INTRODUCERE

Compresia Datelor – metode si tehnici de reducere a volumului unui set de date reprezentate sub forma binara, cu posibilitatea reconstituirii lor fie exact, fie aproximativ.

Set de date = sir finit de simboli, cu reprezentare numerica unica (*cod numeric original*), in care nr. simbolilor ce-l compun (*volumul setului de date*) > nr. simbolilor distincti ce construiesc setul de date.

1 simbol → 8 biti (standard ASCII) => setul de date se construieste din max. $2^8 = 256$ simboli distincti

Alfabet = multimea simbolilor distincti dintr-un set de date

Frecventa diferita de aparitie a simbolilor de-a lungul setului de date => recodificarea alfabetului folosind coduri de lungime variabila (simbolilor ce apar frecvent le sunt asociate coduri mai scurte decat sibolilor ce apar mai rar in setul de date)

Lungimea setului de date = nr. total de biti pe care este reprezentat setul de date

Volumul unui set de date = numarul de simboli (nu neaparat distincti) care compun setul de date

Compresie → *conservativa* – studiaza metodele de reconstructie exacta a setului de date din versiunea sa comprimata (text, progr. sursa)

→ *neconservativa* – studiaza metode de reconstructie aproximative (inexacte) a setului de date original (semnal audio, imagine)

Rata de compresie = parametru prin care se masoara performantele metodei de compresie utilizate

$\gamma = \left[1 - \frac{\#c}{\#o}\right] \times 100\%$, #c lungimea setului de date comprimat, #o lungimea setului de date original.

γ=100%, metoda de compresie ideala, castig maxim

γ=0%, metoda nu realizeaza compresie, castig nul (storing)

γ<0%, metoda nu realizeaza compresie, expandarea datelor

γ masoara castigul relativ de lungume obtinut prin utilizarea unei metode de compresie => $\gamma \in (0,100)\%$ pt o metoda de compresie eficienta.

Viteza de compresie/decompresie ~ $1/\gamma$ (depinde de complexitatea metodei; compromis intre cei doi parametri).

Compresie conservativa - rata de compresie mare, vit. de compresie scazuta

Compresie neconservativa - rata de compresie mica, vit. de compresie mare

Expandare = cresterea lungii setului de date procesat fata de cel original (fenomen produs de metode de compresie ineficiente)

Decompresie = actiunea de reconstituire (exacta sau aproximativa) a setului de date original, plecand de la un set de date original

Factor de compresie $\gamma' = \frac{\#c}{\#o} \times 100\% = 100\% - \gamma$

Metoda de compresie eficienta ⇔ factor de compresie mic

Obiectivul unei metode de compresie a datelor = detectarea si eliminarea *redundantelor* din setul de date (adica a repetabilitatii datelor)

Redundanta – constanta a setului de date, care nu depinde de metoda de compresie utilizata

- se masoara cu ajutorul unui *model statistic* asociat setului de date, placand de la frecventele de aparitie ale simbolilor alfabetului setului de date

Frecventa de aparitie = raportul intre numarul aparitiilor simbolului si numarul total de simboli ai setului de date, notata cu $\upsilon(s)$ (frcventa de aparitie a simbolului s).

- constituie o estimatie a probabilitatii de aparitie a simbolului s, P(s).

Entropie = numarul minim de biti suficient pentru a recodifica simbolul s, fara a se realiza confuzia cu alti simboli.

- numarul poate fi fractionar, nu neaparat intreg, dar de obicei se rotunjeste la intregul superior
- va fi cel mult egala cu nr. de biti ai codului original
- diferenta dintre ele este o masura a redundantei intrinseci (nr. de biti inutili prezenti in codul original)
- $H(s) = -\log_2 \upsilon(s)$ => $\upsilon(s) = 2^{-H(s)} \in (0,1]$
- $0 < \upsilon(s) \leq 1$ => $H(s) > 0$

Ex: nr. aparitii(a) = 38 intr-un text cu Vol = 100 caractere => υ(a)=0,38 si H(a)= -$\log_2$(0,38)≈1,4 biti, deci 6,6 biti (din 8) din codajul initial sunt inutili (redundanti) => sunt suficienti 2 biti pentru alocarea literei „a” la o prima comprimare a textului.

Cantitatea de informatie $I(s) = \frac{C}{\upsilon(s)}$, C – constanta de proportionalitate

Un simbol care apare frecventa in setul de date transporta o cantitate de informatie mica si va primi un numar mic de biti de recodificare: $H(s)=\log_2 I(s)$.

$R(D) = \sum_{s \in D} [8 - H(s)]$, D – setul de date asupra caruia se opereaza; alocare initiala = 8 biti/simbol

Redundanta depinde direct de entropiile simbolilor si este cu atat mai mare cu cat entropiile sunt mai mici

$R(D) = \sum_{s \in D} [8 + \log_2 \upsilon(s)]$ => frecvente mari de aparitie ale simbolilor conduc la o redundanta mare a acestuia.

Ex: sirul „aaaaa” e reprezentat pe 5*8=40 biti. Daca el face parte dintr-un set mai mare de date, in care υ(a)=1/16 => H(a)=4 si R(aaaaa)=40-5*4=20 biti => doar 20 de biti de reprezentare sunt utili (50%). Codificarea ideala a cuvantului este pe 7 biti (in loc de 20), din care 4 biti pentru reprezentarea literei „a” si 3 biti pentru numarul de aparitii consecutive (5=101).