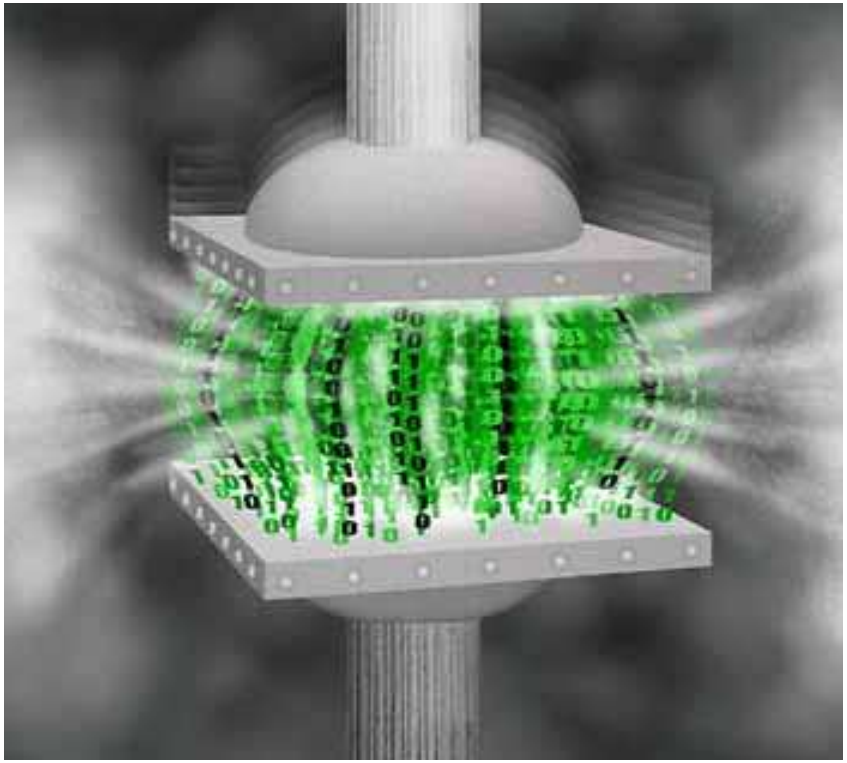
Kompresní datové metody

doc. Ing. Josef Chaloupka, Ph.D.



Komprese dat

- **Bezztrátová komprese** >>> datový řetězec je po dekompresi stejný jako původní
- **Ztrátová komprese**
- Kompresní poměr = délka datového řetězce po kompresi / původní délka
- Komprese zvuku >>> psychoakustický model lidského ucha
- Komprese obrazu >>> transformace, omezení transformačních koef., LZW…
- Komprese videa >>> vyhodnocení změny mezi následujícími snímky
- **Statistické kompresní metody** >>> pravděpodobnost výskytu jednotlivých dat (znaků, čísel,…) v datovém řetězci
- **Slovníkové kompresní metody** >>> vyhledávání podřetězců dat, které se často v datovém řetězci vyskytují. Tvorba slovníku z podřetězců. >>> LZW, …

RLE kódování

- **RLE kódování**
- Run-length encoding
- Bezztrátová kompresní metoda, kódování posloupnosti stejných hodnot do dvou čísel >>> délka posloupnosti, znak (číslo…)
- Využití >>> především u komprese obrazových dat, kde se v obraze mohou vyskytovat barevné plochy se stejnou hodnotou >>> jpg, tiff, …
- **(-) Nevýhody** – V nejhorším případě, kdy se data ani jednou neopakují může být kompresní poměr = 2, tj. komprimovaný datový řetězec je 2x větší než původní
- Př.:

datový řetězec:	AAAAAFFFFCHHH	(13 znaků)
kódovaný datový řetězec:	5A4F1C3H	(8 znaků)
kompresní poměr:	0,62 (62%)	

Huffmanovo kódování

- **Huffmanovo kódování**
- Algoritmus navržen Davidem Huffmanem (1952)
- Využití prefixového kódu - kód žádného znaku není prefixem jiného znaku, neprefixový kód: Morseova abeceda: A (.-), M(--), J(.---)
- Proměnná délka kódových slov >>> „znaky“, které jsou nejvíce četné mají nejkratší délku a naopak: 111011011, A (0), E(10), G(11)
- **Algoritmus kódování:**

1. Zjištění četnosti jednotlivých „znaků“
2. Vytvoření jednotlivých kódů na základě četností
3. Nahrazení jednotlivými znaků v datovém souboru nalezenými kódy

(+) Výhody – rychlá komprese a dekomprese, nenáročné na paměť

(-) Nevýhody – nutnost nalezených kódů, menší kompresní poměr

Huffmanovo kódování

- **Příklad:** znakový řetězec ABRAKADABRA
- 1) četnosti A (5x – 0,46), R (2x – 0,18), B (2x – 0,18), K (1x – 0,09), D (1x – 0,09)
- 2) vytvoření tabulky dle četností
- 3) poslední dvě četnosti se sečtou a zařadí se do tabulky, sčítá se až do 1

A 0,46		A 0,46		A 0,46		KDBR 0,54	1
R 0,18		R 0,18		KDB 0,36	1	A 0,46	0
B 0,18		B 0,18	1	R 0,18	0		
K 0,09	1	KD 0,18	0				
D 0,09	0						

- 4) Posledním dvěma slovům v každém sloupci tabulce přiřadíme 1 (vyšší četnost) a 0 (nižší četnost)
- 5) Výsledný kód znaku >>> posloupnost 0 a 1 dle toho jak se znak seskupoval s dalšími znaky, např. pro znak K: (1) – KDBR, (1) KDB, (0) KD a (1) K. Výsledné kódy A (0), R (10), B (111), K (1101), D (1100)
- 6) Výsledný řetězec pro ABRAKADABRA: 0 111 10 0 1101 0 1100 0 111 10 0, tj.: 011110011010110001111100
- Kompresní poměr (pokud 1 znak = 8 bit.): 0,26 (26%)

Aritmetické kódování

- **Aritmetické kódování**

- Huffmanův kód >>> problém při stejné pravděpodobnosti výskytu >>> možné řešení >>> aritmetické kódování

- Pro bezztrátovou kompresi dat, proměnná délka kódových slov jako u Huffmanova kódování, pří kódování se však vstupní znak nenahrazuje specifickým kódem, ale výsledek, tj. vstupní datový řetězec se nahradí reálným číslem z intervalu <0,1).

- **Algoritmus kódování:**

1. Zjištění četnosti (pravděpodobnosti výskytu) jednotlivých „znaků“
2. Dle pravděpodobnosti výskytu znaků se umístí znak v intervalu <0,1)
3. Celý interval <0,1) je postupně omezován z obou stran na základě přicházejících znaků.
4. Každý znak vybere z aktuálního intervalu odpovídající poměrnou část >>> nový základ pro následující symbol.
5. Po průchodu (načtení) všech znaků dostáváme podinterval z intervalu <0,1), výsledkem je pak libovolné reálné číslo z tohoto intervalu.
6. Na konec kódované zprávy dáme speciální znak, jinak při dekódování není možné určit konec datového toku, nebo uložíme délku původní posloupnosti znaků

Aritmetické kódování

- **Aritmetické kódování** – př.
- Datový řetězec CBAABCADAC (10 znaků)
- 1) Pravděpodobnosti výskytu A – 0.4 (P1), B – 0.2 (P2), C – 0.3 (P3), D – 0.1 (P4)
- 2) rozdělení v intervalu <0, 1):

<0, P1), <P1, P1 + P2), <P1 + P2, P1 + P2 + P3), <P1 + P2 + P3, P1 + P2 + P3 +P4)

Kumulativní pravděpodobnosti:

Q0 = 0, Q1= P1, Q2 = P1 + P2, … ,QN = P1 + P2 + … + PN = 1

<0, Q1), <Q1, Q2), <Q2, Q3), <Q3, Q4)
<0, 0.4), <0.4, 0.6), <0.6, 0.9), <0.9, 1)
A **B** **C** **D**

Aritmetické kódování

- **Aritmetické kódování** – př.
- Datový řetězec CBAABCADAC (10 znaků)
- Rozdělení intervalu

<0, 0.4), <0.4, 0.6), <0.6, 0.9), <0.9, 1)
A **B** **C** **D**

- 3) Kódování >>> postupné omezování intervalu I = <0, 1), I = <L, H)

>>> Postupně jsou brány znaky z datového řetězce, k nim známe IZ = <ZL, ZH)
>>> Nová hodnota intervalu IN = <L + ZL*(H - L), L + ZH*(H - L))

C >>> I = <0, 1), IN = <0 + 0.6*(1 – 0), 0 + 0.9*(1 - 0)) = <0.6, 0.9)
B >>> I = <0.6, 0.9), IN = <0.6 + 0.4*(0.9 – 0.6), 0.6 + 0.6*(0.9 – 0.6)) = <0.72, 0,78)
A >>> I = <0.72, 0.78), IN = <0.72 + 0*(0.78 – 0.72), 0.72 + 0.4*(0.78 – 0.72)) = <0.72, 0,744)
A >>> I = <0.72, 0.744), IN = <0.72 + 0*(0.744 – 0.72), 0.72 + 0.4*(0.744 – 0.72)) = <0.72, 0,7296)
B >>> I = <0.72, 0.7296), IN = <0.72 + 0.4*(0.7296 – 0.72), 0.72 + 0.6*(0.7296 – 0.72)) = <0.72384, 0.72576)
….
….
B >>> I = <0.72519936, 0.725208576), IN = <0.7252048896, 0.7252076544), C = 0.725205

Aritmetické kódování

- **Aritmetické dekódování** – př.
- Datový řetězec CBAABCADAC (10 znaků)
- Rozdělení intervalu

<0, 0.4), <0.4, 0.6), <0.6, 0.9), <0.9, 1)
A **B** **C** **D**

- Výsledek kódování >>> C = 0.725205

1) Počáteční hodnota intervalu dekódování I = <0, 1)
2) dekódování znaku: K = ((C – L) / (H – L)); ZL <= K < ZH >>> nalezneme odpovídající znak
3) počítáme nový interval IN = <L + ZL*(H - L), L + ZH*(H - L))
I = <0, 1), K = 0.725205, znak = C, IN = <0 + 0.6*(1 – 0), 0 + 0.9*(1 – 0) = <0.6, 0.9)
I = <0.6, 0.9), K = 0.41735, znak = B, IN = <0.6 + 0.4*(0.9 – 0.6),0.6 + 0.6*(0.9 – 0.6)) =<0.72, 0.78)
I = <0.72, 0.78), K = 0.08675, znak = A, IN = <0.72, 0.744)
I = <0.72, 0.744), K = 0.216875, znak = A, IN = <0.72, 0.7296)
….