Cvičení - komprese dat II

Michal Kupsa, Štěpán Holub

1. Připomeňme definici frekvenčního kódu:

$$\mathfrak{f}(u) = \gamma_1(|u|_1)\gamma_1(|u|_2)\dots\gamma_1(|u|_m)\gamma_2(\ell(u)),$$

   a uvažujme abecedy $A = B = \{0, 1\}$ a obvyklé lexikografické uspořádání slov. Dekódujte

$$\mathfrak{f}(u) = 111010010110001\,.$$

2. Připomeňme definici LZ kódu:

$$\mathfrak{r}(u) = \gamma_2(a_1)f(c_1)\gamma_2(a_2)f(c_2)\dots\gamma_2(a_{C(u)})f(c_{C(u)}).$$

   a uvažujme f jako identitu. Dekódujte

$$\mathfrak{r}(u) = 000110110101111010010101100110100111100100$$
$$100101100010100011100000100001110111110110111111$$

3. Nechť $L(n)$ je délka LZ rozboru prefixu délky n nekonečného slova $u \in \{0, 1\}^\omega$. Najděte u takové, aby $L(n)$ rostlo

   a) co nejrychleji,

   b) co nejpomaleji.

4. Uvažujme LZ kompresi slova $u = (01)^m$.

   a) Jak taková komprese vypadá?

   b) Srovnejte ji s kompresí danou výrazem v zadání.

   c) Vymyslete modifikaci LZ komprese, která bude využívat fakt, že prefix u délky $2(m-1)$ se znovu opakuje na pozici 3.

Řešení a komentář

1. Kód má v našem případě formu $\gamma_1(|u|_0)\gamma_1(|u|_1)\gamma_2(\ell(u))$. Máme

$$\gamma_1(|u|_0) = \overbrace{1110}^{3}\overbrace{100}^{10}, \quad \gamma_1(|u|_1) = \overbrace{10}^{1}\overbrace{1}^{1}, \quad \gamma_2(\ell(u)) = \overbrace{10}^{1}\overbrace{0}^{2}\overbrace{01}^{5}.$$

Slovo u tedy obsahuje deset nul a jednu jedničku a mezi slovy s tímto Parikhovým vektorem má index 5. Množinu $T((10,1))$ číslujeme od nuly, hledané slovo u je tedy šesé v lexikografickém seznamu této množiny, tedy $u = \texttt{0000100000}$.

2. Máme

$$\begin{aligned}
\gamma_2(a_1) = \gamma_2(a_2) &= \overbrace{0}^{0}, && f(c_1) = 0, && f(c_2) = 1,\\
\gamma_2(a_3) = \gamma_2(a_4) &= \overbrace{10}^{1}\overbrace{1}^{1}\overbrace{1}^{1}, && f(c_3) = 0, && f(c_4) = 1,\\
\gamma_2(a_5) = \gamma_2(a_6) &= \overbrace{10}^{1}\overbrace{1}^{1}\overbrace{0}^{2}, && f(c_5) = 0, && f(c_6) = 1,\\
\gamma_2(a_7) = \gamma_2(a_8) &= \overbrace{10}^{1}\overbrace{0}^{2}\overbrace{11}^{3}, && f(c_7) = 0, && f(c_8) = 1,\\
\gamma_2(a_9) = \gamma_2(a_{10}) &= \overbrace{10}^{1}\overbrace{0}^{2}\overbrace{10}^{4}, && f(c_9) = 0, && f(c_{10}) = 1,\\
\gamma_2(a_{11}) = \gamma_2(a_{12}) &= \overbrace{10}^{1}\overbrace{0}^{2}\overbrace{01}^{5}, && f(c_{11}) = 0, && f(c_{12}) = 1,\\
\gamma_2(a_{13}) = \gamma_2(a_{14}) &= \overbrace{10}^{1}\overbrace{0}^{2}\overbrace{00}^{6}, && f(c_{13}) = 0, && f(c_{14}) = 1\\
\gamma_2(a_{15}) = \gamma_2(a_{16}) &= \overbrace{110}^{2}\overbrace{11}^{3}\overbrace{111}^{7}, && f(c_{13}) = 0, && f(c_{14}) = 1\,.
\end{aligned}$$

Rozbor kódovaného slova u tedy je

$$R(u) = \lambda, \texttt{0}, \texttt{1}, \texttt{00}, \texttt{01}, \texttt{10}, \texttt{11}, \texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111}, \texttt{0000}, \texttt{0001}$$

a kódujeme posloupnost dvojic (a_i, c_i)

$$(0,0)(0,1)(1,0)(1,1)(2,0)(2,1)(3,0)(3,1)(4,0)(4,1)(5,0)(5,1)(6,0)(6,1)(7,0)(7,1)\,.$$

Jedná se o začátek tzvn. Champernowneovy (binární)posloupnosti, která je seznamem všech binárních slov seřazených podle maximo-lexikografického uspořádání (nejprve řadíme podle délky, poté lexikograficky).

K bodům 3. a 4. nejprve obecný komentář. Tyto úlohy se zabývají vlastnostmi komprese **jednoho konkrétního slova**. To nemá žádnou bezprostřední souvislost s teorií univerzálních kódů, protože ta se zabývá kompresí **náhodných procesů**. Pro jedno slovo (jednu zprávu, jeden komprimovaný soubor ...) nedává dobrý smysl se ptát, zda se jedná o optimální kompresi:

- optimalita komprese je definována pro entropii příslušného procesu;
- pevně zvolené slovo lze vždy „optimálně" zakódovat např. nulou. Kód nula pak prostě znamená: „to speciální slovo, které jsme zvolili".

S tímto předběžným varováním můžeme nicméně zkoumat vlastnosti LZ komprese i na jednotlivých slovech.

3. Poznamenejme nejprve, že rychlost růstu rozboru vyjadřuje účinnost komprese: rychlý růst znamená malou kompresi a naopak. Kód je totiž právě zakódovaný rozbor. Krátký rozbor znamená, že se ve slově opakují dlouhé úseky, které umožňují efektivní odkazy typy „to už jsme viděli", které jsou podstatou LZ komprese.

   a) Délka rozboru roste nejpomaleji pro posloupnost 0^ω (nebo 1^ω). Slovo délky $n(n+1)/2$ má rozbor délky $n+1$. Takovou posloupnost můžeme popsat náhodným procesem, kde pravděpodobnost nuly je jedna. Pro takový proces již dává smysl ověřit, zda je LZ komprese optimální. Protože proces má nulovou entropii, musí být limita (průměrné) délky kódu prefixu délky n vydělené n nulová. Slovo „průměrné" je v závorce, protože proces je deterministický a slovo dané délky je jen jedno. Ověřujeme tedy, zda
   $$\lim_{n\to\infty} \frac{|\mathfrak{f}(0^n)|}{n} = 0\,,$$
   což jistě platí, protože délka kódu je shora odhadnuta nějakým násobkem $\log n \cdot \sqrt{n}$ (kde $\log n$ zastupuje délku kódu jednoho slova rozboru).

   b) Nejrychleji naopak roste právě rozbor Champernowneovy posloupnosti. Podslovo délky d se totiž zopakuje až poté, co jsou vyčerpána všechna ostatní slova délky d. Champernowneova posloupnost je tedy pro LZ kompresi nejhorší možností. Lze ukázat, že délka rozkladu prefixu této posloupnosti délky n se asymptoticky blíží $n/\log n$. Protože délka kódu se asymptoticky blíží $C(u)\log C(u)$, vidíme, že kompresní

   Lze i tuto situaci popsat s ohledem na optimalitu komprese, tedy s ohledem na nějaký náhodný proces? V tomto případě je situace výrazně méně jasná. Champernowneova posloupnost je možným výstupem jakéhokoli i.i.d. procesu, jehož nosič obsahuje obě písmena. To ilustruje přídavek „skoro jistě" ve větě o optimalitě komprese. Proces s malou entropií musí být podle této věty dobře komprimovatelný pomocí LZ. To platí pro

všechna slova až na množinu míry nula. Champernowneova posloupnost je kandidátem na prvek takové nekomprimovatelné množiny míry nula.

Nejhůře komprimovatelný **proces** je i.i.d. proces s uniformním rozdělením. Ten má entropii jedna a je tedy nekomprimovatelný. Délka kódu takového procesu musí být (opět skoro jistě) v limitě rovna délce komprimovaného slova. Komprese je tedy zcela neúčinná. Kód bude naopak typicky zprávu prodlužovat o „transakční náklady“, jejichž význam bude ovšem asymptoticky zanedbatelný. Uniformně náhodný proces tedy produkuje skoro jistě nekomprimovatelné posloupnosti, což jsou ty, u kterých (v limitě) odpovídají relativní četnosti podslov jejich pravděpodobnosti. Takovým posloupnostem se říká *normální*. Champernowneova posloupnost je jednou z nich. Z výše uvedeného rozkladu je vidět, že komprese je pro ni asymptoticky stejně dobrá jako nekomprese (opět uvažme, že jedna dvojice má kód délky zhruba $\log n$).

4. a) Pro tento případ platí podobná (byť o hodně komplikovanější) analýza jako pro posloupnost 0^ω výše. Všimněme si zejména, že slovo $(01)^\omega$ obsahuje jen dvě slova libovolné délky, délka rozboru je tedy v porovnání s 0^ω zhruba poloviční. Poznamenejme také, že tato posloupnost je produkována markovským procesem se dvěma stavy a deterministickou přechodovou funkcí. Proces tedy má entropii nula.

   b) Tato otázka poukazuje na to, že posloupnost má velmi efektivní zápis (kompresi) sestávající z čísla m (jehož zápis má délku $\log n$). Je ovšem opět třeba připomenout, že existence **krátkého zápisu** nesouvisí přímo s otázkou komprese procesu. Krátký zápis má i Champernowneova posloupnost (totiž její slovní popis, který jsme uvedli, resp. algoritmus, který ji generuje). V případě periodických posloupností ovšem krátkost popisu odpovídá určitému deterministickému markovský procesu, který má nulovou entropii. Kompresní poměr proto asymptoticky klesá k nule i pro LZ kompresi.

   c) LZ komprese, kterou používáme, se přesněji nazývá LZ78 (podle roku publikace). Varianta, na kterou poukazuje tato otázka, se nazývá LZ77. Tato varianta umožňuje odkazovat na posloupnosti, které se s právě zkoumanou pozicí překrývají. To je účinnější právě pro periodické úseky. Tato varianta používá odkaz na místo předchozího výskytu právě zpracovávaného úseku a jeho délky. Jinak řečeno (i, d) znamená: zopakuj úsek délky d začínající na pozici i. (První výskyt všech písmen je přitom třeba zakódovat jinak). Například rozbor slova 0101010 je tedy podle LZ77 roven $\lambda, 0, 1, 01010$, přičemž poslední člen rozboru je reprezentován dvojicí $(1, 5)$. (Celé slovo lze kódovat např. jako $(0, 0), (0, 1), (1, 5)$.)