Komprese dat

Jan Outrata

KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

přednášky

Statistické metody

Tunstallův kód

= zdrojová slova proměnné délky kódována na kódová slova pevné délky $k \geq \lceil \log_m n \rceil$ kódových symbolů = blokový kód, $\sim$ variable-to-fixed code (n je velikost zdrojové abecedy, m je velikost kódové abecedy)

- chyby v kódových slovech se při dekódování nešíří – robustnost
- požadavky:
  1. každou (neprázdnou) posloupnost zdrojových symbolů musí být možné vyjádřit jako (případně prefix) zřetězení právě jedné posloupnosti zdrojových slov kódovaných na kódové slovo – jednoznačná kódovatelnost
  2. průměrná délka zdrojových slov kódovaných na kódové slovo je maximální = optimální kód $\rightarrow$ delší zdrojová slova mají větší pravděpodobnost výskytu
  3. je použito maximum kódových slov, ideálně všech m^k – optimalita
- prefixový – pro zdrojová slova kódovaná na kódové slovo, jednoznačná kódovatelnost
- průměrná délka kódu: $\frac{k}{\sum_{i=1}^{t} P(w_i) l(w_i)}$, t počet zdrojových slov w_i délky $l(w_i)$ s pravděpodobností výskytu $P(w_i)$
- pouze statický a semi-adaptivní model

Tunstallův kód

- B. P. Tunstall

Input : číslo k
Uses : zdrojová abeceda A, $n = |A|$, pravděpodobnosti $P(A^+)$ výskytu slova z A, velikost m kódové abecedy
Output: $C(U)$
$T \leftarrow A$;
$i \leftarrow 0$;
while $n + (i+1)(n-1) \leq m^k$ **do**
    $x \leftarrow w \in T, P(w) \geq P(w'), w' \in T$;
    $T \leftarrow (T \setminus \{x\}) \cup \{xy \,|\, y \in A\}$;
    $i \leftarrow i + 1$;
$C(T) \leftarrow \{$kódová slova délky $k\}$ libovolně;

PRIKLAD: $k = 4$, $A = \{a, b, r, u, o\}$, $p(a) = \frac{7}{20}$, $p(b) = \frac{5}{20}$, $p(r) = \frac{5}{20}$, $p(u) = \frac{2}{20}$, $p(o) = \frac{1}{20}$, $m = 2$, vstup $barbaraabarboraubaru$, redundance

Shannon-Fanovo kódování

- C. E. Shannon, R. M. Fano
- první pokus o optimální binární prefixový kód – využití distribuční funkce/kumulované pravděpodobnosti (cumulative distribution function) zdroje

Input : čísla a, b
Uses : zdrojová abeceda $A = \{a_1, \ldots, a_n\}, n \geq 2$, pravděpodobnosti $\{p_1, \ldots, p_n\}$, $p_i \geq p_j$ pro $i < j$, výskytu a_i
Output : kód $C(A)$
if $a + 1 = b$ **then**
    $C(a_a) \leftarrow \mathbf{0}$;
    $C(a_b) \leftarrow \mathbf{I}$;
najdi j takové, že $|\sum_{i=a}^{j} p_i - \sum_{i=j+1}^{b} p_i|$ je minimální;
$C(A) \leftarrow$ zavolej se rekurzivně s a, j a s $j + 1, b$;
$C(a_i) \leftarrow \mathbf{0}C(a_i)$ pro $i = a, \ldots, j$;
$C(a_i) \leftarrow \mathbf{I}C(a_i)$ pro $i = j + 1, \ldots, b$;
Run with: $1, n$

PRIKLAD: $A = \{a, b, r, u, o\}$, $p(a) = \frac{7}{20}$, $p(b) = \frac{5}{20}$, $p(r) = \frac{5}{20}$, $p(u) = \frac{2}{20}$, $p(o) = \frac{1}{20}$, vstup $barbaraabarboraubaru$, redundance

- optimální při $|\sum_k p_k = \sum_{i=a}^{j} p_i - \sum_{i=j+1}^{b} p_i = \sum_l p_l|$, $k, l \in \{a, \ldots, b\}, \{k\} \cap \{l\} = \emptyset$, minimální

Huffmanovo kódování

- David A. Huffman: A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E.*, pp. 1098–1102, 1952.
- optimální prefixové, vyplývá z vlastností optimálního prefixového kódu (viz Věta dříve) a následujícího Lemma

Huffmanovo kódování

- David A. Huffman: A method for the construction of minimum-redundancy codes. *Proceedings of the I.R.E.*, pp. 1098–1102, 1952.
- optimální prefixové, vyplývá z vlastností optimálního prefixového kódu (viz Věta dříve) a následujícího Lemma

Lemma

Nechť $C' : A' \mapsto B^+$, *kde* $A' = \{a'_1, \ldots, a'_{n'}\}, n' \geq m \geq 2, a'_i = a_i \in A, i < n'$, *pravděpodobnosti výskytu symbolů* a'_i *jsou* $p'_i = p_i, i < n', p'_{n'} = \sum_{j=n-m'+1}^{n} p_j$, $A = \{a_1, \ldots, a_n\}$, $n = n' + m' - 1$, *s pravděpodobnostmi výskytu* p_i *symbolů* a_i, *kde* $p_{n-m'+1}, \ldots, p_n$ *jsou nejmenší,* $m' \in \{2, 3, \ldots, m\}, m' \equiv n \pmod{(m-1)}$, $B = \{b_1, \ldots, b_m\}$, *je optimální prefixový kód ze zdrojové abecedy* A' *do kódové abecedy* B.
Pak kód $C : A \mapsto B^+$, $C(a_i) = C'(a'_i), i < n', C(a_{n-m'+j}) = C'(a'_{n'})b_j, j \leq m'$, *ze zdrojové abecedy* A *do kódové abecedy* B *je také optimální.* □

Huffmanovo kódování

Statický a semi-adaptivní model

Input : abeceda $A' = \{a'_1, \ldots, a'_{n'}\}$, pravděpodobnosti $\{p'_1, \ldots, p'_{n'}\}$ výskytu a'_i
Uses : zdrojová abeceda $A = \{a_1, \ldots, a_n\}$, pravděpodobnosti $\{p_1, \ldots, p_n\}$ výskytu a_i, kódová abeceda $B = \{b_1, \ldots, b_m\}$
Output : kód $C(A')$

setřiď a'_i a p'_i tak, že $p'_i \geq p'_j$ pro $i < j$;
if $n' \leq m$ **then**
    $C(a'_i) \leftarrow b_i$ pro $i = 1, \ldots, n'$;
else
    if $n' = n$ **then**
        $m' \leftarrow (n-2) \mod (m-1) + 2$;
    else
        $m' = m$
    $C(A') \leftarrow$ zavolej se rekurzivně s $A' \setminus \{a'_{n'-m'+2}, \ldots, a_{n'}\}, \{p'_1, \ldots, p'_{n'-m'}, \sum_{i=n'-m'+1}^{n'} p'_i\}$;
    $C(a'_{n'-m'+i}) \leftarrow C(a'_{n'-m'+1})b_i$ pro $i = 1, \ldots, m'$;
Run with: $A = \{a_1, \ldots, a_n\}, \{p_1, \ldots, p_n\}$

PRIKLAD: $A = \{a, b, r, u, o\}$, $p(a) = \frac{7}{20}$, $p(b) = \frac{5}{20}$, $p(r) = \frac{5}{20}$, $p(u) = \frac{2}{20}$, $p(o) = \frac{1}{20}$, $m = 2$, vstup $barbaraabarboraubaru$, $m = 3$, vstup ??, redundance

Huffmanovo kódování

Proč m' a ne m? Protože při tomto počtu nejdelších kódových slov bude u všech kratších kódových slov a prefixů využito všech m symbolů kódové abecedy a kód má být optimální prefixový.

Huffmanův strom $T_H(A) = \langle V_H(A), E_H(V_H(A)) \rangle$ = reprezentace Huffmanova kódu $C(A)$ formou m-árního stromu:

- listové uzly $v_l(a_i) \in V_H(A)$ pro symboly $a_i \in A, i = 1, \ldots, n$
- vnitřní uzly $v(a'_{n'}) \in V_H(A)$ pro všechny $a'_{n'}$ (ve všech rekurzivních voláních) + kořenový uzel $v_r \in V_H(A)$
- hrany $\langle v(a'_{n'}), v(a'_{n'-m'+i}) \rangle_{b_i} \in E_H(V_H(A))$ a $\langle v_r, v(a'_i) \rangle_{b_i} \in E_H(V_H(A))$ označené $b_i \in B$ z uzlu pro $a'_{n'}$ následujícího rekurzivního volání do uzlů pro $a'_{n'-m'+i}, i = 1, \ldots, m'$ při $n' > m$ a z kořenového uzlu do uzlů pro $a'_i, i = 1, \ldots, n'$ při $n' \leq m$
- $C(a) =$ zřetězení $b \in B$ označujících hrany na cestě stromem z v_r do $v_l(a)$

PRIKLAD

Huffmanovo kódování

- minimální rozdíly v délkách kódových slov (pro jejich minimální bufferování při pevné rychlosti přenosu/ukládání): třídění a'_i a p'_i tak, že původní $a'_{n'}$ bude po zatřídění $a'_i, i < j$ pro všechna j, pro která $p'_j = p'_i$
- $m = 2$ a $p_i > \sum_{j=i+2}^{n} p_j$, $p_i \geq p_j$ pro $i < j$ (speciálně $p_i = 2^{-i}, i = 1, \ldots, n-1$ a $p_n = 2^{-(n-1)}$) $\rightarrow$ unární číselný kód $i - 1$ pro $a_i, i = 1, \ldots, n-1$ a n $\mathbf{I}$ pro a_n
- $m = 2$ a $p_1 < p_{n-1} + p_n$, $p_i \geq p_j$ pro $i < j$ (speciálně $p_i = \frac{1}{n}$) $\rightarrow$ blokový kód délky $k = \lfloor \log_2 n \rfloor$ pro $a_1, \ldots, a_{2^k}$ a délky $k + 1$ pro $a_{2^k+1}, \ldots, a_n$

PRIKLADY

Huffmanovo kódování

Adaptivní model – binární kód

- triviálně: znovuvytváření kódu pro každý další symbol na vstupu – výpočetně náročné
- Faller, Gallager, Knuth, Vitter
- vlastnost Huffmanova stromu (tzv. sibling property): $p'_{n'} \leq \ldots \leq p'_{n'-m'+1} \leq p'_{n'} \leq \ldots \leq p'_{n'-m'+1}$ následujícího rekurzivního volání – musí stále platit $\rightarrow$ v následujících algoritmech zajištěno pomocí (aktuálního) počtu $n(x)$ výskytů symbolu x a čísla $i(v(x)), v(x) \in V_H(A)$
- speciální (escape) symbol e značící neexistující/první výskyt symbolu

$T_H(A) \leftarrow \langle \{v_l(e)\}, \emptyset \rangle$, $C(e) \leftarrow$ prázdný řetězec;
$n(e) \leftarrow 0$;
$i(v_l(e)) \leftarrow 1$;
while načti ze vstupu symbol $a \in A$ **do**
    if $v_l(a) \notin V_H(A)$ **then**
        zapiš na výstup $C(e)$ a kód a;
    else
        zapiš na výstup $C(a)$;
    zavolej následující algoritmus;

Huffmanovo kódování

Adaptivní model – binární kód

if $v_l(a) \notin V_H(A)$ **then**
$V_H(A) \leftarrow V_H(A) \cup \{v_l(a), v(x)\}$;
$n(a) \leftarrow n(x) \leftarrow 1$;
$i(v(x)) \leftarrow i(v_l(e))$; $i(v_l(a)) \leftarrow i(v(x)) + 1$; $i(v_l(e)) \leftarrow i(v_l(a)) + 1$;
$E_H(V_H(A)) \leftarrow (E_H(V_H(A)) \setminus \{\langle u, v_l(e)\rangle_b\}) \cup \{\langle v(x), v_l(e)\rangle_{\mathbf{1}}, \langle v(x), v_l(a)\rangle_{\mathbf{0}}, \langle u, v(x)\rangle_b\}$;
else
$v(x) \leftarrow v_l(a)$;
while $v(x) \neq v_r$ **do**
if $\langle v(x), v_l(e)\rangle \notin E_H(V_H(A))$ **then**
najdi $v(y)$ takové, že $i(v(y)) < i(v(z)), \forall v(z) \in V_H(A), n(y) = n(z)$;
if $v(y) \neq v(x) \wedge \langle v(y), v(x)\rangle \notin E_H(V_H(A))$ **then**
$v \leftarrow v(x)$; $v(x) \leftarrow v(y)$; $v(y) \leftarrow v$;
$i \leftarrow i(v(x))$; $i(v(x)) \leftarrow i(v(y))$; $i(v(y)) \leftarrow i$;
$n(x) \leftarrow n(x) + 1$;
$v(x) \leftarrow u, \langle u, v(x)\rangle \in E_H(V_H(A))$;
$n(x) \leftarrow n(x) + 1$;

PRIKLAD

Huffmanovo kódování

Adaptivní model – binární kód

$T_H(A) \leftarrow \langle \{v_l(e)\}, \emptyset \rangle$;
$n(e) \leftarrow 0$;
$i(v_l(e)) \leftarrow 1$;
while není konec vstupu **do**
    $v \leftarrow v_r$;
    while $v \neq v_l(a)$ **do**
        načti ze vstupu symbol $b \in B$;
        $v \leftarrow u, \langle v, u \rangle_b \in E_H(V_H(A))$;
    if $a = e$ **then**
        načti ze vstupu kód symbolu $a \in A$;
        dekóduj kód a a zapiš na výstup a;
    else
        zapiš na výstup symbol $a \in A$;
    zavolej předchozí algoritmus;

PRIKLAD

Huffmanovo kódování

Aplikace

- často v návaznosti na jiné metody, např. na diferenční kódování (obraz, zvuk)

Aritmetické kódování

- průměrná délka optimálního prefixového kódu, např. Huffmanova, je minimálně rovna entropii zdroje a nejvýše o 1 větší než entropie (Shannon noisless coding theorem, viz Věta dříve) – platí těsnější, nejvýše o nejvyšší pravděpodobnost $p_{max} \geq 0.5$ zdrojových symbolů nebo o $p_{max} + 0.086, p_{max} < 0.5$
- změna zdrojové abecedy na k-tice (nezávislých) symbolů z původní abecedy A pro přiblížení se entropii ale zvyšuje velikost abecedy, a tím i Huffmanova stromu, na $|A|^k$, např. pro $p_1 = 0.95, p_2 = 0.03, p_3 = 0.02$ je entropie přibližně 0.335 b/symbol, průměrná délka Huffmanova kódu 1.05 b/symbol, kódu pro 9 dvojic symbolů přibližně 0.611 b/symbol a kódu pro $?$?tic symbolů přibližně $?$ b/symbol!

$\Rightarrow$ výhodnější kódovat zdrojová slova než samostatné symboly – ale nevytvářet kód pro všechna slova dané délky, např. Huffmanův!

$\rightarrow$ kód pouze pro zdrojová slova na vstupu

- vhodné pro malé zdrojové abecedy, např. binární, s velkými rozdíly v pravděpodobnostech symbolů

$=$ kódování zdrojových slov do čísel z podintervalů $[0, 1)$ kódovaných do binárního kódu

Aritmetické kódování

- Pasco, Rissanen, 1976, Rissanen, Langdon, 1979
- využití distribuční funkce/kumulované pravděpodobnosti (cumulative distribution function) $F_X(i) = \sum_{k=1}^{i} P(X = k), P(X = k) = P(a_k) = p_k$ zdroje (nezávisle se stejným pravděpodobnostním rozložením se vyskytujících) symbolů z abecedy $A = \{a_1, a_2, \ldots, a_n\}$ jako náhodných proměnných $X(a_i) = i$, $F_X(0) = 0$

$l_p \leftarrow 0$; $u_p \leftarrow 1$;
while načti ze vstupu symbol $a_i \in A$ **do**
    $l \leftarrow l_p + (u_p - l_p)F_X(i-1)$;
    $u \leftarrow l_p + (u_p - l_p)F_X(i)$;
    $l_p \leftarrow l$; $u_p \leftarrow u$;
    `// přeškálování` $[l, u)$
zapiš na výstup C = binární reprezentace jakéhokoliv čísla z $[l, u)$ s minimem bitů;
PRIKLAD

Aritmetické kódování

- $C(A^+)$ je prefixový binární kód ze zdrojových slov nad abecedou A, průměrná délka pro slova délky k je $H(A^k) \leq \bar{l}(C(A^k)) < H(A^k) + 1 \Rightarrow$ průměrná délka na symbol z A je $H(A) \leq \bar{l} < H(A) + \frac{1}{k}$
- pro dekódování je nutné znát délku L kódovaného zdrojového slova $\rightarrow$ uložit spolu s komprimovanými daty nebo speciální zdrojový symbol značící konec vstupu

$l_p \leftarrow 0$; $u_p \leftarrow 1$;
$j \leftarrow 0$;
načti ze vstupu binární reprezentaci čísla $x \in [0, 1)$;
while $j < L$ **do**
    najdi $i \in \{1, \ldots, n\}$ takové, že $F_X(i-1) \leq \frac{x - l_p}{u_p - l_p} < F_X(i)$;
    zapiš na výstup symbol $a_i \in A$;
    $j \leftarrow j + 1$;
    if $j < L$ **then**
        $l \leftarrow l_p + (u_p - l_p) F_X(i-1)$;
        $u \leftarrow l_p + (u_p - l_p) F_X(i)$;
        $l_p \leftarrow l$; $u_p \leftarrow u$;
        `// přeškálování` $[l, u)$

PRIKLAD

Aritmetické kódování

- u kódování i dekódování žádoucí průběžný výstup během čtení vstupu, ne až po načtení celého vstupu $\rightarrow$ kód čísla z $[l, u)$ průběžně
- $[l, u)$ se s délkou zdrojového slova zmenšuje, ale uložení necelých čísel je v praxi s omezenou přesností $\rightarrow$ omezení délky slova nebo přeškálování $[l, u)$:
  1. $u < 0.5$: $x \leftarrow 2x, x \in \{l, u\}$
  2. $l \geq 0.5$: $x \leftarrow 2(x - 0.5), x \in \{l, u\}$
  3. $l \geq 0.25 \wedge u < 0.75$: $x \leftarrow 2(x - 0.25), x \in \{l, u\}$
  - případy $\overbrace{3\ldots31}^{c\text{-krát}} = 1\overbrace{2\ldots2}^{c\text{-krát}}$ a $\overbrace{3\ldots32}^{c\text{-krát}} = 2\overbrace{1\ldots1}^{c\text{-krát}}$

Aritmetické kódování

$c \leftarrow 0$;
// while ...
while $u < 0.5 \vee l \geq 0.5 \vee (l \geq 0.25 \wedge u < 0.75)$ **do**
    if $l \geq 0.25 \wedge u < 0.75$ **then**
        // případ 3
        $c \leftarrow c + 1$;
        $d \leftarrow 0.25$;
    else
        if $u < 0.5$ **then**
            // případ 1
            $b \leftarrow \mathbf{0}$;
            $d \leftarrow 0$;
        else
            // případ 2
            $b \leftarrow \mathbf{I}$;
            $d \leftarrow 0.5$;
        zapiš na výstup b;
        while $c > 0$ **do**
            zapiš na výstup inverzi b;
            $c \leftarrow c - 1$;
    $l \leftarrow 2(l - d)$;
    $u \leftarrow 2(u - d)$;

zapiš na výstup $C = \mathbf{I}$;

PŘÍKLAD

Aritmetické kódování

načti ze vstupu $\lceil -\log_2 p_{min} \rceil$ bitů binární reprezentace čísla $x \in [0, 1)$, p_{min} nejnižší pravděpodobnost zdrojových symbolů;

```
// while ...
```

while $u < 0.5 \vee l \geq 0.5 \vee (l \geq 0.25 \wedge u < 0.75)$ **do**

  if $l \geq 0.25 \wedge u < 0.75$ **then**

    $d \leftarrow 0.25$;

  else

    if $u < 0.5$ **then**

      $d \leftarrow 0$;

    else

      $d \leftarrow 0.5$;

  $l \leftarrow 2(l - d)$;

  $u \leftarrow 2(u - d)$;

  načti ze vstupu další bit b anebo $b \leftarrow \mathbf{0}$;

  $x \leftarrow 2(x - d) + b\frac{1}{2^{\lceil -\log_2 p_{min} \rceil}}$;

PRIKLAD

Aritmetické kódování

Celočíselná implementace

- zobrazení $[0,1)$ na $[0,M-1)$ ($0.25 \mapsto \frac{M}{4}$, $0.5 \mapsto \frac{M}{2}$, $0.75 \mapsto \frac{3M}{4}$), $M \geq 4\frac{1}{p_{min}}$, M typicky 2^8, 2^{16}, 2^{32} nebo 2^{64} podle datového typu pro čísla z $[0,M-1)$
- odhad $F_X(i)$ s frekvencemi/četnostmi $f(a_k) = \frac{n(a_k)}{\sum_{j=1}^{|A|} n(a_j)}$ výskytu symbolu $a_k \in A$ místo pravděpodobností $P(a_k)$
- $l \leftarrow l_p + \lfloor (u_p - l_p + 1)F_X(i-1) \rfloor$, $u \leftarrow l_p + \lfloor (u_p - l_p + 1)F_X(i) \rfloor - 1$, $\frac{x-l_p+1}{u_p-l_p+1}$, $u \leftarrow 2(u-d)+1$, $x \leftarrow 2(x-d)+b$ – kvůli celočíselné aritmetice
- načti ze vstupu $\lceil \log_2 M \rceil$ bitů binární reprezentace čísla $x \in [0,M-1)$
- při $M = 2^k$ pro nějaké $k \geq 2$:
  - $u < \frac{M}{2} \rightarrow$ nejvýznamnější bit l i u je $\mathbf{0}$
  - $l \geq \frac{M}{2} \rightarrow$ nejvýznamnější bit l i u je $\mathbf{I}$
  - $l \geq \frac{M}{4} \wedge u < \frac{3M}{4} \rightarrow$ druhý nejvýznamnější bit l je $\mathbf{I}$ a u je $\mathbf{0}$
  - $2x$ a $2(x-\frac{M}{2}) \rightarrow$ bitový posun x doleva o 1 bit, $2(x-\frac{M}{4}) \rightarrow$ navíc inverze (nového) nejvýznamnějšího bitu

PRIKLAD

Aritmetické kódování

Adaptivní model

- průběžný odhad $F_X(i)$ s frekvencemi/četnostmi $f(a_k) = \frac{n(a_k)}{\sum_{j=1}^{|A|} n(a_j)}$ výskytu symbolu $a_k \in A$ místo pravděpodobností $P(a_k)$
  - inicializace na známý odhad nebo počet $n(a) = 1$ výskytu každého symbolu $a \in A$
  - inkrementace $n(a)$ po (de)kódování symbolu a
- uložení p_{min} spolu s komprimovanými daty, u celočíselné implementace nesmí p_{min} klesnout pod $4\frac{1}{M} \rightarrow$ v praxi $n(a_j) \leftarrow \frac{n(a_j)}{2}$ pro $n(a_j) > 1$ při $\sum_{j=1}^{|A|} n(a_j) = \frac{M}{4}$

Aplikace

- ve standardech komprese multimediálních dat (obraz, video, zvuk)

Aritmetické kódování – QM kódování

= modifikované adaptivní binární aritmetické kódování (Q kódování, skew kódování), tj. pro binární zdrojovou abecedu s adaptivním modelem

- $A = u - l$ místo u: $l \leftarrow l_p + A_p F_X(i-1)$ a $A \leftarrow A_p p_i$
- místo 2 symbolů A, $a_1 = \mathbf{0}$ a $a_2 = \mathbf{I}$, $a_1 =$ více (MPS) a $a_2 =$ méně (LPS) pravděpodobný symbol s průběžně odhadovanými frekvencemi/četnostmi $1-q$ a q výskytu
  - $l \leftarrow l_p$ a $A \leftarrow A_p(1-q)$ pro MPS
  - $l \leftarrow l_p + A_p(1-q)$ a $A \leftarrow A_p q$ pro LPS
  - $\frac{x - l_p}{A_p} < 1 - q$ pro MPS a $\geq 1 - q$ pro LPS
- potlačení násobení (i pro nebinární zdrojové abecedy) – udržování hodnoty A v $[0.75, 1.5)$ a zanedbání násobení A_p
  - $l \leftarrow l_p$ a $A \leftarrow A_p - q$ pro MPS
  - $l \leftarrow l_p + A_p - q$ a $A \leftarrow q$ pro LPS
  - $x - l_p < 1 - q$ pro MPS a $\geq 1 - q$ pro LPS
  - přeškálování l, A: $A < 0.75$: $A \leftarrow 2A$, $l \leftarrow 2l$ pro $l < 0.5$ a $l \leftarrow 2(l - 0.5)$ pro $l \geq 0.5$

Aritmetické kódování – QM kódování

while $A < 0.75$ **do**
    if $l < 0.5$ **then**
        $b \leftarrow \mathbf{0}$;
        $d \leftarrow 0$;
    else
        $b \leftarrow \mathbf{I}$;
        $d \leftarrow 0.5$;
    zapiš na výstup b;
    $l \leftarrow 2(l - d)$;
    $A \leftarrow 2A$;

zapiš na výstup $C = \mathbf{I}$;

načti ze vstupu $\lceil -\log_2 q \rceil$ bitů binární reprezentace čísla $x \in [0, 1)$;
`// while ...`
while $A < 0.75$ **do**
    if $l < 0.5$ **then**
        $d \leftarrow 0$;
    else
        $d \leftarrow 0.5$;
    $l \leftarrow 2(l - d)$;
    $A \leftarrow 2A$;
    načti ze vstupu další bit b anebo $b \leftarrow \mathbf{0}$;
    $x \leftarrow 2(x - d) + b\frac{1}{2^{\lceil -\log_2 q \rceil}}$;

Aritmetické kódování – QM kódování

- inicializace $q = 0.5$ a aktualizace q podle tabulky hodnot při přeškálování, ne po (de)kódování každého symbolu
- při častějším výskytu LPS než MPS, $q > A - q$, prohození symbolů včetně aktuálních hodnot l a A, při přeškálování
- celočíselná implementace: zobrazení $[0, 1.5)$ na $[0, M-1)$ ($0.25 \mapsto \frac{M}{6}$, $0.5 \mapsto \frac{M}{3}$, $0.75 \mapsto \frac{M}{2}, 1 \mapsto \frac{2M}{3}$), $M \geq \frac{4}{3}\frac{1}{q}$, i pro hodnoty q
- použití ve standardu JPEG (JBIG) komprese obrazu