli, že můžeme dosáhnou libovolně velké spolehlivosti pouze dostatečně častým opakováním každého zdrojového symbolu. Zřejmě je tato metoda velmi časově náročná a hlavním účelem tohoto odstavce je dokázat překrásné tvrzení C. Shannona (1948), které tvrdí, že za předpokladu, že rychlost (míra) přenosu je pod kapacitou kanálu, lze dosáhnout libovolně velké spolehlivosti. Budeme se koncentrovat na binární symetrický kanál. Tyto myšlenky lze rozšířit na podstatně komplikovanější kanály, ale důležitější je plně porozumět nosným principům, než se obklopit matematickými detaily.

Buď dán kód $\mathcal{C}$ a dekódovací schéma pro $\mathcal{C}$. *Pravděpodobnost chyby* $\mathrm{e}(\mathcal{C})$ je obvykle definovaná jako průměrná pravděpodobnost chyby za předpokladu, že všechna kódová

slova byla vyslána se stejnou pravděpodobností. Jinak řečeno, máme-li M kódových slov $\mathbf{c}_1, \ldots, \mathbf{c}_M$ z $\mathcal{C}$, pak platí
$$\mathrm{e}(\mathcal{C}) = \frac{1}{M}\sum_{i=1}^{M} P(\text{nastala chyba}|\mathbf{c}_i \text{ bylo přeneseno}).$$

V případě binárních kódů můžeme předpokládat, pokud nebude jinak uvedeno, že používáme dekódovací pravidlo maximální pravděpodobnosti (=pravidlo minimální vzdálenosti), a tudíž se často budeme odvolávat na pravděpodobnost chyby kódování bez specifického připomenutí dekódovacího pravidla.

Zřejmě je předmětem příkladu najít kódy s malou průměrnou pravděpodobností chyby. Avšak, podstatně silnějším požadavkem je, že *maximální pravděpodobnost chyby* je malá. Jak lze očekávat, ta je definována jako
$$\hat{\mathrm{e}}(\mathcal{C}) = \max_i P(\text{nastala chyba}|\mathbf{c}_i \text{ bylo přeneseno}),$$
a evidentně
$$\hat{\mathrm{e}} \geq \mathrm{e}.$$

Předpokládejme proto, že máme binární symetrický kanál s pravděpodobností chyby p a tudíž kapacitou C určenou
$$C = C(p) = 1 + p\log p + (1-p)\log(1-p).$$

Dokažme následující verzi Shannonovy věty o kódování se šumem.

Věta 6.1 Shannonova věta o kódování se šumem *Buď dán binární symetrický kanál kapacity C a libovolné R, $0 < R < C$. Pak pro každou posloupnost $(M_n : 1 \leq n < \infty)$ přirozených čísel splňujících*
$$1 \leq M_n \leq 2^{Rn} \quad (1 \leq n < \infty),$$
a všechna kladná $\varepsilon > 0$, existuje posloupnost kódů $(\mathcal{C}_n : 1 \leq n < \infty)$ a přirozené číslo $N_0(\varepsilon)$ tak, že $\mathcal{C}_n$ má M_n kódových slov délky n a maximální pravděpodobnost chyby
$$\hat{\mathrm{e}}(\mathcal{C}_n) \leq \varepsilon$$
pro všechna $n \geq N_0(\varepsilon)$.

Jakým způsobem funguje tato věta. Předpokládejme, že pravděpodobnost chyby takového kanálu je taková, že kapacita kanálu $C(p) = 0.8$. Pak, je-li naše zpráva řetězec nul a jedniček, víme, že pro dostatečně velké n, položíme-li $R = 0.75$, existuje množina $2^{0.75n}$ kódových slov délky n, která mají pravděpodobnost chyby menší než libovolně předem předepsaná hranice. Tudíž, abychom zakódovali zprávu ze zdroje, postup je následující:

(a) Rozdělte zprávu do bloků délky m, přičemž m je takové, že $3\lceil \frac{n}{4} \rceil = m \geq N_0(\varepsilon)$.

(b) Zakódujte každý z těchto m-bloků do kódu $\mathcal{C}_n$ tak, že použijete kódové slovo délky $\frac{4m}{3}$ pro každý m-blok.

(b) Přeneste nově zakódovanou posloupnost kanálem.

Čeho jsme dosáhli? Podstatné redukce pravděpodobnosti chyby. Na čí náklady? Komplexnosti zakódování a menší míry přenosu: zároveň však bohužel doposud neznámé zakódování. Síla Shannonovy věty spočívá v tom, že existují takovéto kódy.

Důkaz Shannonovy věty, který chceme provést níže, závisí na dvou nerovnostech – z nich první je velmi dobře známa – její důkaz lze najít v každém elementárním textu z teorie pravděpodobnosti.

Čebyševova nerovnost

Je-li X libovolná náhodná proměnná tak, že má konečnou variaci (odchylku) $\mathrm{var}(X)$, pak pro každé $a > 0$ máme

$$P(|X - E(X)| \geq a) \leq \mathrm{var}(X)/a^2. \tag{3.18}$$

Druhá nerovnost je méně známá a má rovněž pravděpodobnostní interpretaci; lze ji vyslovit následovně.

Omezená nerovnost

Pro všechna λ, $0 \leq \lambda \leq \frac{1}{2}$, platí

$$\sum_{k=0}^{\lfloor \lambda n \rfloor} \binom{n}{k} \leq 2^{n\mathrm{h}(\lambda)}, \tag{3.19}$$

kde $\mathrm{h}(\lambda) = -[\lambda \log \lambda + (1-\lambda)\log(1-\lambda)]$.

Důkaz. Let $m = \lfloor \lambda n \rfloor$. We put $\lambda_0 = \frac{m}{n}$. Then $\lambda_0 \leq \lambda < \lambda_0 + \frac{1}{n}$. Assume $\lambda > \lambda_0 \geq 0$, $\varepsilon = \lambda - \lambda_0 > 0$. Then

$$\begin{aligned}
2^{n\mathrm{h}(\lambda)} &= 2^{-n\cdot[\lambda_0 \log \lambda + (1-\lambda_0)\log(1-\lambda)]} \,.\, 2^{-n\cdot[\varepsilon \log \lambda - \varepsilon \log (1-\lambda)]} \\
&\geq 2^{-n\cdot[\lambda_0 \log \lambda_0 + (1-\lambda_0)\log(1-\lambda_0)]} \,.\, 2^{n\varepsilon \log \frac{1-\lambda}{\lambda}} \\
&\geq 2^{n\mathrm{h}(\lambda_0)} \,.\, 2^{n\varepsilon \log \frac{1-\lambda}{\lambda}} \geq 2^{n\mathrm{h}(\lambda_0)} \,.\, 2^{\log \frac{1-\lambda}{\lambda}} \\
&\geq 2^{n\mathrm{h}(\lambda_0)} \,.\, \frac{1-\lambda}{\lambda} \geq 2^{n\mathrm{h}(\lambda_0)}
\end{aligned}$$

Můžeme tedy bez újmy na obecnosti předpokládat, že λ bylo vybráno tak, že λn je přirozené číslo. Pak můžeme psát

$$\begin{aligned}
1 = [\lambda + (1-\lambda)]^n &\geq \textstyle\sum_{k=0}^{\lambda n} \binom{n}{k} \lambda^k (1-\lambda)^{n-k} \\
&\geq (1-\lambda)^n \textstyle\sum_{k=0}^{\lambda n} \binom{n}{k} \left(\frac{\lambda}{1-\lambda}\right)^{\lambda n} \\
&= \lambda^{\lambda n}(1-\lambda)^{n(1-\lambda)} \textstyle\sum_{k=0}^{\lambda n} \binom{n}{k}.
\end{aligned}$$

Tudíž

$$\sum_{k=0}^{\lambda n} \binom{n}{k} \geq \lambda^{\lambda n}(1-\lambda)^{n(1-\lambda)} = 2^{n\mathrm{h}(\lambda)},$$

logaritmujeme-li při základu 2 a pak znovu umocníme. ▮

Důkaz věty o kódování se šumem

Nejprve popišme hrubý směr důkazu. Zvolme si pevné přirozené číslo n, a pro daný okamžik, pracujme s binárními kódy ve V_n. Předpokládejme, že se pokoušíme najít kód s M kódovými slovy $\mathbf{c}_i \in V_n$. Vybereme ta kódová slova $\mathbf{c}_i$ trochu bláznivou metodou vybráním vektorů z V_n náhodně a nezávisle na i, $(1 \leq i \leq M)$. Tomuto kódování říkáme *náhodné kódování.*

Budeme kódovat následujícím způsobem: zvolme $r > 0$ a nechť $S_r(\mathbf{y})$ definuje r-*sféru* se středem $\mathbf{y}$, tj.

$$S_r(\mathbf{y}) = \{\mathbf{z} : \mathbf{z} \in V_n, d(\mathbf{y},\mathbf{z}) \leq r\}.$$

Pak, je-li $\mathbf{y}$ obdržený vektor, můžeme dekódovat $\mathbf{y}$ jako kódové slovo $\mathbf{c}_j$, je-li $\mathbf{c}_j$ jediné kódové slovo v $S_r(\mathbf{y})$; jinak budeme dekódovat $\mathbf{y}$ jako libovolné jiné kódové slovo, např. $\mathbf{c}_1$.

Začněme nyní s vlastním důkazem. Nechť $\mathbf{Y}$ je vektor, který obdržíme, když je přenášeno kódové slovo $\mathbf{c}$ a E buď událost, že nastala chyba. Přitom chyba může nastat právě tehdy, když buď

(a) $d(\mathbf{c}, \mathbf{Y}) > r$

nebo

(b) $d(\mathbf{c}, \mathbf{Y}) \leq r$ a $d(\mathbf{c'}, \mathbf{Y}) \leq r$ pro nějaké jiné kódové slovo $\mathbf{c'}$.

Označme po řadě A a B události popsané (a) a (b). Pak $E = A \cup B$ a tudíž

$$P(E) = P(A \cup B) \leq P(A) + P(B).$$

Uvažme událost B. ta nastane, pokud platí zároveň

(i) Ne více než r chyb nastalo při přenosu

(ii) jedno z kódových slov různých od $\mathbf{c}$ je ve vzdálenosti nejvýše $\mathbf{r}$ od obdrženého vektoru $\mathbf{Y}$.

Označíme-li po řadě tyto události B_1 a B_2, máme pak, protože $B = B_1 \cap B_2$,

$$P(B) \leq P(B_2). \tag{3.20}$$

Uvažme nyní B_2; protože kódová slova jsou vybrána náhodně, pravděpodobnost, že $\mathbf{c}_i$ má vzdálenost menší nebo rovnu r od $\mathbf{Y}$ je $N_r(n)/2^n$, kde

$$N_r(n) = \sum_{k=0}^{r} \binom{n}{k} \tag{3.21}$$

je počet vektorů z V_n, které leží v $S_r(\mathbf{y})$. Tudíž pravděpodobnost, že alespoň jedno ze zbývajících $M-1$ kódových slov (různých od $\mathbf{c}$) má vzdálenost menší nebo rovnu r od obdrženého slova $\mathbf{Y}$ splňuje

$$P(B_2) \leq \frac{M-1}{2^n} \sum_{k=0}^{r} \binom{n}{k}. \tag{3.22}$$

Položme tudíž, pro všechna $\varepsilon > 0$,

$$r = \lfloor np + n\varepsilon \rfloor$$

jakožto maximální celé číslo ne větší než $np + n\varepsilon$, obdržíme pak z 3.20, 3.21, 3.22 a omezené nerovnosti, že

$$P(B) \leq \frac{M}{2^n} 2^{nh(p+\varepsilon)}. \tag{3.23}$$

Věnujme se nyní druhému typu chyb způsobenému jevem A. Poznamenejme, že, je-li U (náhodný) počet chybných symbolů vzniklých při přenosu kódového slova $\mathbf{c}$, pak máme

$$P(A) = P(U > r)$$

a U je náhodná proměnná s binomiálním rozdělením s parametry n a p. Tudíž

$$\begin{aligned} P(A) = P(U > np + n\varepsilon) &\leq P(|U - np| > \varepsilon) \\ &\leq \operatorname{var}(U)/n^2\varepsilon^2, \end{aligned}$$

dle Čebyševovy nerovnosti.

Protože U je náhodná proměnná s binomiálním rozdělením, máme

$$\operatorname{var}(U) = npq$$

a tedy úplná pravděpodobnost chyby je

$$P(E) \leq \frac{pq}{n\varepsilon^2} + M2^{-n[1-h(p+\varepsilon)]}.$$

pro dostatečně velká n. Protože kapacita $C(p) = 1 - h(p+\varepsilon)$, máme pak

$$P(E) \leq \frac{pq}{n\varepsilon^2} + M2^{-nC(p+\varepsilon)}.$$

Protože $\varepsilon > 0$, lze pravděpodobnost chyby zvolit libovolně malou pro dostatečně velké n, za předpokladu, že M jakožto funkce n, neroste rychleji než $2^{nC(p)}$.

Dokázali jsme tedy větu o kódování se šumem až na to, že jsme omezeli průměrnou pravděpodobnost chyby a ne maximální pravděpodobnost chyby. K dokončení důkazu potřebujeme dokázat, že existují kódy $\mathcal{C}_n$ s M_n kódovými slovy, kde $M_n \leq 2^{Rn}$ a mající maximální pravděpodobnost chyby $< \varepsilon$. Položme proto $\varepsilon' = \frac{1}{2}\varepsilon$ a $M'_n = 2M_n$. Poznamenejme, že protože $M_n \leq 2^{Rn}$ a $R < C$, musí existovat R' tak, že $R < R' < C$, a N'_0 tak, že pro všechna $n \geq N'_0$ platí

$$M'_n \leq 2^{nR'}$$

a tudíž existuje posloupnost kódů $\mathcal{C}'_n$ tak, že $\mathcal{C}'_n$ má M'_n kódových slov a *průměrnou* pravděpodobnost chyby $< \varepsilon'$ pro $n \geq N'_0$.

Jsou-li $\mathbf{x}_1, \ldots, \mathbf{x}_M$ kódová slova z $\mathcal{C}'_n$, znamená to, že

$$\sum_{i=1}^{M'_n} P(E|\mathbf{x}_i) \leq \varepsilon' M'_n.$$

Tedy alespoň polovina těchto kódových slov $\mathbf{x}_i$ musí splňovat

$$P(E|\mathbf{x}_i) \leq 2\varepsilon' = \varepsilon. \tag{3.24}$$

Buď $\mathcal{C}_n$ kód sestávající z M_n kódových slov splňujících 3.24; pak máme náš požadovaný kód s maximální pravděpodobností $\leq \varepsilon$.▮

Shannonovu větu lze rozšířit i pro obecné kanály bez paměti s libovolnou vstupní a výstupní abecedou. Hlavní myšlenka důkazu se nemění, totiž

(a) zakódujme zprávy náhodně,

(a) dekódujme procedurou maximální pravděpodobnosti.

Technické obtíže jsou způsobeny zejména obecným tvarem kapacity kanálu, pokud se nejedná o symetrický kanál. Případný zájemce může najít úplný důkaz (ve skutečnosti dva) pro tuto obecnou situaci v článku Ashe (1965) nebo Gallagera (1968).

Měli bychom se též zmínit o důležitosti zlepšení hranic pravděpodobnosti vzniku chyby. V našem důkazu nahoře nás pouze zajímalo to, že pravděpodobnost nastání chyby lze dosáhnout libovolně malou. K tomuto problému existuje bohatá a dostatečně technická literatura.

Například následující silnější výsledek přináleží Shannonovi (1957).

Věta 6.2 *Buď dán diskrétní kanál bez paměti kapacity C a libovolné R, $0 < R < C$. Pak existuje posloupnost kódů $(\mathcal{C}_n : 1 \leq n < \infty)$ tak, že:*
(a) $\mathcal{C}_n$ má $\lfloor 2^{Rn} \rfloor$ kódových slov délky n
(b) maximální pravděpodobnost chyby $\hat{e}(\mathcal{C}_n)$ kódování $\mathcal{C}_n$ splňuje

$$\hat{e}(\mathcal{C}_n) \leq Ae^{-Bn},$$

přičemž A a B závisí pouze na kanálu a na R.

Jinak řečeno, neexistují pouze dobré kódy, ale navíc existují kódy, jejichž pravděpodobnost chyby klesá exponenciálně.
Důkaz tohoto tvrzení přesahuje rámec přednášky.

Cvičení 6.3 *1. Binární symetrický kanál mající pravděpodobnost chyby přenosu $p = 0.05$ může přenést 800 binárních číslic za sekundu. Kolik bitů může přenést bez chyby za sekundu?*

2. Binární symetrický kanál s fyzikální kapacitou přenosu 800 číslic za sekundu může přenést 500 číslic za sekundu s libovolně malou pravděpodobností chyby. Co nám to vypovídá o pravděpodobnosti chyby tohoto kanálu?

7 Kapacita jako hranice spolehlivé komunikace

Předpokládejme, že máme diskrétní kanál bez paměti o kapacitě C bitů. Předpokládejme, že tento kanál má mechanickou rychlost jednoho bitu za sekundu. Dokážeme nyní obrácení Shannonovy věty tím, že ukážeme nemožnost přesné informace rychlostí vyšší nebo rovné než je C bitů za sekundu. Přesněji, dokážeme následující základní výsledek.

Věta 7.1 *Pro kanál bez paměti o kapacitě C a pro každé $R > C$ neexistuje posloupnost kódů $(\mathcal{C}_n : 1 \leq n < \infty)$ tak, že: $\mathcal{C}_n$ má 2^{Rn} kódových slov délky n a pravděpodobnost chyby $e(\mathcal{C}_n)$ kódování $\mathcal{C}_n$ konverguje k nule pro $n \to \infty$.*

Ve skutečnosti Wolfowitz v roce 1961 dokázal mnohem silnější výsledek – totiž, za stejných podmínek, maximální pravděpodobnost chyby konverguje k 1 pro $n \to \infty$. My však ukážeme slabší verzi, abychom dokázali, že Shannonova věta je nejlepší možná. Pro důkaz věty potřebujeme následující lemmata.

Lemma 7.2 *Buď $\mathbf{U}, \mathbf{V}, \mathbf{W}$ náhodné vektory. Pak platí*

$$H(\mathbf{U}|\mathbf{V}) \leq H(\mathbf{U}|\mathbf{V}, \mathbf{W}) + H(\mathbf{W}).$$

Důkaz. Máme dle základní identity, že

$$\begin{aligned} H(\mathbf{U}|\mathbf{V}) &= H(\mathbf{U}, \mathbf{V}) - H(\mathbf{V}) \\ &= H(\mathbf{U}, \mathbf{V}, \mathbf{W}) - H(\mathbf{W}|\mathbf{U}, \mathbf{V}) - H(\mathbf{V}) \\ &\leq H(\mathbf{U}, \mathbf{W}|\mathbf{V}), \end{aligned}$$

protože entropie je nezáporná. Ale zároveň

$$\begin{aligned} H(\mathbf{U}, \mathbf{W}|\mathbf{V}) &= H(\mathbf{U}, \mathbf{V}, \mathbf{W}) - H(\mathbf{V}, \mathbf{W}) + H(\mathbf{V}, \mathbf{W}) - H(\mathbf{V}) \\ &= H(\mathbf{U}|\mathbf{V}, \mathbf{W}) + H(\mathbf{W}|\mathbf{V}) \\ &\leq H(\mathbf{U}|\mathbf{V}, \mathbf{W}) + H(\mathbf{W}), \end{aligned}$$

což se mělo dokázat. ∎

Lemma 7.3 Fanova nerovnost *Buď C kód s M kódovými slovy $\{\mathbf{c}_1, \ldots, \mathbf{c}_M\}$ pro daný kanál bez paměti. Buď $\mathbf{X}$ náhodný vektor nabývající hodnoty v množině kódových slov. Nechť $\mathbf{Y}$ obsahuje náhodný vektorový výstup, v případě, že $\mathbf{X}$ je přeneseno kanálem a dekódováno. Pak, je-li p_E pravděpodobnost chyby (totiž $p_E = P(\mathbf{X} \neq \mathbf{Y})$), máme*

$$H(\mathbf{X}|\mathbf{Y}) \leq H(p_E, q_E) + p_E \log(M-1), \tag{3.25}$$

kde $q_E = 1 - p_E$.

Důkaz. Definujme novou náhodnou proměnnou Z jakožto

$$Z = \begin{cases} 0 & \text{pokud } \mathbf{X} = \mathbf{Y} \\ 1 & \text{pokud } \mathbf{X} \neq \mathbf{Y}. \end{cases}$$

Je tedy speciálně entropie náhodné proměnné Z rovna $H(p_E, q_E)$. Uvažme nyní uspořádanou dvojici $(\mathbf{Y}, Z)$. Zřejmě pak

$$H(\mathbf{X}|(\mathbf{Y}, Z) = (\mathbf{y}, 0)) = 0.$$

Zároveň, pokud $(\mathbf{Y}, Z) = (\mathbf{y}, 1)$, je náhodná proměnná $\mathbf{X}$ rozložena mezi $(M-1)$ kódovými slovy, která nejsou rovna $\mathbf{y}$.
Zejména tedy

$$H(\mathbf{X}|(\mathbf{Y}, Z) = (\mathbf{y}, 1)) \leq \log_2(M-1).$$

a

$$\begin{aligned} H(\mathbf{X}|(\mathbf{Y}, Z)) &= \textstyle\sum_{\mathbf{y}} H(\mathbf{X}|(\mathbf{Y}, Z) = (\mathbf{y}, 1)) \cdot P((\mathbf{Y}, Z) = (\mathbf{y}, 1)) \\ &\leq \log_2(M-1) \textstyle\sum_{\mathbf{y}} \cdot P((\mathbf{Y}, Z) = (\mathbf{y}, 1)) \\ &\leq p_E \cdot \log_2(M-1). \end{aligned}$$

Položme pak $\mathbf{U} = \mathbf{X}$, $\mathbf{V} = \mathbf{Y}$ a $\mathbf{W} = Z$. Z lemmatu 7.2 máme Fanovu nerovnost.
∎

Důkaz věty 7.1

Předpokládejme, že takováto posloupnost kódů existuje. Uvažme pak náhodný vektor $\mathbf{X}$, který nabývá hodnot v kódu $\mathcal{C}_n$ tak, že pokud položíme $R = C + \varepsilon$, $\varepsilon > 0$, máme

$$H(\mathbf{X}) = n(C + \varepsilon).$$

Protože kapacita kanálu je C, máme pak pro kódová slova délky n, že odpovídající kapacita rozšíření bez paměti je nC a tedy, označíme-li $\mathbf{Y}$ náhodný vektor výstupu odpovídající vstupnímu náhodnému vektoru $\mathbf{X}$, máme nerovnost

$$H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) \le nC,$$

takže

$$n\varepsilon = n(C + \varepsilon) - nC \le H(\mathbf{X}|\mathbf{Y}).$$

Aplikujeme-li Fanovu nerovnost, pak z toho, že máme dle předpokladu $2^{n(C+\varepsilon)}$ kódových slov, je

$$n\varepsilon \le H(\mathbf{X}|\mathbf{Y}) \le H(p_E, q_E) + p_E \log(M-1) \le H(p_E, q_E) + p_E n(C + \varepsilon),$$

tj.

$$\frac{n\varepsilon - H(p_E, q_E)}{n(C + \varepsilon)} \le p_E.$$

Necháme-li n konvergovat k nekonečnu, pak zcela jistě p_E nekonverguje k nule. Tedy takováto posloupnost kódů $\mathcal{C}_n$ nemůže existovat. ▮

Problémy 3.1 *1. V binárním symetrickém kanálu s pravděpodobností chyby $\epsilon > 0$, kódování sestává ze dvou kódových slov 000 a 111. Zjistěte při použití pravidla maximální pravděpodobnosti pravděpodobnost chyby.*

2. Trhlinová chyba *(burst error) délky k sestává z posloupnosti k symbolů, které byly všechny přeneseny nesprávně. Najděte očekávaný počet trhlinových chyb délky k, pokud je zpráva délky N přenesena binárním symetrickým kanálem s pravděpodobností chyby p.*

3. Nechť kód pro přenos binárním symetrickým kanálem, který má pravděpodobnost chyby $\varepsilon > 0$, sestává ze všech pětic nad množinou $\{0, 1\}$, které obsahují právě dvě jedničky. Jaká je pravděpodobnost, že kódové slovo 11000 se dekóduje na slovo 10001, pokud aplikujeme pravidlo minimální vzdálenosti?

4. Mějme N binárních symetrických kanálů, každý s pravděpodobností chyby p, spojených do série. Ukažte, že celková kapacita tohoto nově vzniklého kanálu je určena vztahem

$$C_N = 1 + p_N \log p_N + q_N \log q_N,$$

kde $p_N = \frac{1}{2}[1 - (q - p)^N]$, $q_N = 1 - p_N$.

5. Uvažme dva diskrétní kanály bez paměti o kapacitách C_1 a C_2 tak, že oba mají vstupní abecedu Σ_1 a výstupní abecedu Σ_2. Součinem kanálů *je kanál, jehož vstupní abeceda je $\Sigma_1^{(2)}$ a výstupní abeceda $\Sigma_2^{(2)}$, přičemž kanálové pravděpodobnosti jsou určeny vztahem*

$$p(y_1 y_2 | x_1 x_2) = p_1(y_1|x_1) p_2(y_2|x_2),$$

kde $p_i(y_i|x_i)$ je pravděpodobnost, že jsme obdželi řetězec y_i, pokud jsme odeslali řetězec x_i prostřednictvím i-tého kanálu. Dokažte, že kapacita C součinu kanálů je určena vztahem (Shannon 1957)

$$C = C_1 + C_2.$$

6. Zdroj bez paměti $\mathcal{S}$ je spojen ke kanálu $\mathcal{C}_1$ o kapacitě C_1 a výsledný výstup $\mathcal{S}_1$ je vstup ke kanálu $\mathcal{C}_2$ o kapacitě C_2 (viz níže uvedený diagram).

Ukažte, že platí

$$I(\mathcal{S}|\mathcal{S}_2) \le I(\mathcal{S}|\mathcal{S}_2) \quad \text{a} \quad I(\mathcal{S}|\mathcal{S}_2) \le I(\mathcal{S}_1|\mathcal{S}_2).$$

Kapitola 4

Kódy opravující chyby

1 Kódování a odhady

Připomeňme si následující předpoklady pro kódování. Zdroj vytváří *zprávu*, která sestává z posloupnosti zdrojových symbolů a tato zpráva je přenesena k příjemci přes kanál s možnou chybou. Přitom lze bez újmy na obecnosti předpokládat, že kanál má stejnou abecedu Σ jak na vstupu tak na výstupu. Kód $\mathcal{C}$ nad abecedou Σ je soubor posloupností symbolů ze Σ, prvky z $\mathcal{C}$ se nazývají *kódová slova* . Budeme předpokládat, že všechna kódová slova mají stejnou délku. Takovéto kódy se nazývají *blokové kódy* a při jejich použití je dekódování podstatně snazší. Pokud mají kódová slova z $\mathcal{C}$ délku n a $|\Sigma| = q$, pak mluvíme o q-árním kódu délky n (binárním kódu, pokud $q = 2$).

Zakódování zdrojové zprávy není nic jiného než přiřazení každé k-dlouhé sekvenci znaků nad zdrojovou abecedou Σ jedno kódové slovo z $\mathcal{C}$.

Během samotného dekódování se přijatá sekvence rozčlení na bloky délky n a každý se zpracovává samostatně. Jelikož přijaté n-bloky mohou mít díky chybám při přenosu obecně jinou podobu než vysílaná kódová slova, musí dekodér rozhodovat, které slovo bylo vysláno. Pokud je kód dobře navržen, je pravěpodobnost špatného rozhodnutí mnohem menší než pravděpodobnost, že libovolný kódový znak je chybně přenesen.

Proces rozhodování může být definován pomocí dekódovací tabulky. Kódová slova tvoří první řádek tabulky. Pokud jsme obdrželi kódové slovo, je logické předpokládat, že i stejné slovo bylo vysláno. Rozhodovací pravidlo pro zbylá možně přijatá slova je dáno rozdělením těchto slov do seznamů pod každým kódovým slovem, podle kterého se tato přijatá slova budou dekódovat. Tedy, každé slovo délky n nad abecedou Σ se objeví v tabulce právě jednou.

Definice. Buď u, v přirozená čísla. Řekneme, že *kód $\mathcal{C}$ určí u chyb*, jestliže, pokud každé kódové slovo změníme alespoň na jednom a ne více než u místech, nebude

výsledný řetězec kódové slovo. Řekneme, že *kód $\mathcal{C}$ určí právě u chyb*, jestliže určí u chyb a neurčí $u + 1$ chyb.

Řekneme, že *kód $\mathcal{C}$ opraví v chyb*, jestliže, pokud použijeme pravidlo minimální vzdálenosti, jsme schopni opravit alespoň v chyb a v případě, kdy se nebudeme moci rozhodnout, dostaneme na výstupu chybu v dekódování. Řekneme, že *kód $\mathcal{C}$ opraví právě v chyb*, jestliže opraví v chyb a neopraví $v + 1$ chyb.

Dále budeme předpokládat, že abychom byli schopni zjistit chyby při přenosu, bude příjemce schopen zkontrolovat přijatý řetězec proti seznamu všech kódových slov. Pokud řetězec nebude na seznamu, příjemce ví, že nastala alespoň jedna chyba, ale není schopen zjistit kolik chyb skutečně nastalo. Zároveň by mělo být jasné, že pokud obdržené slovo nebude kódové slovo, bude podle pravidla minimální vzdálenosti zpátky dekódováno, ale příjemce neví, zda se skutečně jedná o odeslané slovo. Příjemce pouze ví, že, v případě kódu opravujícího v chyb a pokud nastane nejvýše V, pak dekódovací proces bude úspěšný.

Příklad 1.1 *Chceme vysílat čtyři znaky: a, b, c, d a zpráva bude přenášena pomocí binárního blokového kódu délky* 5. *Musíme tedy zvolit čtyři kódová slova, např. 11000 pro a, 00110 pro b, 10011 pro c a 01101 pro d. Dekódování musí zahrnout všech* $2^5 = 32$ *binárních slov délky* 5. *Jedno takové dekódovací pravidlo je na* (obr.4.1).

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

Obrázek 4.1: Příklad kódové tabulky pro binární blokový kód délky 5.

Konstrukce kódu a dekódovacího schématu z příkladu 1.1 opravuje ne více než jednu chybu. V tabulce je to vždy prvních 5 slov v seznamu pod kódovým slovem. U více chyb už nemáme jistotu, že dekódování proběhne správně. Například pokud by při přenosu bloku 11000 vznikly dvě chyby vedoucí k přijetí slova 11110 na výstupu kanálu, pak toto schéma chyby odstraní. Ovšem při obdržení 11011 bude toto slovo dekódováno chybně jako 10011.

Označme dále $V_n(\Sigma)$ množinu všech posloupností délky n nad abecedou Σ a nazývejme prvky ze $V_n(\Sigma)$ *slova* nebo *vektory* . Někdy budeme psát místo $V_n(\Sigma)$ také $V_n(q)$.

Podobně jako v binárním případě je *Hammingova vzdálenost* $\mathrm{d}(\mathbf{x},\mathbf{y})$ mezi vektory $\mathbf{x}$ a $\mathbf{y}$ počet míst, ve kterých se $\mathbf{x}$ a $\mathbf{y}$ liší. *Váha* slova $\mathbf{x} = x_1x_2\cdots x_n$ je pak počet nenulových znaků slova $\mathbf{x}$, tj. $\mathrm{wt}(\mathbf{x}) = \{\mathrm{d}(\mathbf{x},\mathbf{0}) \mid \text{kde } \mathbf{0} \text{ je slovo z } n \text{ nul}\}$.

Definice. Buď $\mathbf{x} \in \mathbf{Z}_r^n$, $\rho \geq 0$. *Sférou* $\mathbf{S}_\rho^{n,r}(\mathbf{x})$ *v* $\mathbf{Z}_r^n$ *se středem* $\mathbf{x}$ *a poloměrem* ρ rozumíme množinu

$$\mathbf{S}_\rho^{n,r}(\mathbf{x}) = \{\mathbf{y} \in \mathbf{Z}_r^n : \mathrm{d}(\mathbf{x},\mathbf{y}) \leq \rho\}.$$

Objemem sféry $\mathbf{S}_\rho^{n,r}(\mathbf{x})$ nazveme číslo

$$\mathbf{V}_\rho^{n,r}(\mathbf{x}) = \mathrm{card}(\{\mathbf{y} \in \mathbf{Z}_r^n : \mathrm{d}(\mathbf{x},\mathbf{y}) \leq \rho\}),$$

tj. počet řetězců délky n, které mají Hammingovu vzdálenost od $\mathbf{x}$ nejvýše ρ. Pokud budou čísla n, r jasná ze souvislosti, budeme psát jednoduše $\mathbf{S}_\rho(\mathbf{x})$ a $\mathbf{V}_\rho(\mathbf{x})$.

Platí pak

Lemma 1.2 *Objem sféry* $\mathbf{S}_\rho^{n,r}(\mathbf{x})$ *je určen vztahem*

$$\mathbf{V}_r^n(\mathbf{x},\rho) = \sum_{k=0}^{\rho} \binom{n}{k} (r-1)^k.$$

Důkaz. Plyne z toho, že počet řetězců délky n, které mají Hammingovu vzdálenost od $\mathbf{x}$ právě k je přesně číslo

$$\binom{n}{k}(r-1)^k.$$

▌

Příklad 1.3 *Hammingova vzdálenost slov* 01110010 *a* 11110101 *je* 4 *a váha slova* 01110101 *je* 5.

Dekódování podle principu *minimální vzdálenosti* znamená, že dekódujeme obdržený vektor $\mathbf{y}$ jako to kódové slovo $\mathbf{c}$, které má minimální vzdálenost od $\mathbf{y}$, pokud máme možný výběr, vybereme libovolně.

Je-li tedy $\mathcal{C}$ kód, je *minimální vzdálenost kódu* $\mathcal{C}$ číslo

$$\mathrm{d}(\mathcal{C}) = \min \mathrm{d}(\mathbf{c}_i, \mathbf{c}_j),$$

kde je minimum bráno přes všechny navzájem různé dvojice kódových slov z $\mathcal{C}$. Pojem minimální vzdálenosti je klíčový pojem pro hodnocení kódu; dobré kódy mají rozložená kódová slova tak, že jejich minimální vzdálenost je velká. Důvod důležitosti minimální vzdálenosti je jasný z následující věty.

Věta 1.4 *Má-li kód minimální vzdálenost d, lze opravit pomocí dekódování podle pravidla minimální vzdálenosti až* $\frac{1}{2}(d-1)$ *chyb.*

Důkaz. Položme $v = \lfloor \frac{1}{2}(d-1) \rfloor$ a uvažme v-sféru bodu $\mathbf{x}$. To je množina

$$\mathbf{S}_u(\mathbf{x}) = \{\mathbf{y} : \mathrm{d}(\mathbf{x},\mathbf{y}) \leq u\}.$$

Jsou-li $\mathbf{x}, \mathbf{z}$ různá kódová slova, platí

$$\mathbf{S}_v(\mathbf{x}) \cap \mathbf{S}_v(\mathbf{z}) = \emptyset.$$

Tedy dekódování podle pravidla minimální vzdálenosti opraví až v chyb. ▌

Máme pak následující jednoduché tvrzení.

Věta 1.5 *Buďte* u, v *přirozená čísla. Pak kód* $\mathcal{C}$ *určí* u *(opraví* v*) chyb právě tehdy, když* $d(\mathcal{C}) \geq u+1$ $(d(\mathcal{C}) \geq 2v+1)$.

Důkaz. První část tvrzení je jednoduché přeformulování definice kódu určujících u chyb. Pro druhou část tvrzení jsme ukázali ve větě 1.4 implikaci zprava doleva. Předpokládejme nyní, že $\mathcal{C}$ je kód opravující v chyb a že existují dvě různá kódová slova $\mathbf{c}$ a $\mathbf{d}$ tak, že $\mathrm{d}(\mathbf{c},\mathbf{d}) = \mathrm{d}(\mathcal{C}) \leq 2v$. Budeme chtít dokázat, že za předpokladu, že jsme odeslali kódové slovo $\mathbf{c}$ a nastalo nejvýše v chyb, je přesto možné, abychom podle pravidla minimální vzdálenosti obdrželi buď chybové hlášení nebo dekódovali obdržené slovo nesprávně jako $\mathbf{d}$. To pak bude spor s tím, že $\mathcal{C}$ opravuje v chyb.

Nejdříve si uvědomme, že $\mathrm{d}(\mathbf{c},\mathbf{d}) = \mathrm{d}(\mathcal{C}) \geq v+1$. Jinak bychom totiž mohli převést $\mathbf{c}$ na $\mathbf{d}$ při nejvýše v chybách, které by zůstaly neopraveny. Můžeme pak předpokládat, že se $\mathbf{c}$ a $\mathbf{d}$ liší na prvních $k = \mathrm{d}(\mathcal{C})$ místech, přičemž $v+1 \leq k \leq 2v$ (jinak provedeme permutaci souřadnic). Uvažme nyní obdržené slovo $\mathbf{x}$, které se shoduje se slovem $\mathbf{c}$ na prvních $k-v$ pozicích, dále se shoduje se slovem $\mathbf{d}$ na dalších v pozicích a shoduje se s oběma slovy $\mathbf{c}$ a $\mathbf{d}$ na posledních $n-k$ pozicích, tj.

$$\mathbf{x} = \underbrace{x_1 \ldots x_{k-v}}_{\text{shoduje se s } \mathbf{c}} \underbrace{x_{k-v+1} \ldots x_k}_{\text{shoduje se s } \mathbf{d}} \underbrace{x_{k+1} \ldots x_n}_{\text{shoduje se s oběma}}.$$

Protože nutně $\mathrm{d}(\mathbf{c},\mathbf{x}) = v$, $\mathrm{d}(\mathbf{d},\mathbf{x}) = k-v \leq v$, je buďto $\mathrm{d}(\mathbf{c},\mathbf{x}) = d(\mathbf{d},\mathbf{x})$ (v tomto případě obdržíme chybové hlášení) nebo $\mathrm{d}(\mathbf{c},\mathbf{x}) > \mathrm{d}(\mathbf{d},\mathbf{x})$ (v tomto případě je $\mathbf{x}$ dekódováně nesprávně jako $\mathbf{d}$).▌

Definice. Pokud má kód $\mathcal{C}$ právě M kódových slov délky n a má minimální vzdálenost d, mluvíme o (n, M, d)-kódu.

Pro pevné n působí parametry M a d navzájem proti sobě tak, že zvětšení M způsobí zmenšení d a naopak.

Máme pak následující důsledek.

Důsledek 1.6 *1. (n, M, d)-kód $\mathcal{C}$ opraví právě v chyb tehdy a jen tehdy, když $d = 2v + 1$ nebo $d = 2v + 2$.*

2. Kód $\mathcal{C}$ má minimální vzdálenost $u = \mathrm{d}(\mathcal{C})$ tehdy a jen tehdy, když opraví právě $\lfloor \frac{1}{2}(u-1) \rfloor$ chyb.

Poznamenejme nyní, že určení chyby a její oprava jdou proti sobě, takže nemůžeme naráz dosáhnout jejich maximální úrovně. Uveďme si to na jednoduchém příkladě.

Příklad 1.7 *Předpokládejme nyní, že kód $\mathcal{C}$ má minimální vzdálenost d. Je to tedy kód určující $d-1$ chyb a opravující $u = \lfloor \frac{1}{2}(d-1) \rfloor$ chyb.*

Pokud použijeme $\mathcal{C}$ pouze pro určení chyb, je schopen určit až $d-1$ chyb. Z druhé strany, pokud chceme na $\mathcal{C}$ opravit chybu, kdykoliv je to možné, pak je schopen opravit až v chyb, ale není schopen určit situaci, kdy nastalo více než v a méně než d chyb. Totiž, pokud nastalo více než v chyb, můžeme podle pravidla minimální vzdálenosti „opravit“ přijatý řetězec na špatné kódové slovo a pak bude chyba nedetekovatelná.

Máme pak následující definici.

Definice. Uvažme následující strategii pro opravu/určení chyby. Buď u, v přirozená čísla. Obdržíme-li slovo $\mathbf{x}$ a pokud má nejbližší kódové slovo $\mathbf{c}$ ke slovu $\mathbf{x}$ vzdálenost nejvýše v a existuje-li pouze jediné takové kódové slovo, budeme dekódovat $\mathbf{x}$ jako kódové slovo $\mathbf{c}$. Pokud existuje více než jedno kódové slovo se stejnou minimální vzdáleností k $\mathbf{x}$ nebo má nejbližší kódové slovo vzdálenost větší než v, obdržíme na výstup chybové hlášení.

Řekneme, že *kód $\mathcal{C}$ zároveň opraví v chyb a určí u chyb*, jestliže nastala alespoň jedna a nejvýše v chyb, výše popsaná strategie opraví tyto chyby a kdykoliv nastane alespoň $u+1$ a nejvýše $u+v$ chyb, výše popsaná strategie nahlásí chybu.

Věta 1.8 *Kód $\mathcal{C}$ zároveň opraví v chyb a určí u chyb právě tehdy, když $d(\mathcal{C}) \geq 2v + u + 1$.*

Důkaz. Předpokládejme nejprve, že $d(\mathcal{C}) \geq 2v + u + 1$. Obdržíme-li slovo $\mathbf{x}$ a pokud má nejbližší kódové slovo $\mathbf{c}$ ke slovu $\mathbf{x}$ vzdálenost nejvýše v a existuje-li alespoň jedno další takové kódové slovo $\mathbf{d}$, máme

$$2v + u + 1 \leq \mathrm{d}(\mathbf{c}, \mathbf{d}) \leq \mathrm{d}(\mathbf{c}, \mathbf{x}) + \mathrm{d}(\mathbf{x}, \mathbf{d}) \leq 2v,$$

což je spor. Nutně tedy máme, že pokud obdržíme slovo $\mathbf{x}$ a nejbližší kódové slovo $\mathbf{c}$ ke slovu $\mathbf{x}$ má vzdálenost nejvýše v, je toto kódové slovo jediné s touto vlastností a

podle pravidla minimální vzdálenosti budeme správně dekódovat. Obdržíme-li slovo $\mathbf{x}$ a pokud má nejbližší kódové slovo $\mathbf{c}$ ke slovu $\mathbf{x}$ vzdálenost alespoň $v+1$ a nejvýše $u+v$, při použití výše uvedené strategie dostaneme chybové hlášení.

Předpokládejme nyní, že $\mathcal{C}$ je kód opravující v chyb a určující u chyb. Nechť dále $\mathrm{d}(\mathcal{C}) \leq 2v + u$. Nutně pak $2v + 1 \leq \mathrm{d}(\mathcal{C})$. Víme, že existují dvě různá kódová slova $\mathbf{c}$ a $\mathbf{d}$ tak, že $k = \mathrm{d}\,(\mathbf{c}, \mathbf{d}) = \mathrm{d}(\mathcal{C})$. Můžeme pak předpokládat, že se $\mathbf{c}$ a $\mathbf{d}$ liší na prvních $k = \mathrm{d}(\mathcal{C})$ místech, přičemž $2v + 1 \leq k \leq 2v + u$ (jinak provedeme permutaci souřadnic). Uvažme nyní obdržené slovo $\mathbf{x}$, které se shoduje se slovem $\mathbf{c}$ na prvních v pozicích, dále se shoduje se slovem $\mathbf{d}$ na dalších $k - v$ pozicích a shoduje se s oběma slovy $\mathbf{c}$ a $\mathbf{d}$ na posledních $n - k$ pozicích, tj.

$$\mathbf{x} = \underbrace{x_1 \dots x_v}_{\text{shoduje se s } \mathbf{c}} \underbrace{x_{v+1} \dots x_k}_{\text{shoduje se s } \mathbf{d}} \underbrace{x_{k+1} \dots x_n}_{\text{shoduje se s oběma}}.$$

Nutně pak $\mathrm{d}(\mathbf{c}, \mathbf{x}) = k - v$, $\mathrm{d}(\mathbf{d}, \mathbf{x}) = v \leq v$, $v + 1 \leq k - v \leq v + u$. Je-li tedy $\mathbf{c}$ odesláno a $\mathbf{x}$ je obdrženo, nutně je pak počet chyb při přenosu (tj. číslo $k - v$) mezi $v + 1$ a $v + u$, uvažovaná strategie by nám měla dát na výstupu chybové hlášení, ale místo toho nám dekóduje $\mathbf{x}$ nesprávně na $\mathbf{d}$. ▮

Definice. (n, M, d)-kód $\mathcal{C}$ se nazývá maximální, pokud není obsažen v žádném větším kódu se stejnou minimální vzdáleností tj. není obsažen v žádném $(n, M+1, d)$-kódu.

Je zřejmé, že pro každý kód $\mathcal{C}$ můžeme vždy najít maximální kód $\mathcal{C}'$, který jej obsahuje. Přitom platí

Věta 1.9 *(n, M, d)-kód $\mathcal{C}$ je maximální právě tehdy, když pro všechna slova $\mathbf{x}$ existuje kódové slovo $\mathbf{c}$ s vlastností $d(\mathbf{x}, \mathbf{c}) < d$.*

Důkaz. (n, M, d)-kód $\mathcal{C}$ je maximální právě tehdy, když není obsažen v žádném $(n, M+1, d)$-kódu. Předpokládejme, že existuje slovo $\mathbf{x}$ tak, že pro všechna kódová slova $\mathbf{c}$ platí $\mathrm{d}(\mathbf{x}, \mathbf{c}) \geq d$. Položíme-li $\mathcal{C}' = \mathcal{C} \cup \{x\}$, je pak evidentně $\mathcal{C}'$ $(n, M+1, d)$-kód obsahující kód $\mathcal{C}$.

Obráceně, nechť pro všechna slova $\mathbf{x}$ existuje kódové slovo $\mathbf{c}$ s vlastností $\mathrm{d}(\mathbf{x}, \mathbf{c}) < d$. Předpokládejme, že kód $\mathcal{C}$ není maximální tj. existuje $(n, M+1, d)$-kód obsahující kód $\mathcal{C}$. Vyberme slovo $\mathbf{x} \in \mathcal{C}' - \mathcal{C}$. Pak ale existuje kódové slovo $\mathbf{c} \in \mathcal{C} \subseteq \mathcal{C}'$ tak, že $\mathrm{d}(\mathcal{C}') \leq \mathrm{d}(\mathbf{x}, \mathbf{c}) < d$, spor. ▮

Poznamenejme, že pokud $(n, M, d)-$kód $\mathcal{C}$ není maximální, mohou nastat jak výhodné tak nevýhodné situace při jeho rozšíření na maximální kód $\mathcal{C}'$. Víme, že kód $\mathcal{C}'$ rovněž opraví $\lfloor \frac{1}{2}(d-1) \rfloor$ chyb, což je dobré a přitom $\mathcal{C}'$ může zakódovat více zdrojových symbolů, což je rovněž dobré. Ale zatímco $\mathcal{C}$ může být případně schopen opravit více než $\lfloor \frac{1}{2}(d-1) \rfloor$ chyb, kód $\mathcal{C}'$ nebude nikdy schopen opravit více než $\lfloor \frac{1}{2}(d-1) \rfloor$ chyb.

Příklad 1.10 *Uvažme kód* $\mathcal{C} = \{00000, 11000\}$*, který má minimální vzdálenost* 2*. Tento kód opravuje jednu chybu, ale je přitom schopen opravit další jiné chyby. Například, pokud bylo odesláno slovo* 00000 *a přijato slovo* 00111*, bude toto slovo správně dekódováno (totiž* $d(00000, 00111) = 3$*,* $d(11000, 00111) = 5$*), ačkoliv při přenosu nastaly tři chyby. Pokud ale doplníme* $\mathcal{C}$ *do maximálního kódu, bude dekódování chybné.*

Z výše uvedeného příkladu vyplývá, že maximální kódy jsou nejlepší, pokud nás u kódu pouze zajímá předem určená schopnost opravení chyby. Je tedy daleko obtížnější zkoumat pravděpodobnost chyby při dekódování u kódů, které nejsou maximální. Pro maximální kódy je to jednodušší.

Věta 1.11 *Buď* $\mathcal{C}$ (n, M, d)*-kód. Pak pro binární symetrický kanál s pravděpodobností chyby* p *je při použití dekódovacího pravidla minimální*

$$P(\text{nastala chyba při dekódování}) \leq 1 - \sum_{k=0}^{\lfloor \frac{1}{2}(d-1) \rfloor} \binom{n}{k} p^k (1-p)^{n-k}.$$

Je-li navíc kód $\mathcal{C}$ *maximální, je*

$$\sum_{k=0}^{d} \binom{n}{k} p^k (1-p)^{n-k} \leq P(\text{nastala chyba při dekódování}).$$

Důkaz. Každý (n, M, d)-kód $\mathcal{C}$ opravuje evidentně $\lfloor \frac{1}{2}(d-1) \rfloor$ chyb. Je tedy pravděpodobnost správného dekódování alespoň tak velká jako je pravděpodobnost, že nastane nejvýše $\lfloor \frac{1}{2}(d-1) \rfloor$ chyb tj.

$$P(\text{správné dekódování}) \geq \sum_{k=0}^{\lfloor \frac{1}{2}(d-1) \rfloor} \binom{n}{k} p^k (1-p)^{n-k}.$$

Máme pak

$$\begin{aligned} P(\text{nastala chyba při dekódování}) &= 1 - P(\text{správné dekódování}) \\ &\leq 1 - \textstyle\sum_{k=0}^{\lfloor \frac{1}{2}(d-1) \rfloor} \binom{n}{k} p^k (1-p)^{n-k}. \end{aligned}$$

Buď dále (n, M, d)-kód $\mathcal{C}$ maximální. Pak, je-li přeneseno slovo $\mathbf{c}$ a nastane-li d nebo více chyb, tj. $\mathrm{d}(\mathbf{c}, \mathbf{x}) \geq d$, bude nutně $\mathbf{x}$ bližší k jinému kódovému slovu $\mathbf{d} \neq \mathbf{c}$ a tedy při použití pravidla minimální vzdálenosti nastane dekódovací chyba. Protože pravděpodobnost, že nastane právě k chyb při průchodem binárním symetrickým kanálem, je

$$\binom{n}{k} p^k (1-p)^{n-k},$$

obdržíme následující dolní hranici pro pravěpodobnost dekódovací chyby

$$\sum_{k=0}^{d} \binom{n}{k} p^k (1-p)^{n-k} \leq P(\text{nastala chyba při dekódování}),$$

čímž je věta dokázána. ▮

2 Ekvivalence kódů a konstrukce nových kódů

Užitečným prostředkem pro redukci množství práce při nalezení dobrých kódů je pojem ekvivalence kódů. Předpokládejme, že máme (n, M, d)-kód $\mathcal{C}$. Přirozeným způsobem jeho prezentace je pomocí matice o rozměrech $M \times n$, přičemž řádky jsou různá kódová slova.

Předpokládejme nyní, že π je permutace množiny $\{1, 2, \ldots, n\}$ a pro každé kódové slovo $\mathbf{c} \in \mathcal{C}$ budeme aplikovat transformaci $\overline{\pi} : \mathcal{C} \to \mathcal{C}'$ definovanou předpisem $\overline{\pi}(\mathbf{c}) = (c_{\pi(1)}, \ldots, c_{\pi(n)})$. Takovou transformaci nazýváme *poziční permutací*. Podobně, je-li π permutace množiny Σ, pak pro každý index i, $1 \leq i \leq$ můžeme aplikovat transformaci $\hat{\pi}_i : \mathcal{C} \to \mathcal{C}'$ definovanou předpisem $\hat{\pi}_i(\mathbf{c})_j = \begin{cases} c_j & \text{pokud } i \neq j \\ \pi(c_i) & \text{pokud } i = j. \end{cases}$

Mluvíme pak o *symbolové permutaci*. Lze-li kód $\mathcal{C}'$ získat z kódu $\mathcal{C}$ pomocí konečné posloupnosti pozičních nebo symbolových permutací, říkáme že kód $\mathcal{C}'$ je ekvivalentní kódu $\mathcal{C}$.

Příklad 2.1 Předpokládejme, že máme kód $\mathcal{C}$ délky 5 nad abecedou $\Sigma = \{a, b, c\}$ s kódovými slovy $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ a $\mathbf{c}_4$ tak, že

$$\mathcal{C} = \begin{array}{c} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \end{array} \begin{bmatrix} a & b & c & a & c \\ b & a & b & a & b \\ b & c & c & b & a \\ c & b & a & c & a \end{bmatrix}.$$

Aplikujeme-li permutaci $(1 \mapsto 2, 2 \mapsto 3, \ldots, 5 \mapsto 1)$, obdržíme poziční permutaci a k ní odpovídající ekvivalentní kód je

$$\mathcal{C}' = \begin{array}{c} \mathbf{c}'_1 \\ \mathbf{c}'_2 \\ \mathbf{c}'_3 \\ \mathbf{c}'_4 \end{array} \begin{bmatrix} c & a & b & c & a \\ b & b & a & b & a \\ a & b & c & c & b \\ a & c & b & a & c \end{bmatrix}.$$

Podobně, aplikujeme-li permutaci $(a \mapsto b, b \mapsto c, c \mapsto a)$ na první sloupec kódu $\mathcal{C}'$, obdržíme symbolovou permutaci a k ní odpovídající ekvivalentní kód je

$$\mathcal{C}'' = \begin{array}{c} \mathbf{c}_1'' \\ \mathbf{c}_2'' \\ \mathbf{c}_3'' \\ \mathbf{c}_4'' \end{array} \left[\begin{array}{ccccc} a & a & b & c & a \\ c & b & a & b & a \\ b & b & c & c & b \\ b & c & b & a & c \end{array} \right].$$

Lemma 2.2 *Jsou-li $\mathcal{C}$ a $\mathcal{C}'$ ekvivalentní kódy, jsou množiny vzdáleností kódových slov z $\mathcal{C}$ a $\mathcal{C}'$ stejné.*

Důkaz. Protože jak poziční tak symbolová permutace zachovávají vzdálenost permutovaných slov, platí totéž i pro takovouto posloupnost permutací. ∎

Lemma 2.3 *Je-li $\mathcal{C}$ kód délky n a $\mathbf{u}$ vektor délky n nad stejnou abecedou, pak existuje kód $\mathcal{C}'$, který je ekvivalentní s $\mathcal{C}$ a obsahuje vektor $\mathbf{u}$.*

Důkaz. První kódové slovo $\mathbf{c}_1$ lze převést na $\mathbf{u}$ pomocí nejvýše n symbolových permutací. ∎

Definice. Buď $\mathbf{x}, \mathbf{y}$ binární slova délky n. *Průnik* $\mathbf{x} \cap \mathbf{y}$ *binárních slov* $\mathbf{x}$ *a* $\mathbf{y}$ je binární řetězec délky n, který má jedničku přesně na těch místech, na kterých ji mají obě slova $\mathbf{x}$ a $\mathbf{y}$. Všude jinde má pak nuly. Jinak řečeno, $\mathbf{x} \cap \mathbf{y} = x_1 \cdot y_1 x_2 \cdot y_2 \dots x_n \cdot y_n$.

Platí pak následující jednoduché lemma.

Lemma 2.4 *Jsou-li $\mathbf{x}$ a $\mathbf{y}$ binární řetězce délky n, pak*

$$d(\mathbf{x}, \mathbf{y}) = \mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathbf{y}) - 2\mathrm{w}(\mathbf{x} \cap \mathbf{y}).$$

Důkaz. Položme $A_{11} = \{i : 1 \le i \le n, x_i = y_i = 1\}$, $a_{11} = \mathrm{card}(A_{11})$, $A_{10} = \{i : 1 \le i \le n, x_i = 1, y_i = 0\}$, $a_{10} = \mathrm{card}(A_{10})$, $A_{01} = \{i : 1 \le i \le n, x_i = 0, y_i = 1\}$, $a_{01} = \mathrm{card}(A_{01})$, $A_{00} = \{i : 1 \le i \le n, x_i = 0, y_i = 0\}$, $a_{00} = \mathrm{card}(A_{00})$. Pak platí

$$\begin{aligned} \mathrm{d}(\mathbf{x}, \mathbf{y}) = a_{10} + a_{01} &= (a_{11} + a_{10}) + (a_{11} + a_{01}) - 2a_{11} \\ &= \mathrm{w}(\mathbf{x}) + \mathrm{w}(\mathbf{y}) - 2\mathrm{w}(\mathbf{x} \cap \mathbf{y}), \end{aligned}$$

čímž je lemma dokázáno. ∎

Definice. Postup, při kterém přidáme ke všem kódovým slovům z daného kódu jednu nebo více dodatečných pozic, a tedy zvýšíme délku kódu, se nazývá *rozšíření kódu.*

Nejznámější metoda rozšíření kódu se nazývá *kontrola parity.* Pro jednoduchost uvažme binární případ.

Je-li $\mathcal{C}$ binární (n, M, d)-kód, budeme konstruovat nový kód následovně. Ke každému kódovému slovu $\mathbf{c} = c_1 c_2 \dots c_n \in \mathcal{C}$ přidáme dodatečný bit tak, že nové výsledné kódové slovo bude mít sudou váhu. Tedy, mělo-li $\mathbf{c}$ lichou váhu, přidáme jedničku, mělo-li $\mathbf{c}$ sudou váhu, přidáme nulu. Označíme-li tedy výsledné slovo jako $\overline{\mathbf{c}}$, máme

$$\overline{\mathbf{c}} = \begin{cases} c_1 c_2 \dots c_n 0 & \text{pokud } \mathrm{w}(c) \text{ je sudá,} \\ c_1 c_2 \dots c_n 1 & \text{pokud } \mathrm{w}(c) \text{ je lichá.} \end{cases}$$

Nový kód $\overline{\mathcal{C}}$ má pak délku $n+1$ a velikost M. Minimální vzdálenost kódu $\overline{\mathcal{C}}$ bude buď d nebo $d+1$ a toto číslo bude záviset na tom, zda bude d sudé nebo liché číslo. Totiž, protože všechna kódová slova v $\overline{\mathcal{C}}$ mají sudou váhu, bude vzdálenost mezi každými dvěma slovy sudé číslo (to plyne z lemmatu 2.4). Je tedy i minimální vzdálenost kódu $\overline{\mathcal{C}}$ sudé číslo. Nutně pak dostáváme, že, je-li $\mathrm{d}(\mathcal{C}) = d$ sudé číslo a nastává pro pro slova $\mathbf{c}, \mathbf{d}$, pak nutně mají obě slova stejnou paritu a tedy nutně platí $\mathrm{d}(\mathbf{c}, \mathbf{d}) = \mathrm{d}(\overline{\mathbf{c}}, \overline{\mathbf{d}})$. Je tedy $\mathrm{d}(\mathcal{C}) = \mathrm{d}(\overline{\mathcal{C}})$. Nechť je minimální vzdálenost kódu $\mathcal{C}$ liché číslo a nastává pro pro slova $\mathbf{c}, \mathbf{d}$, pak nutně má jedno ze slov sudou paritu (například $\mathbf{c}$) a druhé lichou paritu ($\mathbf{d}$). Pak $\mathrm{w}(\mathbf{c}) = \mathrm{w}(\overline{\mathbf{c}})$, $\mathrm{w}(\mathbf{d}) + 1 = \mathrm{w}(\overline{\mathbf{d}})$, $\mathrm{w}(\mathbf{c} \cap \mathbf{d}) = \mathrm{w}(\overline{\mathbf{c}} \cap \overline{\mathbf{d}})$. Tedy $\mathrm{d}(\mathcal{C}) + 1 = \mathrm{d}(\overline{\mathcal{C}})$.

V obou případech pak máme

$$\lfloor \frac{1}{2}(\mathrm{d}(\mathcal{C}) - 1) \rfloor = \lfloor \frac{1}{2}(\mathrm{d}(\overline{\mathcal{C}}) - 1) \rfloor.$$

Z toho pak plyne, že se nám při použití kontroly parity nezvýší schopnost opravit chybu.

Obecně pak, máme-li kódovou abecedu vybránu tak, že nám tvoří konečné pole, řekněme $\mathbf{Z}_p$, kde p je prvočíslo, můžeme definovat *kontrolu parity* jako

$$\overline{\mathbf{c}} = c_1 c_2 \dots c_n c_{n+1}, \text{kde } c_{n+1} = -\sum_{i=1}^{n} c_n.$$

Definice. Postup, při kterém odebereme ta kódová slova z daného kódu, která se liší na určené pozici i od určeného symbol s, a ze zbývajících slov tuto pozici odstraníme, a tedy zkrátíme délku kódu, se nazývá *zkrácení kódu typu $x_i = s$.*

Je-li pak $\mathcal{C}$ (n, M, d)-kód, má zkrácený kód $\mathcal{C}^{\odot}$ délku $n-1$ a minimální vzdálenost alespoň d. Opravdu, zkrácení kódu může mít za důsledek podstatné zvětšení minimální vzdálenosti tedy i schopnosti opravit nového kódu, protože můžeme odstranit ta kódová slova, která se „špatně chovají vzhledem ke vzdálenosti“. Samozřejmě ale zkrácením kódu se nám zmenší i počet kódových slov, což není zrovna lákavé.

Věta 2.5 *Buď d liché přirozené číslo. Pak existuje binární (n, M, d)-kód právě tehdy, když existuje binární $(n+1, M, d+1)$-kód.*

Důkaz. Pokud existuje binární $(n+1, M, d+1)$-kód $\mathcal{C}$, můžeme snadno zkonstruovat binární (n, M, d)-kód. Jednoduše vybereme dvě kódová slova $\mathbf{c}$ a $\mathbf{d}$ tak, že $\mathrm{d}(\mathbf{c}, \mathbf{d}) = d+1$, najděme pozici, na které se liší a odebereme tuto pozici z každého kódového slova. Nový kód označíme $\mathcal{C}'$. Nutně pak mají nová zkrácená slova $\mathbf{c}'$ a $\mathbf{d}'$ vzdálenost $\mathrm{d}(\mathbf{c}', \mathbf{d}') = d$ a žádná jiná dvě slova nemají od sebe menší vzdálenost než d. Celkem je tedy $\mathcal{C}'$ binární (n, M, d)-kód.

Obráceně, předpokládejme, že máme binární (n, M, d)-kód $\mathcal{D}$ (d liché). Kód $\overline{\mathcal{D}}$, který vznikl jako kód kontroly parity z $\mathcal{D}$, má délku $n+1$, velikost M a minimální vzdálenost $d+1$.

3 Hlavní problém teorie kódování

Definice. Buď dána přirozená čísla d, n, q. Položme $A_q(n, d)$ jakožto maximální M takové, že existuje q-ární (n, M, d)-kód. Každý takový q-ární (n, M, d)-kód nazýváme *optimální.*

Čísla $A_q(n, d)$ hrají ústřední roli v teorii kódování a na jejich nalezení bylo vynaloženo velké úsilí. Často se mluví o *hlavním problému teorie kódování.* V dalším pro ilustraci určíme jisté hodnoty $A_q(n, d)$ pro malé hodnoty n a d a dokážeme obecná tvrzení o těchto číslech.

Poznamenejme, že abychom dokázali, že $A_q(n, d) = K$ pro jisté přirozené číslo K, stačí ověřit, že $A_q(n, d) \leq K$ a následně najít vhodný q-ární (n, K, d')-kód $\mathcal{C}$, kde $d \leq d'$. Pak totiž $K \leq A_q(n, d') \leq A_q(n, d)$.

Věta 3.1 *Je-li d sudé číslo, je $A_2(n, d) = A_2(n-1, d-1)$.*

Důkaz. Plyne okamžitě z věty 2.5. Totiž pak nutně platí $A_2(n, d) \leq A_2(n-1, d-1)$ a $A_2(n, d) \geq A_2(n-1, d-1)$. ∎

Důsledek 3.2 *Je-li d sudé číslo a existuje binární (n, M, d)-kód, pak existuje binární (n, M, d)-kód, ve kterém mají všechna kódová slova sudou váhu.*

Důkaz. Plyne okamžitě z věty 3.1. Totiž pak nutně existuje binární kód $(n-1, M, d-1)$-kód a pomocí operace kontroly parity existuje binární (n, M, d)-kód, ve kterém mají všechna kódová slova sudou váhu. ∎

Následující dva snadné výsledky nám budou ilustrovat, jakým způsobem můžeme určit hodnoty $A_2(n, d)$ pro malé hodnoty n a d. Použijeme přitom lemma 2.3, ze kterého plyne, že pro daný (n, M, d)-kód $\mathcal{C}$ existuje ekvivalentní (n, M, d)-kód $\mathcal{C}'$, který obsahuje nulové slovo (samozřejmě za předpokladu, že kódová abeceda obsahuje 0 – jinak ji lze dodat záměnou za jiný symbol). Můžeme tedy v dalším předpokládat, že naše kódy obsahují nulové slovo.

Věta 3.3 *Platí $A_2(4, 3) = 2$.*

Důkaz. Buď $\mathcal{C}$ nějaký $(4, M, 3)$-kód. Můžeme bez újmy na obecnosti předpokládat, že $\mathbf{0} = 0000 \in \mathcal{C}$. Protože $\mathrm{d}(\mathcal{C}) = 3$, libovolné další kódové slovo $\mathbf{c}$ z $\mathcal{C}$ musí splňovat $\mathrm{d}(\mathbf{c}, \mathbf{0}) \geq 3$ a tedy musí obsahovat alespoň tři jedničky. Máme tedy celkem pět možností pro nenulová slova ležící v $\mathcal{C}$, a to

$$1110, 1101, 1011, 0111, 1111.$$

Ale každá takováto dvě slova mají vzdálenost nejvýše 2 a tedy pouze jedno z nich může být obsaženo v $\mathcal{C}$. Platí tedy $A_2(4, 3) \leq 2$. Dále platí, protože $\mathcal{C} = \{0000, 1110\}$ je $(4, 2, 3)$-kód, máme $A_2(4, 3) \geq 2$ a tedy celkem $A_2(4, 3) = 2$. ∎

Věta 3.4 *Platí $A_2(5, 3) = 4$.*

Důkaz. Buď $\mathcal{C}$ nějaký $(5, M, 3)$-kód. Můžeme bez újmy na obecnosti předpokládat, že $\mathbf{0} = 0000 \in \mathcal{C}$ a přitom pro vhodné $\mathbf{c}$ z $\mathcal{C}$ platí $\mathrm{d}(\mathbf{0}, \mathbf{c}) = 3$, $c_1 = 0$. Uvažme nyní zkrácení $\mathcal{C}^{\odot}$ typu $x_1 = 0$. Víme pak, že $\mathbf{0}^{\odot}, \mathbf{c}^{\odot} \in \mathcal{C}^{\odot}$ a $\mathrm{d}(\mathbf{0}^{\odot}, \mathbf{c}^{\odot}) = 3$. Dále víme, že $A_2(4, 3) = 2$ a $A_2(4, 4) = A_2(3, 3) = 2$. Tedy i $\mathrm{card}(\mathcal{C}^{\odot}) = 2$. Definuje nyní zkrácený kód $\mathcal{C}^{\ominus}$ jakožto zkrácení typu typu $x_1 = 1$. Pak buďto $\mathrm{card}(\mathcal{C}^{\ominus}) = 1$ nebo $\mathrm{card}(\mathcal{C}^{\ominus}) > 1$ a $\mathrm{d}(\mathcal{C}^{\ominus}) = 3$ a tedy nutně jako výše $\mathrm{card}(\mathcal{C}^{\ominus}) = 2$. Celkem tedy $\mathrm{card}(\mathcal{C}^{\ominus}) + \mathrm{card}(\mathcal{C}^{\odot}) = \mathrm{card}(\mathcal{C}) \leq 4$. Platí tedy $A_2(5, 3) \leq 4$. Dále platí, protože $\mathcal{C} = \{00000, 11100, 00111, 11011\}$ je $(5, 4, 3)$-kód, máme $A_2(5, 3) \geq 4$ a tedy celkem $A_2(5, 3) = 4$. ∎

Věta 3.5 *Platí následující:*

1. $A_q(n, d) \leq q^n$ *pro všechna* $1 \leq d \leq n$*;*
2. $A_q(n, 1) = q^n$*;*
3. $A_q(n, n) = q$.

Důkaz. První tvrzení platí, protože pro každý kód $\mathcal{C}$ je $\mathcal{C} \subseteq V_n(q)$ tj. $\mathrm{card}(\mathcal{C}) \leq q^n$. Druhé tvrzení plyne z toho, že uvážíme-li $\mathcal{C} = V_n(q)$, máme $\mathrm{d}(V_n(q)) = 1$. Třetí část plyne z toho, že se kódová slova musí lišit na všech pozicích a takových kódových slov můžeme vybrat nejvýše q. Ale máme, pro kód $\mathcal{D} = \{\mathbf{0}, \ldots, \mathbf{q\text{-}1}\}$, že $\mathcal{D}$ je (n, q, n)-kód. ∎

n	$d=3$	$d=5$	d=7
5	4	2	-
6	8	2	-
7	16	2	2
8	20	4	2
9	40	6	2
10	72-79	12	2
11	144-158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2560-3276	256-340	36-37

Tabulka 4.1: Hodnoty $A_2(n,d)$

Už pro malé hodnoty q, n a d není velikost $A_q(n,d)$ známa. Následující tabulka shrnuje většinu našich znalostí o $A_2(n,d)$.

Poznamenejme, že problém určení $A_2(n,d)$ je problémem konečných geometrií.

Pro odhad $A_q(n,d)$ platí následující jednoduché tvrzení.

Věta 3.6 *Pro všechna $n \geq 2$,*

$$A_q(n,d) \leq qA_r(n-1,d). \tag{4.1}$$

Důkaz. Buď $\mathcal{C}$ kód realizující hodnotu $A_q(n,d)$. Uvažme nyní zkrácení $\mathcal{C}_j$ typu $x_n = j$. Pak nutně $\mathrm{card}(\mathcal{C}_j) \leq A_q(n-1,d)$ (mohou totiž nastat pouze dva případy: $\mathrm{card}(\mathcal{C}_j) = 1$, což evidentně platí, a $K = \mathrm{card}(\mathcal{C}_j) > 1$, kde pak $\mathcal{C}_j$ je $(n-1,K,d')$-kód, $d' \geq d$ a tedy tvrzení rovněž platí). Celkem pak $\mathcal{C} = \bigcup_{j=0}^{q-1} \mathcal{C}_j$ tj. $A_q(n,d) = \sum_{j=0}^{q-1} \mathrm{card}(\mathcal{C}_j) \leq qA_r(n-1,d)$. ▮

Cvičení 3.7 *1. Ukažte, že $A_2(3,2) = 4$.*

4 Dolní a horní hranice $A_q(n,d)$; perfektní kódy

Určeme nejprve horní hranici (sphere-packing upper bound) čísla $A_q(n,d)$.

Lemma 4.1 *Nechť $e = \lfloor \frac{1}{2}(d-1) \rfloor$. Pak platí*

$$A_q(n,d) \sum_{k=0}^{e} \binom{n}{k} (q-1)^k \leq q^n. \tag{4.2}$$

Důkaz. Buď $\mathcal{C}$ kód s minimální vzdáleností d; pak, je-li $\mathbf{S}_e(\mathbf{x})$ koule o poloměru e se středem $\mathbf{x}$, máme pro každou dvojici kódových slov $\mathbf{x}$ a $\mathbf{y}$, že

$$\mathbf{S}_e(\mathbf{x}) \cap \mathbf{S}_e(\mathbf{y}) = \emptyset.$$

Ale je evidentní, že

$$|\mathbf{S}_e(\mathbf{x})| = \sum_{k=0}^{e} \binom{n}{k} (q-1)^k. \tag{4.3}$$

Pravá strana nerovnosti 4.2 je celkový počet slov délky n nad abecedou o q symbolech. Levá strana je počet prvků obsažených v disjunktním sjednocení koulí, jejichž středy jsou navzájem různá kódová slova. Maximální počet takovýchto různých kódových slov je $A_q(n,d)$. Tedy dostáváme nerovnost 4.2. ▮

Podobně platí

Lemma 4.2 (**Gilbert-Varshamova hranice**)

$$A_q(n,d) \sum_{k=0}^{d-1} \binom{n}{k} (q-1)^k \geq q^n. \tag{4.4}$$

Důkaz. Buď $\mathcal{C}$ (n,M,d)-kód s maximálním počtem kódových slov. Pak zcela jistě neexistuje vektor z $V_n(q) - \mathcal{C}$, jehož vzdálenost od všech kódových slov je alespoň d, jinak by totiž M nebyl maximální počet kódových slov. Jinak řečeno, koule o poloměru d musí pokrývat celý prostor $V_n(q)$. Ale to je přesně podmínka 4.4. ▮

Definice. *Poloměr pokrytí* blokového kódu $\mathbf{C}$ je nejmenší poloměr ρ takový,

$$\mathbf{F}_q^n \subseteq \bigcup_{\mathbf{c} \in \mathbf{C}} S_\rho(\mathbf{c}).$$

Poloměr pokrytí je další charakterizací kódů, nemá však tak hojné uplatnění jako minimální vzdálenost.

Věta 4.3 *Nechť $\mathcal{C}$ je blokový kód délky n. Pak ρ je poloměr pokrytí kódu $\mathcal{C}$ právě tehdy, když $\rho = \max_{\mathbf{f} \in \mathbf{F}_q^n} \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{c},\mathbf{f})$.*

Důkaz: Nechť $\mathbf{F}_q^n \subseteq \bigcup_{\mathbf{c} \in \mathcal{C}} S_\rho(\mathbf{c})$, kde ρ je minimální. Pak pro každé $\mathbf{f} \in \mathbf{F}_q^n$ existuje $\mathbf{c} \in \mathcal{C}$ takové, že $d(\mathbf{c},\mathbf{f}) \leq \rho$, a současně existují $\mathbf{f}' \in \mathbf{F}_q^n$ a $\mathbf{c}' \in \mathbf{C}$ splňující $d(\mathbf{c}',\mathbf{f}') = \rho$. Z minimality ρ plyne, že $\rho = \max_{\mathbf{f} \in \mathbf{F}_q^n} d(\mathbf{f},\mathcal{C}) = \max_{\mathbf{f} \in \mathbf{F}_q^n} \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{c},\mathbf{f})$.

Naopak, nechť $\rho = \max_{\mathbf{f} \in \mathbf{F}_q^n} \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{c},\mathbf{f}) = \max_{\mathbf{f} \in \mathbf{F}_q^n} d(\mathbf{f},\mathcal{C})$. Pak pro všechna $\mathbf{f} \in \mathbf{F}_q^n$ platí $\rho \geq d(\mathbf{f},\mathcal{C})$ a existují $\mathbf{f}' \in \mathbf{F}_q^n$ a $\mathbf{c}' \in \mathcal{C}$ splňující rovnost a pro všechna $\mathbf{c} \in \mathcal{C}$ je $\rho \leq d(\mathbf{c},\mathbf{f}')$. Předpokládejme, že existuje s, $\rho > s$, takové, že každé $\mathbf{f} \in \mathbf{F}_q^n$ je prvkem množiny $\{x \in \mathbf{F}_q^n \mid d(\mathbf{c},x) \leq s\}$ pro nějaké $\mathbf{c} \in \mathcal{C}$. To je ale spor s existencí slov $\mathbf{f}'$ a $\mathbf{c}'$, a tedy ρ je poloměr pokrytí kódu $\mathcal{C}$. ▮

Ideální situace z ekonomického pohledu je najít kód $\mathcal{C}$ nad $V_n(q)$ tak, že pro jisté kladné $t > 0$ jsou všechny prvky z $V_n(q)$ obsaženy v disjunktním sjednocení koulí, jejichž středy jsou navzájem různá kódová slova. Takový kód se pak nazývá *perfektní*. Z jeho definice je zřejmé, že perfektní kód dokáže pomocí pravidla minimální vzdálenosti opravit až t chyb, a nedokáže opravit $t+1$ chyb.

Je tedy nutná podmínka pro to, aby (n,M,d)-kód byl perfektní, že d je liché číslo. Celkem tedy je (n,M,d)-kód perfektní právě tehdy, když $M = A_q(n,d)$ a

$$A_q(n,d)\sum_{k=0}^{\frac{d-1}{2}}\binom{n}{k}(q-1)^k = q^n. \tag{4.5}$$

Příklad 4.4 Zřejmým příkladem perfektního kódu je

1. každý kód s právě jedním kódovým slovem,
2. každý binární kód s právě dvěma slovy lichých délek, např. $00\ldots0$ a $11\ldots1$.

Tyto kódy se nazývají *triviální perfektní kódy*.

Věta 4.5 (**Singletonova hranice**) *Platí*

$$A_q(n,d) \le q^{n-d+1}. \tag{4.6}$$

Důkaz. Buď $\mathcal{C}$ nějaký (n,M,d)-kód. Pokud odstraníme posledních $d-1$ pozic z každého kódového slova z $\mathcal{C}$, musí být nutně výsledná zkrácená slova navzájem různá (jinak by původní slova musela mít vzdálenost $\le d-1$). Ale počet všech slov délky $n-(d-1)$ je právě q^{n-d+1} tj. $A_q(n,d)\le q^{n-d+1}$. ∎

Lemma 4.6 *Buď M přirozené číslo. Pak funkce $f:\{0,\ldots,M\}\to\mathbf{N}$ definovaná jako $f(k)=k(M-k)$ nabývá svého maxima pro*

$$\overline{k} = \begin{cases} \frac{M}{2} & \text{pokud } M \text{ je sudé} \\ \frac{M\pm1}{2} & \text{pokud } M \text{ je liché} \end{cases} \quad \text{a} \quad f(\overline{k}) = \begin{cases} \frac{M^2}{4} & \text{pokud } M \text{ je sudé} \\ \frac{M^2-1}{4} & \text{pokud } M \text{ je liché.} \end{cases}$$

Důkaz. Důkaz okamžitě plyne ze vztahu $\sqrt{a\cdot b}\le\frac{1}{2}(a+b)$ a z průběhu funkce f. ∎

Věta 4.7 (**Plotkinova hranice**) *Je-li $n<2d$, máme*

$$A_2(n,d) \le 2\lfloor\frac{d}{2d-n}\rfloor. \tag{4.7}$$

Důkaz. Buď $\mathcal{C}=\{\mathbf{c}_1,\ldots,\mathbf{c}_M\}$ nějaký (n,M,d)-kód. Uvažme součet $S=\sum_{i<j}\mathrm{d}(\mathbf{c}_i,\mathbf{c}_j)$. To není nic jiného, než součet všech vzdáleností kódových slov z $\mathcal{C}$. Protože ale $d\le\mathrm{d}(\mathbf{c}_i,\mathbf{c}_j)$ pro všechna i,j a máme právě $\binom{M}{2}$ dvojic kódových slov z $\mathcal{C}$, platí

$$S=\sum_{i<j}\mathrm{d}(\mathbf{c}_i,\mathbf{c}_j)\ge d\binom{M}{2}. \tag{4.8}$$

Pokusme se nyní spočítat S tím, že se podíváme na každou pozici zvlášť. Uvažme tedy kódová slova ve tvaru

$$\begin{aligned}\mathbf{c}_1 &= c_{11}\ c_{12}\ \ldots c_{1n}\\ \mathbf{c}_2 &= c_{21}\ c_{22}\ \ldots c_{2n}\\ &\ \ \vdots\\ \mathbf{c}_M &= c_{M1}c_{M2}\ldots c_{Mn}.\end{aligned}$$

Máme pak, pro všechna $j, 1\le j\le n$, že pokud k_j bitů slova $c_{1j}\ldots c_{Mj}$ je rovno 1 a zbývajících $M-k_j$ bitů je nulových, pak tyto bity přispějí k součtu všech vzdáleností číslem $f(k_j)=k_j(M-k_j)$. Máme tedy celkem (protože všech sloupců je n) $S\le nf(\overline{k})$. Zejména tedy platí

$$S\le nf(\overline{k})=\begin{cases} n\frac{M^2}{4} & \text{pokud M je sudé}\\ n\frac{M^2-1}{4} & \text{pokud M je liché.}\end{cases}$$

Dáme-li obě nerovnosti dohromady, máme

$$\binom{M}{2}\cdot d\le\begin{cases} n\frac{M^2}{4} & \text{pokud M je sudé}\\ n\frac{M^2-1}{4} & \text{pokud M je liché.}\end{cases}$$

Po jednoduché úpravě pak obdržíme

$$M\le\begin{cases}\frac{2d}{2d-n} & \text{pokud M je sudé}\\ \frac{n}{2d-n}<\frac{2d}{2d-n} & \text{pokud M je liché.}\end{cases}$$

položme $a=\frac{2d}{2d-n}$. Pak máme

$$M\le\begin{cases}\lfloor 2a\rfloor & \text{pokud M je sudé}\\ \lfloor 2a\rfloor-1 & \text{pokud M je liché.}\end{cases}$$

Předpokládejme nejprve, že $k\le a<k+\frac{1}{2}$ pro nějaké přirozené číslo k. Pak $\lfloor 2a\rfloor = 2k$ a $2\lfloor a\rfloor=\lfloor 2a\rfloor$. Máme tedy, nezávisle na paritě M, že $M\le 2\lfloor a\rfloor$. Předpokládejme nyní, že $k+\frac{1}{2}\le a<k+1$ pro nějaké přirozené číslo k. Pak $\lfloor 2a\rfloor=2k+1$ a $2\lfloor a\rfloor=2k$. Je-li M liché, máme $M\le\lfloor 2a\rfloor-1=2k=2\lfloor a\rfloor$ a, je-li M sudé, máme $M\le\lfloor 2a\rfloor-1=2k+1$ tj. $=M\le 2k=2\lfloor a\rfloor$.

Nutně tedy celkem $M\le 2\lfloor\frac{d}{2d-n}\rfloor$. ∎

Lemma 4.8 *Buď k přirozené číslo. Pak $A_2(4k-1, 2k-1) \leq 8k$ a $A_2(4k, 2k) \leq 8k$.*

Důkaz. Ověřme nejprve, že $A_2(4k, 2k) \leq 8k$. Víme ale, že $A_2(4k, 2k) \leq 2A_2(4k-1, 2k) \leq 4\lfloor\frac{2k}{1}\rfloor = 8k$ dle 4.7. Protože dále platí $A_2(4k-1, 2k-1) = A_2(4k, 2k) \leq 8k$, tvrzení je dokázáno. ▮

Cvičení 4.9 *1. Ukažte, že $19 \leq A_2(10, 3) \leq 93$.*

2. Ukažte, že pro všechna přirozená čísla q, parametry $n = (q^r - 1)/(q-1)$, $M = q^{n-r}$, $d = 3$, kde r je nějaké přirozené číslo ≥ 2, splňují podmínku 4.5 proto, aby se jednalo o perfektní kód. Poznamenejme, že ačkoliv tyto parametry splňují 4.5 pro každé přirozené číslo q, byla vyslovena hypotéza, že příslušné perfektní kódy existují právě tehdy, když je q mocnina prvočísla.

5 Lineární kódy

Předpokládejme, že $\mathcal{C}$ je kód s minimální vzdáleností $d = 2e + 1$ a lze tedy pomocí metody nejbližšího kódového slova opravit až e chyb. Má-li kód $\mathcal{C}$ málo prvků, jedná se o velmi praktickou metodu. V případě, že číslo $|\mathcal{C}|$ bude velké, bude tato metoda opravdu velmi časově náročná, protože musíme srovnávat přijatý vektor $\mathbf{y}$ s velkým množstvím kódových slov. To je důvod pro studium více strukturovaných kódů, jako jsou např. lineární kódy.

Předpokládejme tedy, že počet prvků q naší abecedy je prvočíselná mocnina p^m. Můžeme tedy považovat Σ za těleso F_q o q-prvcích.

Buď dále $V_n(q)$ vektorový prostor dimenze n nad tělesem F_q. Typický prvek tohoto vektorového prostoru budeme psát jakožto $\mathbf{x} = (x_1, \ldots, x_n)$, občas pak zkráceně jakožto $\mathbf{x} = x_1 \ldots x_n$, kde $x_i \in F_q$.

Lineární kód $\mathcal{C}$ nad Σ je definován jakožto podprostor prostoru $V_n(q)$. Má-li tento podprostor dimenzi k, mluvíme o $[n, k]$-kódu nebo, chceme-li specifikovat minimální vzdálenost, mluvíme o $[n, k, d]$-kódu. Protože každý k-dimenzionální podprostor nad F_q má q^k prvků, máme:

Každý $[n, k, d]$-kód nad F_q je (n, q^k, d)-kód.

Výhoda lineárních kódů je to, že pomocí k kódových slov délky n můžeme zcela popsat kód s právě q^k kódovými slovy délky n. Tím dosáhneme obrovské úspory paměti. Totiž každý podprostor dimenze k je úplně popsán k lineárně nezávislými vektory.

Definujeme pak *generující matici* pro lineární $[n, k]$-kód $\mathcal{C}$ jakožto libovolnou matici rozměru $k \times n$, jejíž řádky tvoří k lineárně nezávislých kódových slov z $\mathcal{C}$. Předpokládejme nyní, že G je generující matice kódu $\mathcal{C}$ a G' je matice, kterou můžeme obdržet z G pomocí konečné posloupnosti permutací následujícího typu:

1. záměna řádků,
2. násobení řádku nenulovým skalárem,
3. přičtení k řádku skalární násobek jiného řádku,
4. záměna sloupců,
5. násobení sloupce nenulovým skalárem.

Pak lze snadno ukázat následující tvrzení.

Lemma 5.1 *G' je generující matice kódu $\mathcal{C}'$, který je ekvivalentní s kódem $\mathcal{C}$.*

Důkaz. Stačí ukázat, že každá z operací 1-5 odpovídá vytvoření generující matice G' kódu $\mathcal{C}'$, který je ekvivalentní s kódem $\mathcal{C}$. Evidentně, záměna řádků, vynásobení řádku nenulovým skalárem a přičtení řádků jsou operace takové, že dokonce kód $\mathcal{C}'$ je totožný s kódem $\mathcal{C}$. Záměna sloupců v matici G znamená, že provedeme poziční permutaci určenou transpozicí sloupců. Vynásobení sloupce nenulovým skalárem je symbolová permutace tohoto sloupce (pracujeme nad tělesem a pak je množina nenulových skalárů grupou). Jsou tedy odpovídající kódy ekvivalentní s kódem $\mathcal{C}$. ▮

Lemma 5.2 *Buď G matice typu $k \times n$ jejíž řádky jsou lineárně nezávislé; pak, aplikujeme-li posloupnost operací typu (1)-(5) na G, je možné G převést na matici $G' = [E_k, A]$, kde E_k je jednotková matice typu $k \times k$.*

Důkaz. Důkaz je veden indukcí vzhledem ke k. Pokud $k = 1$, je tvrzení evidentní. Stačí vynásobit řádek matice G prvkem inverzním k prvku g_{11}. Předpokládejme, že tvrzení platí pro k a chceme jej dokázat pro $k+1$. Protože hodnost matice G je $k+1$ existuje v matici G $k+1$ nezávislých sloupců. Pomocí operace typu (4) tyto sloupce dostaneme na prvních $k+1$ sloupců nové matice G'. Pak nutně v $k+1$-ním sloupci existuje nenulový prvek – jinak by hodnost matice nebyla $k+1$. Provedeme pak pomocí operace typu (1) záměnu příslušného řádku s posledním řádkem. Nový poslední řádek vynásobíme po řadě vhodnými skaláry a odečteme jej od předchozích řádků tak, aby se nám $k+1$-ní sloupec až na poslední řádek vynuloval. Pak obdržíme matici G'', jejichž prvních k řádků (po vynechání $k+1$-ního sloupce) je matice typu $k \times (n-1)$, která má hodnost k. Lze pak aplikovat indukční předpoklad a obdržíme matici, která má na prvních k řádcích submatici typu $[E_k, A]$ (přitom na vynechaný $k+1$-ní sloupec budeme provádět pouze řádkové úpravy). Snadno se zbavíme nenulových prvků v posledním řádků na prvních k-místech pomocí operace typu (3). Přitom nutně na místě $(k+1, k+1)$ je nenulový prvek, stačí pak vynásobit prvkem k němu inverzním. ▮

V důsledku 5.2 můžeme bez újmy na obecnosti pracovat s generujícími maticemi ve výše uvedené standardní formě. Jinou užitečnou vlastností lineárních kódů je, že jejich minimální vzdálenost lze najít mnohem snáze, než v případu obecných kódů. Máme pak následující výsledek

Věta 5.3 *Minimální vzdálenost d lineárního kódu $\mathcal{C}$ je minimální váha w všech nenulových vektorů z $\mathcal{C}$.*

Důkaz. Buď d minimální vzdálenost lineárního kódu $\mathcal{C}$ a předpokládejme, že $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ tak, že $\mathrm{d}(\mathbf{x}, \mathbf{y}) = d$. Pak $\mathbf{x} - \mathbf{y} \in \mathcal{C}$, $\mathrm{w}(\mathbf{x} - \mathbf{y}) = d \geq w = \mathrm{w}(\mathbf{z}) = \mathrm{d}(\mathbf{z}, \mathbf{0}) \geq d$ pro vhodný vektor $\mathbf{z} \in \mathcal{C}$. ▮

6 Použití lineárních kódů

Předpokládejme, že $\mathcal{C}$ je lineární $[n, k]$-kód nad $F_q = \Sigma$ a že má generující matice G tvaru

$$G = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_k \end{bmatrix} = [E_k, A],$$

kde $\mathbf{r}_i$ jsou vektory délky n nad F_q a A je matice typu $k \times (n-k)$. Kódová slova kódu $\mathcal{C}$ jsou vektory délky n tvaru

$$\sum_{i=1}^{k} a_i \mathbf{r}_i, \quad a_i \in F_q.$$

Základní myšlenka zakódování je následující. Pokud je zpráva braná jako posloupnost $\mathbf{s} = (s_1, \ldots, s_k)$, zakódujeme $\mathbf{s}$ pomocí kódového slova $\mathbf{c} = (c_1, \ldots, c_n)$, kde c_i jsou určena předpisem

$$[c_1, \ldots, c_n] = [s_1, \ldots, s_k]\,[E_k, A], \tag{4.9}$$

tj. $c_i = s_i$ pro $1 \leq i \leq k$.

Příklad 6.1 Předpokládejme, že kód $\mathcal{C}$ nad tělesem F_3 (což je těleso zbytkových tříd po dělení 3) má generující matici G tvaru

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 0 & 1 \end{bmatrix}.$$

Je-li vstupní zpráva ze zdroje tvaru

$$102101210122\ldots$$

rozdělíme ji nejprve do bloků délky tři a obdržíme

$$102 \mid 101 \mid 210 \mid 122 \mid \ldots$$

a pak zakódujeme zdrojová slova jakožto

$$\begin{array}{ll} 102 \mapsto \mathbf{r}_1 + 2\mathbf{r}_3 = 102222, & 101 \mapsto \mathbf{r}_1 + \mathbf{r}_3 = 101021 \\ 201 \mapsto 2\mathbf{r}_1 + \mathbf{r}_3 = 210221, & 122 \mapsto \mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 = 122211. \end{array}$$

Dostaneme tedy posloupnost

$$102 \mid 222 \mid 101 \mid 021 \mid 210 \mid 221 \mid 122 \mid 211 \mid \ldots$$

Tedy jsme, při zdvojení délky zprávy, zpomalili rychlost přenosu na polovic. Zvýšili jsme ale spolehlivost.

Poznamenejme, že vztah 4.9 je ekvivalentní s rovnicí

$$\left[-A^\top, E_{n-k}\right][c_1, \ldots, c_n]^\top = \mathbf{0}. \tag{4.10}$$

Matice $H = \left[-A^\top, E_{n-k}\right]$ se nazývá matice *kontroly parity* kódu $\mathcal{C}$. Zejména tedy platí, že $\mathbf{z} \in \mathcal{C}$ právě tehdy, když $H\,[z_1, \ldots, z_n]^\top = \mathbf{0}$.

Přitom matice H kontroly parity definuje jak vlastní kód $\mathcal{C}$ tak i příslušnou generující matici G. Název matice kontroly parity znamená, že na jistých kontrolních místech jsou přidány jisté kontrolní součty, které zkontrolují naše kódová slova. Občas budeme pro $[n, k]$-kód říkat, že prvních k složek kódového slova je nazýváno *informačními znaky* a zbývajících $n - k$ složek jsou *symboly kontroly parity* (*kontrolní znaky*).

Příklad 6.2 Určeme nyní matici H kontroly parity z příkladu 6.1. Ta má tvar

$$H = \begin{bmatrix} -1 & 0 & -2 & 1 & 0 & 0 \\ -2 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 0 & 0 & 1 \end{bmatrix}.$$

Kódové slovo 102222 sestává z ze zprávových čísel 102 a symbolů kontroly parity 222. Evidentně, rovnost 4.10 je pro toto kódové slovo (a všechna zbývající) splněna.

Obecně musí tedy kódová slova splnit systém rovnic

$$2c_1 + c_3 + c_4 = 0 \quad c_1 + 2c_2 + c_5 = 0 \quad 2c_2 + 2c_3 + c_6 = 0. \tag{4.11}$$

Základní idea pro metodu opravování chyb je vidět na tomto příkladě. Předpokládejme, že naše obdržené slovo nesplňuje první a třetí rovnici 4.11 v rovnicích kontroly parity. Pak můžeme dedukovat, že chybná číslice zprávy je číslice c_3, protože to je jediná číslice, která se vyskytuje v obou rovnicích. ▮

Věta 6.3 *Je-li H matice kontroly parity kódu $\mathcal{C}$ délky n, pak kód $\mathcal{C}$ má minimální vzdálenost d tehdy a jen tehdy, když každých $d-1$ sloupců matice H je nezávislých, ale některých d sloupců je lineárně závislých.*

Důkaz. Označme po řadě sloupce matice H jako $\mathbf{h}_1, \ldots, \mathbf{h}_n$. Připomeňme, že pro každý řádkový vektor $[c_1, \ldots, c_n]$ máme

$$[c_1, \ldots, c_n]H^T = \sum_{i=1}^{n} c_i \mathbf{p}_i^T.$$

Předpokládejme, že d je minimální počet lineárně závislých sloupců matice H. Pak existují skaláry $c_1, \ldots, c_n$, z nichž je právě d nenulových tak, že

$$[c_1, \ldots, c_n]H^T = \mathbf{0}^T.$$

Ale to neříká nic jiného, že $\mathbf{c} = c_1 \ldots c_n \in \mathcal{C}$. Protože $\mathrm{wt}(\mathbf{c}) = d$, máme $\mathrm{d}(\mathcal{C}) \leq \mathrm{wt}(\mathbf{c}) = d$. Obráceně, je-li $\mathbf{c} = c_1 \ldots c_n \in \mathcal{C}$ kódové slovo minimální délky, pak nutně $[c_1, \ldots, c_n]H^T = \mathbf{0}^T$ a tedy je $\mathrm{d}(\mathcal{C})$ sloupců z H odpovídajícím d nenulovým prvkům lineárně závislých. Je tedy $d \leq \mathrm{d}(\mathcal{C})$ tj. $d = \mathrm{d}(\mathcal{C})$. ▮

7 Pravidlo minimální vzdálenosti pro lineární kódy

Uvažme problém dekódování pro lineární kódy. Je-li $\mathcal{C}$ $[n, k]$-kód nad abecedou $\Sigma = F_q$, pak $\mathcal{C}$ obsahuje q^k kódových slov délky n a počet možných obdržených vektorů je q^n. Prohlížecí tabulka, která by pro každý možný obdržený vektor obsahovala „nejbližší“ kódové slovo by zabírala příliš velké množství paměti, dokonce pro malá n a k. Jednou z hlavních výhod používání lineárních kódů je, že existuje elegentní způsob vyhnutí se výše uvedenému problému.

Tento postup popíšeme pouze v binárním případě. Rozšíření na jiné abecedy mohutnosti $q = p^m$, kde p je prvočíslo je bezprostřední i když technicky náročnější.

Předpokládejme, že $\mathcal{C}$ je $[n, k]$-binární kód. Protože je $\mathcal{C}$ podprostor vektorového prostoru V_n binárních vektorů délky n, musí být $\mathcal{C}$ podgrupa aditivní grupy V_n.

Připomeňme, že *řád* konečné grupy G je definovaný jako mohutnost $|G|$ nosné množiny G, tj. počet prvků G a *index* $[G : S]$ podgrupy $S \subseteq G$ je počet $|G/S|$ různých (levých) tříd rozkladu G podle S. Přitom (levá) třída určená prvkem $a \in G$ má tvar $Sa = \{sa : s \in S\}$. Speciálně je přiřazení $a \mapsto Sa$ surjektivní homomorfismus $G \to G/S$. Přitom dvě třídy Sa, Sb jsou totožné právě tehdy, když $a = s_0 b$ pro některé $s_0 \in S$. Pravé násobení prvkem a je bijekce $S \to Sa$ a proto má každá (levá) třída rozkladu stejný počet prvků jako podgrupa S. Protože G je sjednocení příslušných levých tříd rozkladu, je počet prvků celé grupy G stejný jako počet tříd rozkladu krát počet prvků v S:

$$|G| = |G/S| \cdot |S|.$$

Zejména tedy kód $\mathcal{C}$ určuje soubor levých tříd (*kosetů*) podprostoru V_n a libovolný vektor $\mathbf{a} \in V_n$ určuje jediný koset $\mathbf{a} + \mathcal{C} = \{\mathbf{b} : \mathbf{b} = \mathbf{a} + \mathbf{c}$ pro vhodný vektor $\mathbf{c} \in \mathcal{C}\}$.

Předpokládejme tedy, že $\mathbf{y}$ je obdržený vektor v tom případě, že jisté kódové slovo bylo přeneseno kanálem. Řekneme, že vektor $\mathbf{e}$ je *možný chybový vektor* vektoru $\mathbf{y}$, pokud existuje kódové slovo $\mathbf{c} \in \mathcal{C}$ tak, že

$$\mathbf{y} - \mathbf{c} = \mathbf{e}.$$

Speciálně je tedy interpretace následující: vektor $\mathbf{e}$ je chybový vektor přidružený k obdrženému vektoru, jestliže je schopen reprezentovat jistou možnou posloupnost chyb při přenosu. Následující triviální pozorování je klíčové.

Lemma 7.1 *Je-li $\mathbf{y}$ obdržený vektor, je množina možných chybových vektorů ten koset množiny $\mathcal{C}$, který obsahuje vektor $\mathbf{y}$.*

Důkaz. Bylo-li obdrženo slovo $\mathbf{y}$, je vektor $\mathbf{e}$ chybový vektor právě tehdy, když existuje kódové slovo $\mathbf{c} \in \mathcal{C}$ tak, že $\mathbf{y} - \mathbf{c} = \mathbf{e}$. Ale $\mathcal{C}$ je podprostor; tedy i $-\mathbf{c} \in \mathcal{C}$, tj. $\mathbf{e} = \mathbf{y} + \mathbf{c}'$ a $\mathbf{e} \in \mathbf{y} + \mathcal{C}$. ▮

Co to je dekódování podle pravidla minimální vzdálenosti? To není nic jiného, než nalezení chybového vektoru s minimální vahou. Známe-li tedy, pro každý koset jeho prvek minimální váhy, pak máme základ pro dekódování podle pravidla minimální vzdálenosti. Řekneme pak, že vektor $\mathbf{e}$ je *reprezentant* kosetu $\mathcal{H}$, jestliže má nejmenší váhu ze všech vektorů obsažených v $\mathcal{H} = \mathbf{e} + \mathcal{C}$. Zdůrazněme, že takovýto reprezentant nemusí být vybrán jednoznačně.

Algoritmus I

Krok 1: Po přijetí vektoru $\mathbf{y}$ najdeme reprezentanta $\mathbf{z}_0$ kosetu $\mathbf{y} - \mathcal{C}$.

Krok 2: Vektor $\mathbf{y}$ pak dekódujeme jakožto kódové slovo $\mathbf{y} - \mathbf{z}_0$.

Evidentně, Krok 1 může být velmi časově náročný. Urychlíme jej použitím následující vlastností lineárních kódů.

Lemma 7.2 *Dva vektory $\mathbf{y}_1$ a $\mathbf{y}_2$ leží v témže kosetu právě tehdy, když*

$$H\mathbf{y}_1^\top = H\mathbf{y}_2^\top.$$

Důkaz. Dva vektory $\mathbf{y}_1$ a $\mathbf{y}_2$ leží v témže kosetu právě tehdy, když existuje kódové slovo $\mathbf{c} \in \mathcal{C}$ tak, že

$$\mathbf{y}_1 = \mathbf{y}_2 + \mathbf{c}$$

tj.

$$H\mathbf{c}^\top = H(\mathbf{y}_1 - \mathbf{y}_2)^\top = \mathbf{0}$$

tj.

$$H\mathbf{y}_1^\top = H\mathbf{y}_2^\top.$$

▮

Definujme *syndrom* kosetu $\mathcal{H} = \mathbf{a} + \mathcal{C}$ jakožto vektor $H\mathbf{a}^\top$ délky k. Evidentně, je tato definice korektní. Máme tedy pro každý koset $\mathcal{H}$ jeho syndrom a jeho reprezentanta kosetu. Máme-li tedy předem vypočtenou *tabulku*, ve které je pro každý syndrom určen příslušný reprezentant, můžeme urychlit výše uvedný algoritmus následovně:

Algoritmus II – poloeficientní

Krok 1a: Po přijetí vektoru $\mathbf{y}$ najdeme syndrom $H\mathbf{y}^\top$.

Krok 1b: Z výše uvedené tabulky najdeme odpovídajícího reprezentanta $\mathbf{z}_0$ kosetu $\mathbf{y} - \mathcal{C}$.

Krok 2: Vektor $\mathbf{y}$ pak dekódujeme jakožto kódové slovo $\mathbf{y} - \mathbf{z}_0$.

Platí pak

Věta 7.3 *Algoritmus II pracuje jakožto dekódovací pravidlo podle minimální vzdálenosti pro lineární kód $\mathcal{C}$.*

Důkaz. Poznamenejme nejprve, že každé obdržené slovo $\mathbf{y}$ můžeme dekódovat jakožto kódové slovo. To je z toho důvodu, že $\mathbf{y}$ a $\mathbf{z}_0$ jsou ve stejném kosetu a tedy $\mathbf{y} - \mathbf{z}_0 \in \mathcal{C}$. Předpokládejme, že existuje kódové slovo $\mathbf{c}$ tak, že

$$\mathrm{d}(\mathbf{y}, \mathbf{y} - \mathbf{z}_0) > \mathrm{d}(\mathbf{y}, \mathbf{c}).$$

To je ekvivalentní s tím, že

$$\mathrm{d}(\mathbf{0}, \mathbf{z}_0) > \mathrm{d}(\mathbf{y} - \mathbf{c}, \mathbf{0})$$

tj. $\mathrm{w}(\mathbf{z}_0) > \mathrm{w}(\mathbf{y} - \mathbf{c})$.

Ale pak

$$H(\mathbf{y} - \mathbf{c})^\top = H\mathbf{y}^\top - H\mathbf{c}^\top = H\mathbf{y}^\top,$$

protože $\mathbf{c}$ je kódové slovo. Zejména tedy má $\mathbf{y} - \mathbf{c}$ stejný syndrom jako $\mathbf{y}$, leží ve stejném kosetu a a má váhu ostře menší než je váha reprezentanta kosetu $\mathbf{z}_0$, což je spor. ▮

V dalším budeme předpokládat, že kódování a dekódování pro lineární $[n, k]$-kód $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_M, \mathbf{c}_1 = \mathbf{0}$ při přenosu binárním symetrickým kanálem s pravděpodobností chyby p bude probíhat pomocí následující tabulky

Dále předpokládejme, že každý reprezentant $\mathbf{f}_i$ příslušného kosetu má váhu $wt(\mathbf{f}_i)$. Platí pak následující tvrzení. Označme, pro $1 \leq j \leq n$, w_j počet reprezentantů váhy j.

Věta 7.4 *Buď $\mathcal{C}$ binární lineární $[n, k]$-kód. Pak pravděpodobnost správného dekódování při přenosu binárním symetrickým kanálem je*

$$P(\text{správné dekódování}) = \sum_{i=1}^{s} p^{wt(\mathbf{f}_i)}(1-p)^{n-wt(\mathbf{f}_i)}$$

tj.

$$P(\text{správné dekódování}) = \sum_{j=1}^{n} w_j p^j (1-p)^{n-j}.$$

Důkaz. Protože reprezentant $\mathbf{f}_i$ příslušného kosetu má váhu $wt(\mathbf{f}_i)$, pravděpodobnost, že nám vznikne z $\mathbf{c}$ slovo $\mathbf{d}$ je stejná, že nám vznikne z $\mathbf{0}$ příslušný reprezentant $\mathbf{f}_i$ tj.

$$P(\text{reprezentant je } \mathbf{f}_i) = p^{wt(\mathbf{f}_i)}(1-p)^{n-wt(\mathbf{f}_i)}.$$

Posčítáme-li přes všechny reprezentanty, obdržíme požadované tvrzení. ▮

Nevýhody této kódovací metody lze nejlépe vidět na následujícím příkladu.

Příklad 7.5 Předpokládejme, že C je binární kód, jehož generující matice G je určena následovně

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

Pak kontrolní matice H má tvar

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

Kódová slova kódu $\mathcal{C}$ jsou

$$0000, 1010, 0111, 1101.$$

Příslušná prohlížecí tabulka má tvar

Syndrom $\mathbf{s}$	Reprezentant $\mathbf{z}_0$ kosetu $\mathcal{H}$, $s = H\mathbf{z}_0^\top$
00	0000
10	0010
01	0001
11	0100

.

$\mathbf{0}$	$\mathbf{c}_2$	$\mathbf{c}_3$	$\cdots$	$\mathbf{c}_M$
$\mathbf{f}_2$	$\mathbf{f}_2 + \mathbf{c}_2$	$\mathbf{f}_2 + \mathbf{c}_3$	$\cdots$	$\mathbf{f}_2 + \mathbf{c}_M$
$\mathbf{f}_3$	$\mathbf{f}_3 + \mathbf{c}_2$	$\mathbf{f}_3 + \mathbf{c}_3$	$\cdots$	$\mathbf{f}_3 + \mathbf{c}_M$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{f}_s$	$\mathbf{f}_s + \mathbf{c}_2$	$\mathbf{f}_s + \mathbf{c}_3$	$\cdots$	$\mathbf{f}_s + \mathbf{c}_M$

Předpokládejme tedy, že jsme obdrželi vektor $\mathbf{y} = 1111$. Je ho syndrom je vektor $H\mathbf{y}^\top = 10$. Odpovídající reprezentant je 0010, je tedy slovo 1111 dekódováno jakožto 1101. Poznamenejme, že tato prohlížecí tabulka není určena jednoznačně, například za reprezentanta syndromu 10 lze vzít vektor 0111.

V případě binárního $[n, k]$-kódu máme právě $|V_n|/|\mathcal{C}| = 2^{n-k}$ (obecně pak q^{n-k}) různých kosetů; zejména tedy bude mít prohlížecí tabulka v Kroku 1(b) právě 2^{n-k} různých položek. Prohledávání takovéto tabulky je pro velká k, n velmi náročné. Avšak ostatní výhody této metody opravňují její široké používání.

8 Binární Hammingovy kódy

Abychom ilustrovali dříve uvedené techniky, uvažme následující příklad. Omezme naši pozornost na binární příklad; buď r nějaké kladné celé číslo a položme $n = 2^r - 1$. Dále definujme kontrolní matici H jakožto matici typu $r \times (2^r - 1)$, jejíž sloupce tvoří všechny navzájem různé nenulové vektory z V_r. Pak H je kontrolní matice binárního $[n, k]$-kódu, kde

$$n = 2^r - 1, \quad k = n - r.$$

Mluvíme pak o *Hammingově* $[n, k]$*-kódu* .

Klíčovou vlastnost Hammingových kódů lze zformulovat v následující větě.

Věta 8.1 *Každý Hammingův kód je perfektní kód opravující jednu chybu.*

Důkaz. Nejprve ukažmě, že minimální vzdálenost každého Hammingova kódu je alespoň 3. Protože $\mathcal{C}$ je lineární kód, je minimální vzdálenost $\mathrm{d}(\mathcal{C})$ rovna minimální váze vektorů z $\mathcal{C}$.

Předpokládejme nejprve, že $\mathcal{C}$ má kódové slovo $\mathbf{u}$ váhy 1 s nenulovým vstupem v i-té souřadnici. Pak platí

$$H\mathbf{u}^\top = \mathbf{0},$$

tj. i-tý sloupec $\mathbf{h}_i$ matice H je nulový, což není z definice matice H možné. Předpokládejme dále, že $\mathcal{C}$ má kódové slovo $\mathbf{v}$ váhy 2 s nenulovými vstupy v i-té a j-té souřadnicích. Pak platí

$$H\mathbf{v}^\top = \mathbf{0},$$

tj.

$$\mathbf{h}_i + \mathbf{h}_j = \mathbf{0}.$$

Protože pracujeme s binárními kódy, je nutně

$$\mathbf{h}_i = \mathbf{h}_j,$$

což není možné. je tedy $\mathrm{d}(\mathcal{C}) \geq 3$. Ukažme, že $\mathcal{C}$ je perfektní. Poznamenejme, že každá 1-koule kolem kódového slova bude obsahovat právě $1 + n = 2^r$ vektorů délky $n = 2^r - 1$. Protože $\mathcal{C}$ obsahuje právě $2^k = 2^{n-r}$ kódových slov, disjunktní sjednocení těchto 1-koulí je právě celá množina V_n vektorů délky n, jichž je právě $2^n = 2^{n-r} \cdot 2^r$. ▮

Důležitým důsledkem perfektnosti Hammingových kódů je, že

1. Pro Hammingův $[n, k]$-kód jsou reprezentanti kosetů vektory z V_n váhy ≤ 1.

   To vede k následujícímu elegantnímu dekódovacímu algoritmu pro Hammingovy kódu. Nejprve poznamenejme:

2. Sloupce matice H lze přemístit tak, že j-tý sloupec matice H je právě binární reprezentace čísla j.

   Je-li obdržen vektor $\mathbf{y}$, spočtěme jeho syndrom $H\mathbf{y}^\perp$ a předpokládejme, že reprezentuje číslo j. Předpokládáme-li pouze jednu chybu, pravidlo minimální vzdálenosti (=pravidlo maximální pravděpodobnosti) nám dává:

   (a) Pokud $j = H\mathbf{y}^\perp = \mathbf{0}$, pak nepředpokládáme žádnou chybu a $\mathbf{y}$ je kódové slovo.

   (b) Pokud $j = H\mathbf{y}^\perp \neq \mathbf{0}$, pak předpokládáme chybu v j-té pozici a dekódujeme $\mathbf{y}$ jeho změnou v j-té pozici.

Příklad 8.2 Hammingův $[7, 4]$-kód má matici kontroly parity

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

Předpokládejme, že jsme obdrželi vektor $\mathbf{y} = (1, 0, 1, 0, 1, 1, 0)$. Pak $H\mathbf{y}^\top = (001)$. Tedy za předpokladu, že nenastala více než jedna chyba, předpokládáme, že se chyba vyskytla na prvním místě a dekódujeme pak $\mathbf{y}$ jakožto $\mathbf{y}^*$,

$$\mathbf{y}^* = (0, 0, 1, 0, 1, 1, 0).$$

Cvičení 8.3

1. *Napište matici kontroly parity binárního* $[15, 11, 3]$*-kódu. Jak bychom dekódovali obdržené vektory:*

   (a) (100000000000000),

   (b) (111111111111111)*?*

9 Cyklické kódy

Diskutujme nyní důležitou skupinu lineárních kódů. Kód $\mathcal{C}$ se nazývá *cyklický*, jestliže platí následující podmínky:

1. $\mathcal{C}$ je lineární,
2. pokud vektor $\mathbf{w} = (w_1, \ldots, w_n) \in \mathcal{C}$, pak i vektor $\mathbf{w}' = (w_n, w_1, \ldots, w_{n-1}) \in \mathcal{C}$.

Tyto kódy mají atraktivní algebraické vlastnosti a můžeme je snadno sestrojit pomocí lineárních posouvacích registrů (blíže viz [7]).

Budeme dále pracovat pouze v binárním případě a během tohoto paragrafu budeme identifikovat vektor

$$\mathbf{w} = (w_1, \ldots, w_n)$$

s polynomem

$$w(x) = w_1 + w_2x + w_3x^2 + \ldots + w_nx^{n-1}.$$

Dále budeme počítat pouze v okruhu R_n binárních polynomů stupně nejvýše $n-1$ modulo polynom $x^n - 1$. Tedy R_n se skládá z polynomů stupně $\leq n-1$ s koeficienty 0 a 1 tak, že platí následující pravidla pro sčítání a násobení polynomů:

$$\begin{aligned} a(x) + b(x) &= \textstyle\sum_{i=0}^{n-1}(a_i + b_i)x^i \\ a(x) \cdot b(x) &= a(x)b(x) \bmod (x^n - 1). \end{aligned}$$

Základním pozorováním je následující skutečnost: posunu v kódovém slově odpovídá násobení odpovídajícího polynomu monomem x v okruhu R_n. Totiž $w(x) \cdot x = w_1x + w_2x^2 + w_3x^3 + \ldots + w_{n-1}x^{n-1} + w_nx^n \bmod (x^n - 1) = w_1x + w_2x^2 + w_3x^3 + \ldots + w_{n-1}x^{n-1} + w_nx^0$.

Platí pak následující lemma

Lemma 9.1 *Je-li $w(x)$ polynomiální reprezentace kódového slova $\mathbf{w} \in \mathcal{C}$, je i $w(x)f(x)$ kódové slovo pro každý polynom f stupně nejvýše $n-1$.*

Důkaz. Protože $w(x) \in \mathcal{C}$, je i $xw(x) \in \mathcal{C}$ (posunutí o 1 místo doprava). Nutně tedy pro každé přirozené číslo k platí, že $x^kw(x) \in \mathcal{C}$. Ale protože je $\mathcal{C}$ lineární kód, je i libovolná lineární kombinace kódových slov tvaru $x^kw(x)$ opět v $\mathcal{C}$, zejména tedy je polynom $f(x)w(x)$ v $\mathcal{C}$. ▮

Lemma 9.2 *Je-li $g(x)$ nenulový polynom minimálního stupně v $\mathcal{C}$, pak $g(x)$* generuje *kód $\mathcal{C}$ v tom smyslu, že každé kódové slovo $w(x) \in \mathcal{C}$ je tvaru*

$$w(x) = f(x)g(x)$$

pro vhodný polynom $f(x)$.

Důkaz. Předpokládejme, že existuje $w(x) \in \mathcal{C}$, které nelze napsat ve výše uvedeném tvaru. Pak lze psát

$$w(x) = q(x)g(x) + r(x),$$

kde $r(x)$ je zbytek po dělení polynomem $g(x)$ tj. jeho stupeň je menší než stupeň polynomu $g(x)$. Ale nutně $q(x)g(x), w(x) \in \mathcal{C}$ tj. $r(x) \in \mathcal{C}$ tj. nutně je $r(x)$ nulový polynom, spor. Tedy je každý polynom z $\mathcal{C}$ násobkem $g(x)$. ▮

Mluvíme pak o polynomu $g(x)$ jakožto o *generujícím polynomu* kódu $\mathcal{C}$. Tím pak dostaneme velmi dobrou reprezentaci kódu $\mathcal{C}$. Připomeňme dále, že cyklický kód není nic jiného než ideál v okruhu polynomu a 9.2 plyne bezprostředně z toho, že každý ideál v okruhu polynomů je hlavní ideál.

Příklad 9.3 Předpokládejme, že $n = 3$ a násobení je prováděno modulo polynom $x^3 - 1$. Pak kód

$$\mathcal{C} = \{0, 1 + x, x + x^2, 1 + x^2\}$$

je cyklický a generován polynome $1 + x$. Standardní reprezentace kódu $\mathcal{C}$ sestává z vektorů

$$\{(0, 0, 0), (1, 1, 0), (0, 1, 1), (1, 0, 1)\}.$$

▮

Lemma 9.4 *Je-li $\mathcal{C}$ cyklický kód délky n s generujícím polynomem $g(x) = g_1 + g_2x + \ldots + g_kx^{k-1}$, pak jeho generující matice typu $(n - k + 1) \times n$ má tvar*

$$G = \begin{bmatrix} g_1 & g_2 & g_3 & \ldots & g_k & 0 & 0 & \ldots & 0 \\ 0 & g_1 & g_2 & \ldots & g_{k-1} & g_k & 0 & \ldots & 0 \\ 0 & 0 & g_1 & \ldots & g_{k-2} & g_{k-1} & g_k & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \ldots & \ldots & \ldots & g_{k-2} & g_{k-1} & g_k \end{bmatrix}.$$

Důkaz. Evidentně, řádky matice G jsou lineárně nezávislé. Ukažme, že každé kódové slovo lze reprezentovat pomocí těchto řádků. Je-li tedy $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ kódové slovo, je odpovídající polynom

$$c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$$

tvaru

$$c(x) = g(x)f(x)\ (\mathrm{mod}(x^{n1} - 1))$$

pro jistý polynom f stupně nejvýše $\leq n - 1$. Ale to neznamená nic jiného, než že

$$c(x) = \sum_{i=0}^{n-1} f_ix^ig(x)\ (\mathrm{mod}(x^{n1} - 1));$$

tj. položíme-li $g^{(i)}(x) = x^ig(x)$ a je-li $\mathbf{g}^{(i)}$ odpovídající reprezentace polynomu $g^{(i)}(x)$ ($i + 1$-ní řádek matice G), máme

$$\mathbf{c} = \sum_{i=0}^{n-1} f_i\mathbf{g}^{(i)}.$$

▮

Lemma 9.5 *Je-li g generující polynom cyklického kódu délky n $\mathcal{C}$, pak g dělí polynom* $(x^{n1}-1)$.

Důkaz. Předpokládejme, že tomu tak není. Můžeme pak psát

$$x^n - 1 = g(x)q(x) + r(x),$$

kde $r(x)$ je nenulový polynom se stupněm menším než je stupeň g. Protože $q(x)f(x) \in \mathcal{C}$ a $r = -qg$ v tomto okruhu, plyne z linearity, že i r je kódové slovo, tj. se nemůže jednat o polynom minimálního stupně a tedy $r = 0$, spor. ▮

Lemma 9.6 *Je-li dán polynom p stupně $< n$, pak množina všech polynomů $\mathcal{C} = \{qp(\mathrm{mod}(x^{n1}-1)) : q$ je polynom stupně $< n\}$ je cyklický kód délky n.*

Důkaz. Evidentně, $\mathcal{C}$ je lineární kód. Zároveň, je-li $qp \in \mathcal{C}$, je i $xqp \in \mathcal{C}$ tj. $\mathcal{C}$ je cyklický kód. ▮

10 Marinerův kód a Reed-Mullerovy kódy

V tomto odstavci se budeme věnovat dalším dvěma příkladům, kdy pomocí klasické moderní algebry byly zkonstruovány a vyvinuty nové třídy kódů.

10.1 Hadamardovy kódy

Kódování $R(1,5)$ použité v roce 1969 kosmickým korábem Mariner 9 pro přenos fotografií z Marsu je speciálním případem následujících obecných kódů.

Nejprve si připomeňme některé pojmy z moderní algebry. *Hadamardova matice* je matice H typu $n \times n$, jejíž koeficienty jsou buď $+1$ nebo -1 tak, že

$$HH^T = nE_n, \tag{4.12}$$

kde E_n je jednotková matice typu $n \times n$.

Jsou-li dále A a B čtvercové matice typu $m \times m$ a $n \times n$, definujeme jejich *Kroneckerův součin* jako matici $A \otimes B$ typu $mn \times mn$

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{bmatrix}. \tag{4.13}$$

Přímým výpočtem pak snadno dokážeme:

Lemma 10.1 *Jsou-li H_1 a H_2 Hadamardovy matice, je jejich Kroneckerův součin $H_1 \otimes H_2$ Hadamardova matice.*

Důkaz. Počítejme $G = (H_1 \otimes H_2)(H_1 \otimes H_2)^T$. Pak g_{ij}, $i = \bar{i} + n\cdot(\hat{i}-1)$, $j = \bar{j} + n\cdot(\hat{j}-1)$. Zejména $g_{ij} = \sum_{l=1}^m a_{\hat{i}l}a_{\hat{j}l}(\sum_{k=1}^n b_{\bar{i}k}b_{\bar{j}k})$. Pokud $i = j$, je nutně $\hat{i} = \hat{j}$ a $\bar{i} = \bar{j}$ a tedy i $g_{ii} = \sum_{l=1}^m a_{\hat{i}l}a_{\hat{i}l}(\sum_{k=1}^n b_{\bar{i}k}b_{\bar{i}k})$. Ale $\sum_{k=1}^n b_{\bar{i}k}b_{\bar{i}k} = n$ a tedy $g_{ii} = \sum_{l=1}^m a_{\hat{i}l}a_{\hat{i}l}n = mn$. Nechť tedy $i \neq j$. Pak buď $\bar{i} \neq \bar{j}$ nebo $\hat{i} \neq \hat{j}$. Nechť například $\hat{i} \neq \hat{j}$. Pak $g_{ij} = (\sum_{l=1}^m a_{\hat{i}l}a_{\hat{j}l})(\sum_{k=1}^n b_{\bar{i}k}b_{\bar{j}k}) = 0(\sum_{k=1}^n b_{\bar{i}k}b_{\bar{j}k}) = 0.$. Nechť tedy $\bar{i} \neq \bar{j}$. Podobně, $g_{ij} = (\sum_{l=1}^m a_{\hat{i}l}a_{\hat{j}l})(\sum_{k=1}^n b_{\bar{i}k}b_{\bar{j}k}) = (\sum_{l=1}^m a_{\hat{i}l}a_{\hat{j}l})0 = 0.$ ▮

Začneme-li tedy s nejmenší netriviální Hadamardovou maticí

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

můžeme postupně iterovat Kroneckerovým součinem, abychom obdrželi posloupnost Hadamardových matic s exponenciálně rostoucím typem. Chceme-li pak tuto posloupnost použít pro účely kódování, předpokládejme, že H je Hadamardova matice rozměru $n \times n$, přičemž n je sudé. Definujme pak A jakožto matici typu $2n \times n$

$$A = \begin{bmatrix} H \\ -H \end{bmatrix}.$$

Pak definujme M jakožto matici, kterou získáme z matice A tak, že nahradíme každý výskyt -1 v A číslem 0.

Snadno se ověří následující tvrzení

Lemma 10.2 Hadamardovo kódování

1. *Jsou-li $\mathbf{x}$ a $\mathbf{y}$ dva různé řádky matice M, je pak vzdálenost $d(\mathbf{x},\mathbf{y})$ vektorů $\mathbf{x}$ a $\mathbf{y}$ rovna číslu $\frac{n}{2}$ nebo n.*

2. *Řádky matice M tvoří binární $(n, 2n, \frac{n}{2})$-kód.*

Důkaz. Snadné cvičení. Totiž, vezmeme-li 2 různé řádky vzniklé z H, nutně je počet míst, kde se řádky liší, roven počtu míst, kde jsou oba řádky stejné, tj. $\mathrm{d}(\mathbf{x},\mathbf{y}) = \frac{n}{2}$. Podobně, vezmeme-li 2 různé řádky, z nichž jeden vznikl z H a druhý z $-H$ a zároveň oba z různých vektorů z H, je nutně opět počet míst, kde se řádky liší, roven počtu míst, kde jsou oba řádky stejné, tj. opět $\mathrm{d}(\mathbf{x},\mathbf{y}) = \frac{n}{2}$. Případ, kdy oba řádky vznikly ze stejného vektoru, nám dává vzdálenost rovnu n. Poslední případ, kdy oba řádky vznikly z $-H$, se převede na první případ. Zbývající část tvrzení je triviální. ▮

Provedeme-li výše uvedené pětkrát za sebou na matici H_2, obdržíme $n = 32$ a to je přesně kódování použité Marinerem. Kódy získané výše uvedeným postupem se nazývají *Hadamardovy kódy*.

10.2 Reed-Mullerovo kódování

Tato prakticky důležitá třída kódů byla objevena I.S. Reedem a D.E. Mullerem v roce 1954. Abychom mohli popsat tyto kódy, budeme nejprve prezentovat jednoduchý způsob zkonstruování nových kódování ze starých původních.

Lemma 10.3 *Je-li dán (n, M_1, d_1) binární kódování $\mathcal{C}_1$ a jiné (n, M_2, d_2) binární kódování $\mathcal{C}_2$, můžeme pak definovat třetí binární kódování $\mathcal{C}_3 = \mathcal{C}_1 * \mathcal{C}_2$ jakožto*

$$\mathcal{C}_3 = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) : \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}.$$

Přitom $\mathcal{C}_3$ je $(2n, M_1 M_2, d_3)$-kód, kde

$$d_3 = min\{2d_1, d_2\}. \tag{4.14}$$

Důkaz. Totiž délka kódových slov v $\mathcal{C}_3$ je nutně $2n$ a snadno se ověří, že jich je právě $M_1 M_2$. To plyne z toho, že pokud $(\mathbf{u}_1, \mathbf{u}_1 + \mathbf{v}_1) = (\mathbf{u}_2, \mathbf{u}_2 + \mathbf{v}_2)$ je nutně $\mathbf{u}_1 = \mathbf{u}_2$ a tedy i $\mathbf{v}_1 = \mathbf{v}_2$. Takovýchto uspořádaných dvojic $(\mathbf{u}, \mathbf{v})$ je právě $M_1 M_2$. Zbývá ověřit, že minimální délka kódování $\mathcal{C}_3$, kterou značíme d_3, je rovna $\min\{2d_1, d_2\}$. To je ale zřejmé. Totiž, je-li $(\mathbf{u}_1, \mathbf{u}_1 + \mathbf{v}_1) \neq (\mathbf{u}_2, \mathbf{u}_2 + \mathbf{v}_2)$, pak mohou nastat následující případy

1. $\mathbf{u}_1 = \mathbf{u}_2$: pak $\mathrm{d}(\mathbf{v}_1, \mathbf{v}_2) = \mathrm{d}((\mathbf{u}_1, \mathbf{u}_1 + \mathbf{v}_1), (\mathbf{u}_2, \mathbf{u}_2 + \mathbf{v}_2)) \geq d_2$.
2. $\mathbf{v}_1 = \mathbf{v}_2$: pak $\mathrm{d}(\mathbf{u}_1, \mathbf{u}_2) + \mathrm{d}(\mathbf{u}_1, \mathbf{u}_2) = \mathrm{d}((\mathbf{u}_1, \mathbf{u}_1 + \mathbf{v}_1), (\mathbf{u}_2, \mathbf{u}_2 + \mathbf{v}_2)) \geq 2d_1$.
3. $\mathbf{v}_1 \neq \mathbf{v}_2$ a $\mathbf{v}_1 \neq \mathbf{v}_2$: pak označíme-li $I_n = \{i : \pi_i(u_1) \neq \pi_i(u_2)\}$, $I_e = \{i : \pi_i(u_1) = \pi_i(u_2)\}$, $J_n = \{i : \pi_i(v_1) \neq \pi_i(v_2)\}$, $J_e = \{i : \pi_i(v_1) = \pi_i(v_2)\}$, je $\mathrm{d}((\mathbf{u}_1, \mathbf{u}_1 + \mathbf{v}_1), (\mathbf{u}_2, \mathbf{u}_2 + \mathbf{v}_2)) = |I_n| + |J_e \cap I_n| + |J_n \cap I_e| = |J_n| + 2|J_e \cap I_n| \geq d_2$.

Přitom v prvním a druhém případě může pro vhodně vybrané dvojice nastat rovnost, tj. tvrzení platí. ▮

Definujme nyní rekurzivně *Reed-Mullerův kód* $\mathcal{C}(r, m)$ předpisem:

Pro všechna nezáporná celá čísla m a r taková, že $0 \leq r \leq m$, definujeme $\mathcal{C}(r, m)$ jakožto kód délky $n = 2^m$ takový, že

$$\mathcal{C}(0, m) = \{\mathbf{0}, \mathbf{1}\}, \tag{4.15}$$

kde $\mathbf{0} = (0, 0, \ldots, 0)$ a $\mathbf{1} = (1, 1, \ldots, 1)$, $\mathcal{C}(m, m)$ je množina všech binárních vektorů délky $n = 2^m$ tj. $\mathcal{C}(m, m) = \mathbf{2}^{2^m}$ a

$$\mathcal{C}(r + 1, m + 1) = \mathcal{C}(r + 1, m) * \mathcal{C}(r, m) \tag{4.16}$$

pro $r < m$. Můžeme pak tyto kódy konstruovat následovně:

$$\begin{array}{lllll} m = 1 & \mathcal{C}(0, 1) & = & \{00, 11\} \\ & \mathcal{C}(1, 1) & = & \{00, 01, 10, 11\} \\ m = 2 & \mathcal{C}(0, 2) & = & \{0000, 1111\} \\ & \mathcal{C}(1, 2) & = & \mathcal{C}(1, 1) * \mathcal{C}(0, 1) \\ & & = & \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\} \end{array}$$

atd. Aplikujeme-li nyní lemma 10.3, obdržíme následující větu:

Věta 10.4 *Pro všechna nezáporná celá čísla m a r taková, že $0 \leq r \leq m$ je Reed-Mullerův $\mathcal{C}(r, m)$ binární kód charakteristiky (n_r, M_r, d_r), kde*

1. $M_r = 2^a$, *kde* $a = \sum_{i=0}^{r} \binom{m}{i} = 1 + \binom{m}{1} + \ldots + \binom{m}{r}$,
2. $n_r = 2^m$,
3. $d_r = 2^{m-r}$.

Důkaz. Důkaz povedeme indukcí vzhledem k definici $\mathcal{C}(r, m)$. Totiž pro $\mathcal{C}(0, m) = \{\mathbf{0}, \mathbf{1}\}$ je počet slov M_0 roven dvěma a protože zároveň nutně $a = 1$, první část tvrzení pro $\mathcal{C}(0, m)$ platí. Protože délka n_r vektorů je dle definice 2^m, platí druhá část tvrzení. Protože kód obsahuje pouze dva vektory, které se liší na $n_r = 2^m$ místech, je vzdálenost kódu rovna $n_r = 2^{m-0} = d_r$. Uvažme nyní kód $\mathcal{C}(m, m)$, což je množina všech binárních vektorů délky $n = 2^m = n_m$. Je tedy v našem případě $a = n$ a tedy i $M_m = 2^a$.Vzdálenost kódu je pak nutně $1 = 2^{m-m}$. Věnujme se nyní případu $\mathcal{C}(r + 1, m + 1) = \mathcal{C}(r + 1, m) * \mathcal{C}(r, m)$. Z indukčního předpokladu a lemmatu 10.3 víme, že

$$M_{r+1} = 2^{\sum_{i=0}^{r+1} \binom{m}{i}} \cdot 2^{\sum_{i=0}^{r} \binom{m}{i}} = 2^{\sum_{i=0}^{r+1} \binom{m}{i} + \sum_{i=0}^{r} \binom{m}{i}} = 2^{\sum_{i=0}^{r+1} \binom{m}{i}}.$$

Dále víme, že $n_{r+1} = 2 \cdot 2^m = 2^{m+1}$ a $d_{r+1} = \min\{2 \cdot 2^{m-r-1}, 2^{m-r}\} = 2^{m-r} = 2^{m+1-(r+1)}$. ▮

Problémy 4.1 *1. Ukažte, že pokud platí*

(a) $2^k \sum_{i=0}^{d-2} \binom{n-1}{i} < 2^n$,

pak existuje binární lineární $[n, k]$-kód s minimální vzdáleností alespoň d. Tudíž odvoďte, že *(a)* $2^k \leq A_2(n, d)$,

kde k *je největší přirozené číslo splnující nerovnost (1). Návod: Zkonstruujte matici* H *typu* $(n-k) \times n$, *takovou, že její hodnost je nejvýše* $d-2$.

2. *Ukažte, že je-li* H *Hadamardova matice řádu* n, *je nutně* $n = 1, 2$ *nebo je* n *násobek* 4. *Poznamenejme, že existuje hypotéza, že pokud* n *je násobek čtyř, existuje Hadamardova matice řádu* n.

Příloha A

Náhodné jevy a náhodné veličiny

Tento dodatek obsahuje základní pojmy nutné pro pochopení probírané látky. Je založen na skriptech [4].

1 Měřitelný prostor a vztahy mezi náhodnými jevy

Neprázdnou množinu možných výsledků náhodného pokusu značíme Ω a nazýváme ji *základní prostor* . Možné výsledky náhodného pokusu značíme ω_t, $t \in T$, kde T je indexová množina.

Systém podmnožin $\mathbf{A}$ základního prostoru Ω, který

1. obsahuje základní prostor,
2. s každými dvěma množinami obsahuje i jejich rozdíl,
3. obsahuje-li každou ze spočetné posloupnosti množin, obsahuje i jejich sjednocení

se nazývá *jevové pole* .

Poznamenejme, že má-li základní prostor alespoň dva prvky, není jevové pole uvedenými třemi axiomy určeno jednoznačně.

Je-li $A \in \mathbf{A}$, řekneme, že A je *náhodný jev* vzhledem k jevovému poli $\mathbf{A}$. Dvojici $(\Omega, \mathbf{A})$ nazveme *měřitelný prostor* , množinu Ω pak nazveme *jistý jev*, množinu $\emptyset$ *nemožný jev*. Prvek $\omega \in \Omega$ nazveme *elementární jev*. Nechť I je libovolná neprázdná indexová množina. Pak $\bigcap_{i\in I} A_i$ značí *společné nastoupení náhodných jevů* A_i, $i \in I$ a $\bigcup_{i\in I} A_i$ značí *nastoupení alespoň jednoho z náhodných jevů* A_i, $i \in I$. $\overline{A_i} = \Omega - A_i$ značí *opačný náhodný jev k náhodnému jevu* A_i, $i \in I$. Přitom to, že $\omega \in A_i$, $i \in I$ znamená, že *možný výsledek* ω *je příznivý jevu* A_i.

Nechť $i, j \in I$. Pak $A_i - A_j$ znamená *nastoupení jevu* A_i *za nenastoupení jevu* A_j, $A_j \subseteq A_i$ znamená, že *náhodný jev* A_j *má za důsledek náhodný jev* A_i a $A_i \cap A_j = \emptyset$ znamená, že náhodné jevy A_i a A_j *jsou neslučitelné*.

2 Pravděpodobnostní prostor

Nechť $(\Omega, \mathbf{A})$ je měřitelný prostor. *Pravděpodobností* rozumíme reálnou množinovou funkci $P : \mathbf{A} \to \mathbf{R}$, která je

1. nezáporná (pro všechna $A \in \mathbf{A}$ je $P(A) \geq 0$),
2. spočetně aditivní ($\forall_{i=1}^{\infty}\forall_{j=i+1}^{\infty} A_i \cap A_j = \emptyset \implies P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$),
3. normovaná ($P(\Omega) = 1$).

Trojice $(\Omega, \mathbf{A}, P)$ se nazývá *pravděpodobnostní prostor*. Za předpokladu, že $\mathbf{A} \neq \{\emptyset, \Omega\}$, není pravděpodobnost P uvedenými třemi axiomy jednoznačně určena.

3 Klasická pravděpodobnost

Nechť základní prostor Ω je konečná neprázdná množina a nechť jevové pole $\mathbf{A}$ obsahuje všechny podmnožiny základního prostoru tj. $\mathbf{A} = 2^{\Omega}$. Označme $m(\Omega) = |\Omega|$ počet všech možných výsledků a pro libovolný jev $A \in \mathbf{A}$ označme $m(A) = |A|$ počet možných výsledků příznivých jevu A. Pak reálnou funkci $P : \mathbf{A} \to \mathbf{R}$ definovanou pro všechna $A \in \mathbf{A}$ vztahem

$$P(A) = \frac{m(A)}{m(\Omega)}$$

nazveme *klasická pravděpodobnost*. Všem elementárním jevům pak přiřazujeme stejnou pravděpodobnost $\frac{1}{m(\Omega)}$. Není-li tato podmínka splněna, nelze klasickou pravděpodobnost použít.

Věta 3.1 *Buďte* $A_1, \ldots, A_n \in \mathbf{A}$ *libovolné náhodné jevy. Pak platí*

$$\begin{aligned} P(\textstyle\bigcup_{i=1}^{n} A_i) &= \textstyle\sum_{i=1}^{n} P(A_i) - \sum_{1\leq i<j\leq n} P(A_i \cap A_j) + \ldots + \\ & (-1)^l \textstyle\sum_{1\leq i_1<\ldots<i_l\leq n} P(A_{i_1} \cap \ldots \cap A_{i_l}) + \ldots + (-1)^{n-1} P(A_1 \cap \ldots \cap A_n). \end{aligned}$$

4 Podmíněná pravděpodobnost

Nechť $(\Omega, \mathbf{A}, P)$ je pravděpodobnostní prostor, $H \in \mathbf{A}$ náhodný jev s nenulovou pravděpodobností. Pro každé $A \in \mathbf{A}$ definujeme *podmíněnou pravděpodobnost* vzorcem

$$P(A|H) = \frac{P(A \cap H)}{P(H)}.$$

Věta 4.1 Věta o násobení pravděpodobností *Nechť* $(\Omega, \mathbf{A}, P)$ *je pravděpodobnostní prostor,* $A_1, \ldots, A_n \in \mathbf{A}$ *náhodné jevy takové, že* $P(A_1 \cap \ldots \cap A_{n-1}) > 0$. *Pak platí*

$$P(A_1 \cap \ldots \cap A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_1 \cap A_2) \cdot \ldots \cdot P(A_n|A_1 \cap A_2 \cap \ldots \cap A_{n-1}).$$

Nechť $(\Omega, \mathbf{A}, P)$ je pravděpodobnostní prostor a nechť je dán rozklad $(H_i : i \in I)$ základního prostoru Ω na nejvýše spočetně mnoho jevů H_i o nenulových pravděpodobnostech $P(H_i)$. Říkáme, že je dán *úplný systém hypotéz*. Potom

1. pro libovolný jev $A \in \mathbf{A}$ platí *formule úplné pravděpodobnosti:*
$$P(A) = \sum_{i \in I} P(H_i) \cdot P(A|H_i),$$
2. pro náhodný jev A s nenulovou pravděpodobností a pro libovolný jev H_k z úplného systému hypotéz platí *1. Bayesův vzorec*:
$$P(H_k|A) = \frac{P(H_k) \cdot P(A|H_k)}{\sum_{i \in I} P(H_i) \cdot P(A|H_i)},$$
3. pro náhodný jev A s nenulovou pravděpodobností a pro libovolný jev $B \in \mathbf{A}$ platí *2. Bayesův vzorec*:
$$P(B|A) = \frac{\sum_{i \in I} P(H_i) \cdot P(A|H_i) \cdot P(B|A \cap H_i)}{\sum_{i \in I} P(H_i) \cdot P(A|H_i)}.$$

Použití 1. Bayesova vzorce: V souladu s textem úlohy stanovíme jevy H_i, $i \in I$, které se navzájem vylučují a přitom vyčerpávají všechny možnosti. Jeden z nich musí být náhodný jev, jehož pravděpodobnost nás zajímá. Jev, o němž je v úloze řečeno, že skutečně nastal, označíme A. Pak pro $i \in I$ vypočteme apriorní pravděpodobnosti $P(H_i)$ a podmíněné pravděpodobnosti $P(A|H_i)$. Dosazením do 1. Bayesova vzorce vypočteme aposteriorní pravděpodobnost $P(H_k|A)$.
Použití 2. Bayesova vzorce: Stanovíme opět úplný systém hypotéz H_i, $i \in I$. Jev, který skutečně nastal, označíme A a náhodný jev, na jehož pravděpodobnost se ptáme, označíme B. Pak pro $i \in I$ vypočteme apriorní pravděpodobnosti $P(H_i)$ a podmíněné pravděpodobnosti $P(A|H_i)$ a $P(B|A \cap H_i)$. Dosazením do 2. Bayesova vzorce dostaneme pravděpodobnost $P(B|A)$.

Charakteristický znak, který odlišuje úlohy vedoucí na 1. Bayesův vzorec od úloh vedoucích na 2. Bayesův vzorec spočívá v tom, že v prvním případě se ptáme na pravděpodobnost jevu, který je totožný s jednou z hypotéz, kdežto ve druhém případě se ptáme na pravděpodobnost jevu, který je zcela nový a nesouvisí s ostatními jevy v úloze popsanými.

5 Stochasticky nezávislé náhodné jevy

Nechť $(\Omega, \mathbf{A}, P)$ je pravděpodobnostní prostor. Řekneme, že náhodné jevy $A_1, A_2, \ldots, A_n$ jsou *stochasticky nezávislé* vzhledem k pravděpodobnosti P, jestliže platí multiplikativní vztahy:

$$\begin{aligned} \forall i < j: \quad & P(A_i \cap A_j) = P(A_i) \cdot P(A_j) \\ \forall i < j < k: \quad & P(A_i \cap A_j \cap A_k) = P(A_i) \cdot P(A_j) \cdot P(A_k) \\ & \vdots \\ & P(A_1 \cap \ldots \cap A_n) = P(A_1) \cdot \ldots \cdot P(A_n). \end{aligned}$$

Definici lze rozšířit i na spočetnou posloupnost jevů: Řekneme, že jevy $A_1, A_2, \ldots, A_k, \ldots \in \mathbf{A}$ jsou *stochasticky nezávislé* vzhledem k pravděpodobnosti P, jestliže pro všechna přirozená n jsou stochasticky nezávislé náhodné jevy $A_1, A_2, \ldots, A_n$.

Ověřujeme-li stochastickou nezávislost náhodných jevů, musíme prozkoumat platnost *všech* multiplikativních jevů.

Je-li v textu úlohy řečeno, že jevy jsou stochasticky nezávislé a máme-li stanovit pravděpodobnost nastoupení alespoň jednoho z těchto jevů, využijeme de Morganova pravidla a počítáme
$$P(\bigcup_{i=1}^{n} A_i) = P(\overline{\bigcap_{i=1}^{n} \overline{A_i}}) = 1 - P(\bigcap_{i=1}^{n} \overline{A_i}) = 1 - \prod_{i=1}^{n}(1 - P(A_i)).$$

6 Borelovské množiny a náhodné veličiny

Nechť n je přirozené číslo. Množinu $\mathbf{R}^n$ nazýváme *n-rozměrným prostorem* a množinu $\mathbf{R}^\infty = \mathbf{R}^\mathbf{N}$ nazýváme *spočetně rozměrným prostorem.*

Minimální jevové pole na $\mathbf{R}^n$ obsahující třídu všech intervalů $(-\infty, x_1] \times (-\infty, x_2] \times \ldots \times (-\infty, x_n]$ pro $(x_1, x_2, \ldots, x_n) \in \mathbf{R}^n$, nazýváme *n-rozměrným borelovským polem* $\mathbf{B}^{(n)}$ a prvky tohoto pole nazýváme *n-rozměrnými borelovskými množinami.* Podobně pro spočetné borelovské pole $\mathbf{B}^{(\infty)}$.

Mezi borelovské množiny patří zejména prázdná množina, celý konečně rozměrný popř. spočetně rozměrný prostor, všechny jednobodové, konečné resp. spočetné množiny, všechny intervaly, všechny otevřené i uzavřené oblasti a všechna nejvýše spočetná sjednocení takových množin. Kartézský součin borelovských množin je opět borelovská množina, ale vyšší dimenze.

Zobrazení $X : \Omega \to \mathbf{R}$ se nazývá náhodná veličina vzhledem k jevovému poli $\mathbf{A}$, jestliže
$$\forall B \in \mathbf{B} : \{\omega \in \Omega : X(\omega) \in B\} = X^{-1}(B) \in \mathbf{A},$$
tj. úplný vzor každé borelovské množiny je náhodným jevem.

Obraz $X(\omega)$ se nazývá číselná realizace náhodné veličiny X příslušná k možnému výsledku ω. Jestliže nehrozí nebezpečí nedorozumění, zapisujeme náhodnou veličinu i její realizaci týmž symbolem X. Množinu $\{\omega \in \Omega : X(\omega) \in B\}$ zkráceně zapisujeme $\{X \in B\}$.

Víme, že každá množina typu $\{X \in B\}$ (kde $B \in \mathbf{B}^{(n)}$) je jevem. Ve speciálním případě $B = \{x\}$ nebo $B = (-\infty, x]$ píšeme jednodušeji $\{X = x\}$ $\{X \leq x\}$. Podmínky mohou být vyjadřovány logickými spojkami negace, konjunkce, disjunkce a implikace. Jim odpovídají množinové operace komplement, průnik, sjednocení a relace množinové inkluze.

Zápis odpovídající pravděpodobnosti zkrátíme takto: $P(\{\omega \in \Omega : X(\omega) \in B\}) := P(X \in B)$, což je pravděpodobnost, že náhodná veličina X se realizuje v množině B; $P(\{\omega \in \Omega : X(\omega) \in B\} | \{\omega \in \Omega : Y(\omega) \in C\} := P(X \in B | Y \in C)$, což je pravděpodobnost, že náhodná veličina X se realizuje v množině B za podmínky, že náhodná veličina Y se realizuje v množině C.

Funkce $\Phi : \mathbf{R} \to \mathbf{R}$ definovaná vztahem $\forall x \in \mathbf{R} : \Phi(x) = P(X \leq x)$ se nazývá *distribuční funkce náhodné veličiny X vzhledem k pravděpodobnosti P.*

Náhodná veličina X se nazývá *diskrétní*, jestliže existuje funkce $\pi(x)$ nulová v $\mathbf{R}$ s výjimkou nejméně jednoho a nejvýše spočetně mnoha bodů, kde je kladná (vlastnost D1: pro všechna $x \in \mathbf{R}$ je $\pi(x) \geq 0$), je normovaná (vlastnost D2: $\sum_{x=-\infty}^{\infty} \pi(x) = 1$, kde se sčítají jen kladné hodnoty) a platí pro ni $\forall x \in \mathbf{R} : \Phi(x) = \sum_{t \leq x} \pi(t)$. Tato funkce se nazývá *pravděpodobnostní funkce* náhodné veličiny X. Kromě D1, D2 má tyto vlastnosti: $\forall x \in \mathbf{R} : \pi(x) = P(X = x)$, je-li $x_0 \in \mathbf{R}$ libovolný, ale pevně daný bod, pak $\pi(x_0) = \Phi(x_0) - \lim_{x \to x_0^-} \Phi(x)$.

Pro diskrétní náhodnou veličinu je charakteristické, že nabývá pouze konečně nebo nejvýše spočetně mnoha hodnot. Její distribuční funkce má schodovitý charakter.

Má-li funkce $\pi(x)$ vlastnosti D1, D2, pak existuje pravděpodobnostní prostor $(\Omega, \mathbf{A}, P)$ a na něm definovaná diskrétní náhodná veličina X tak, že $\pi(x)$ je její pravděpodobnostní funkce.

7 Náhodné vektory

Nechť $(\Omega, \mathbf{A}, P)$ je pravděpodobnostní prostor, $X_1, X_2, \ldots, X_n$ náhodné veličiny definované na $(\Omega, \mathbf{A}, P)$, $\Phi_1, \Phi_2, \ldots, \Phi_n$ jejich distribuční funkce a jedná-li se o diskrétní náhodné veličiny, pak $\pi_1, \pi_2, \ldots, \pi_n$ buďte jejich pravděpodobnostní funkce.

Náhodný vektor je uspořádaná n-tice $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Jeho distribuční funkci definujeme vztahem:

$$\Phi(x_1, x_2, \ldots, x_n) = P(X_1 \leq x_1 \cap X_2 \leq x_2 \cap \ldots \cap X_n \leq x_n).$$

Náhodný vektor $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ se nazývá diskrétní, jestliže existuje funkce $\pi(x)$ nulová v $\mathbf{R}^n$ s výjimkou nejméně jednoho a nejvýše spočetně mnoha bodů, kde je kladná (vlastnost D1: pro všechna $x \in \mathbf{R}^n$ je $\pi(x) \geq 0$), je normovaná (vlastnost D2: $\sum_{x_1=-\infty}^{\infty} \cdots \sum_{x_n=-\infty}^{\infty} \pi(x_1, \ldots, x_n) = 1$, kde se sčítají jen kladné hodnoty) a platí pro ni $\forall x \in \mathbf{R} : \Phi(x_1, \ldots, x_n) = \sum_{t_1 \leq x_1} \cdots \sum_{t_n \leq x_n} \pi(t_1, \ldots, t_n)$. Tato funkce se nazývá *pravděpodobnostní funkce* diskrétního náhodného vektoru $\mathbf{X}$. Kromě D1, D2 má tyto

vlastnosti: $\forall x \in \mathbf{R}^n : \pi(x) = P(X = x)$, je-li $1 \leq i \leq n$ libovolný index, pak $\sum_{x_1=-\infty}^{\infty} \cdots \sum_{x_{i-1}=-\infty}^{\infty} \sum_{x_{i+1}=-\infty}^{\infty} \cdots \sum_{x_n=-\infty}^{\infty} \pi(x_1, \ldots, x_n) = \pi_i(x_i)$.

Nechť T je neprázdná indexová množina. Řekneme, že náhodné veličiny X_t jsou stochasticky nezávislé, jestliže pro všechny konečné podmnožiny $\{u, \ldots, v\} \subseteq T$ a všechny borelovské podmnožiny $B_u, \ldots, B_v$ jsou stochasticky nezávislé jevy $\{X_u \in B_u\}, \ldots, \{X_v \in B_v\}$ tj. $\pi(x_1, \ldots, x_n) = \pi_1(x_1 \cdot \ldots \cdot \pi_n(x_n)$.

Zobrazení $g : \mathbf{R} \to \mathbf{R}$ nazýváme *borelovskou funkcí*, jestliže vzor borelovské množiny je borelovská množina tj.

$$\forall B \in \mathbf{B}^{(1)} : \{x \in \mathbf{R} : g(x) \in B\} \in \mathbf{B}^{(1)}.$$

Obecně zobrazení $g = (g_1, g_2, \ldots, g_n) : \mathbf{R}^n \to \mathbf{R}^m$ nazýváme *borelovskou funkcí*, jestliže vzor borelovské množiny je borelovská množina tj.

$$\forall B \in \mathbf{B}^{(m)} : \{(x_1, \ldots, x_n) \in \mathbf{R}^n : (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)) \in B\} \in \mathbf{B}^{(n)}.$$

Mezi borelovské funkce náleží zejména všechny funkce spojité na celém $n-$rozměrném prostoru nebo v každé z disjunktních oblastí, na něž byl tento prostor rozložen. Rovněž limita všude konvergentní posloupnosti takovýchto funkcí je borelovská funkce.

Z dané náhodné veličiny $X : \Omega \to \mathbf{R}$ vytvoříme pomocí borelovské funkce $g : \mathbf{R} \to \mathbf{R}$ zobrazení $Y : \Omega \to \mathbf{R}$ dané takto:

$$\forall \omega \in \Omega : Y(\omega) = g(X(\omega)).$$

Toto zobrazení je opět náhodná veličina. Nazývá se *transformovaná náhodná veličina* .

Z daného náhodného vektoru $\mathbf{X} = (X_1, \ldots, X_n) : \Omega \to \mathbf{R}^n$ vytvoříme pomocí borelovské funkce $g = (g_1, \ldots, g_m) : \mathbf{R}^n \to \mathbf{R}^m$ zobrazení $Y : \Omega \to \mathbf{R}^m$ dané takto:

$$\forall \omega \in \Omega : Y(\omega) = g(X(\omega)).$$

Toto zobrazení je opět náhodný vektor. Nazývá se *transformovaný náhodný vektor*. Rozepsáno:

$$\forall \omega \in \Omega : (Y_1(\omega), \ldots, Y_m(\omega)) = (g_1(X_1(\omega), \ldots, X_n(\omega)), \ldots, g_m(X_1(\omega), \ldots, X_n(\omega))).$$

Předpokládejme nyní, že $\mathbf{X}$ je diskrétní náhodný vektor s pravděpodobnostní funkcí $\pi(x_1, \ldots, x_n)$ a $g : \mathbf{R}^n \to \mathbf{R}$ je borelovská funkce. Odvodíme pravděpodobnostní funkci $\pi_*(y)$ transformované náhodné veličiny $Y(\omega) = g(X(\omega))$. Totiž

$$\pi_*(y) = P(Y = y) = P(g(X_1, \ldots, X_n) = y) = P((X_1, \ldots, X_n) \in S(y)),$$

kde $S(y) = \{(x_1, \ldots, x_n) \in \mathbf{R}^n : g(x_1, \ldots, x_n) = y\}$, tedy

$$\pi_*(y) = \sum_{(x_1, \ldots, x_n) \in S(y)} \pi(x_1, \ldots, x_n).$$

8 Střední hodnota, rozptyl, kovariance a koeficient korelace náhodných veličin

Nechť je dán pravděpodobnostní prostor $(\Omega, \mathbf{A}, P)$ a skalární náhodná veličina X. Je-li náhodná veličina X diskrétní a má pravděpodobnostní funkci $\pi(x)$, nazýváme její *střední hodnotou* vzhledem k pravděpodobnosti P číslo $E(X) = \sum_{x=-\infty}^{\infty} x \cdot \pi(x)$ za předpokladu, že případná nekonečná řada vpravo absolutně konverguje. Jinak řekneme, že $E(X)$ neexistuje. Střední hodnota $E(X)$ je číslo, které charakterizuje polohu číselných realizací náhodné veličiny X na číselné ose. Nechť $Y = g(X)$, kde g je borelovská funkce, X diskrétní náhodná veličina. Pak $E(Y) = \sum_{x\in\mathbf{R}} g(x)\pi(x)$, pokud nekonečná řada vpravo absolutně konverguje. Číslo $D(X) = E([X - E(X)]^2)$ nazýváme *rozptylem* náhodné veličiny X vzhledem k pravděpodobnosti P za předpokladu, že obě střední hodnoty vpravo existují. Číslo $\sqrt{D}$ nazýváme *směrodatnou odchylkou* náhodné veličiny X. Pro výpočty je výhodný tvar $D(X) = E(X^2) - [E(X)]^2$. Rozptyl $D(X)$ je číslo, které charakterizuje variabilitu číselných realizací náhodné veličiny X kolem střední hodnoty $E(X)$.

Nechť (X_1, X_2) je náhodný vektor. *Kovariancí* náhodných veličin X_1, X_2 nazýváme číslo $C(X_1, X_2) = E([X_1 - E(X_1)][X_2 - E(X_2)])$ za předpokladu, že střední hodnoty vpravo existují. Kovariance $C(X_1, X_2)$ je číslo, které charakterizuje společnou variabilitu číselných realizací náhodných veličin X_1, X_2 kolem středních hodnot $E(X_1), E(X_2)$. Je-li $C(X_1, X_2) = 0$ řekneme, že náhodné veličiny X_1, X_2 jsou nekorelované, tzn. že mezi nimi neexistuje žádná lineární závislost. Pro výpočty je výhodný tvar $C(X_1, X_2) = E(X_1, X_2) - E(X_1)E(X_2)$.

Koeficientem korelace náhodných veličin X_1, X_2 nazýváme číslo

$$R(X_1, X_2) = E\left(\frac{X_1 - E(X_1)}{\sqrt{D(X_1)}} \cdot \frac{X_2 - E(X_2)}{\sqrt{D(X_2)}}\right) = \frac{C(X_1, X_2)}{\sqrt{D(X_1)}\sqrt{D(X_2)}},$$

$\sqrt{D(X_1)}\sqrt{D(X_2)} \neq 0$ za předpokladu, že střední hodnoty vpravo existují. Koeficient korelace $R(X_1, X_2)$ je číslo, které charakterizuje míru těsnosti lineární závislosti číselných realizací náhodných veličin X_1, X_2. Nabývá hodnot z intervalu $< -1, 1 >$.

9 Cauchy-Schwarz-Buňakovského nerovnost, Markovova a Čebyševova nerovnost

a) *Cauchy-Schwarz-Buňakovského nerovnost*

Nechť X_1, X_2 jsou náhodné veličiny. Jestliže existují jejich střední hodnoty a rozptyly, pak

$$|C(X_1, X_2)| \leq \sqrt{D(X_1)}\sqrt{D(X_2)}, \text{ tj. } |R(X_1, X_2)| \leq 1$$

a rovnost nastane tehdy a jen tehdy, když mezi veličinami X_1, X_2 existuje s pravděpodobností 1 úplná lineární závislost, tedy existují konstanty a_1, a_2 tak, že $P(X_2 = a_1 + a_2X_1) = 1$.

b) *Markovova nerovnost*

Jestliže je $P(X > 0) = 1$ a $E(X)$ existuje, pak pro všechna $\varepsilon > 0$ platí

$$P(X > \varepsilon E(X)) \leq \frac{1}{\varepsilon}.$$

b) *Čebyševova nerovnost*

Jestliže existují $E(X)$ a $D(X)$, pak pro každé $t > 0$ platí:

$$P(|X - E(X)| > t\sqrt{D(X)}) \leq \frac{1}{t^2}.$$

Rejstřík

Literatura

[1] J. Adámek: *Stochastické procesy a teorie informace - úlohy.* ČVUT, Praha 1989.

[2] J. Adámek: *Kódování.* SNTL, Praha 1989.

[3] J. Adámek: *Foundations of Coding Theory.* John Wiley & Sons, New York 1991.

[4] M. Budíková, Š. Mikoláš, P. Osecký: *Teorie pravděpodobnosti a matematická statistika - sbírka příkladů.* Masarykova univerzita, Brno 1996.

[5] V. Klíma: *Kódy, komprimace a šifrování.* Chip 2/1993, str. 24-28.

[6] L. Kučera a J. Nešetřil: *Algebraické metody diskrétní matematiky.* SNTL, Praha 1989.

[7] J. Paseka: *Kryptografie.* Učební texty Masarykova univerzita, Brno 1997.

[8] Š. Porubský a O. Grošek: *Šifrovanie. Algoritmy, Metódy, Prax.* Grada, Praha 1992.

[9] S. Roman: *Introduction to Coding and Information Theory.* Springer-Verlag New York, New York 1997.

[10] D. Welsh: *Codes and Cryptography.* Oxford University Press, New York 1989.

[11] A. D. Wyner: *The wire-tap channel.* Bell Syst. Tech. J. 54, 1975, str. 1355-1387.