



Laborator PS

ALGORITMI DE COMPRESIE

Algoritmul Shannon-Fano

Prof. dr. ing. Dan STEFANOIU

As. Ing. Alexandru DUMITRASCU

1. Algoritmul SHANNON-FANO

Metoda **SHANNON-FANO** consta in constructia arborelului binar asociat setului de date D , care se efectueaza dupa urmatorul algoritm (5 pasi):

Pas 1: Se parcurge setul de date (D) in mod secvential, citind fiecare simbol in parte:

- Se construiesc alfabetul de ordin 0 (A^0)
- Se contorizeaza nr. de aparitii ale fiecarui simbol $s \in A^0 \Rightarrow$ contorul $N(s) \in \mathbf{N}$

$$v(s) = \frac{N(s)}{\#D}, \quad \forall s \in A^0 \quad (\text{estimatie probabilitatii de aparitie a lui } s)$$

$$\#D = \sum_{s \in A^0} N(s) \quad (\text{volumul setului de date este constant})$$

Pas 2: Simbolii din A^0 se aranjeaza in ordinea descrescatoare a valorilor contoarelor; in caz de egalitate se ordoneaza lexicografic.

$$A^0 = \{....., s, t,\} \text{ cu } N(s) \geq N(t)$$

Pas 3: Alfabetul A^0 se divizeaza in doua parti disjuncte, numite subalfabete (A^{0L} si A^{0R}), dar se pastreaza doua conditii de baza:

- se pastreaza ordinea simbolilor
- ponderile subalfabetelor sa fie cat mai apropiate

$$A^0 = A^{0L} \cup A^{0R}$$

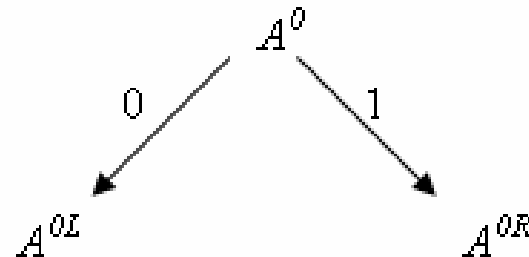
$$A^{0L} \cap A^{0R} = \emptyset$$

$$N(A^{0L}) = \sum_{s \in A^{0L}} N(s) \approx N(A^{0R}) = \sum_{s \in A^{0R}} N(s)$$

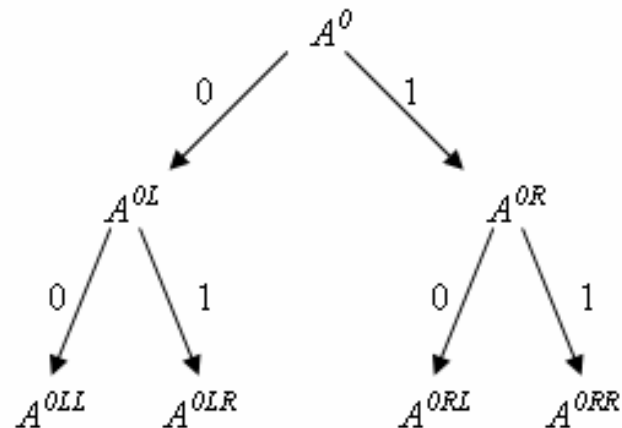
Obs.: La constructia subalfabetelor se pleaca simultan din cele doua capete ale alfabetului A^0 , mergand spre centru, comparand mereu ponderile partiale ale subalfabetelor, pana la epuizarea simbolilor alfabetului A^0 .

$$A^0 = \{s_1, s_2, \dots, s_k, \dots, s_{N-1}, s_N\}$$
$$\xrightarrow{A^{0L}} \qquad \qquad \qquad \xleftarrow{A^{0R}}$$

Pas 4: Se construiesc arborele binar



Pas 5: Se repeta pasii 3 si 4 pentru fiecare nod terminal al arborelui (subalfabet) care contine mai mult de un simbol.



STOP: Procesul de constructie a arborelui se opreste cand toate subalfabetele nodurilor terminale (frunzelor) contin doar un singur simbol. Codul unui simbol se obtine parcurgand arborele de la radacina pana la nodul ocupat de acel simbol, concatenand valorile arcelor prin care se trece.

Obs: Pentru un simbol frecvent in setul de date codul corespondent este scurt, pentru ca nodul lui va fi in apropierea radacinii. Simbolii rari au o lungime mai mare a codului.



Compresia SHANNON-FANO

Pas 1: Se construiește arborele binar asociat setului de date D

Pas 2: Se construiește setul de date comprimat alcătuit din:

- A^0
- N – numărul de simboluri al alfabetului A^0 . Fiecare simbol este reprezentat inițial pe 8 biți.
- Tabela contoarelor asociate simbolurilor alfabetului. Fiecare contor va fi reprezentat pe maxim 32 de biți.
- Sirul noilor coduri, în ordinea în care se succed simbolii în setul original de date. Aceste coduri se iau din arborele binar construit.

Obs: Pe fluxul de ieșire avem două tipuri de date:

- utile (sirul noilor coduri)
- auxiliare (date referitoare la alfabet și contoare).

Algoritmul este eficient dacă lungimea informației auxiliare este mult mai mică decât lungimea informației utile. Informația auxiliară este strict necesară în faza de decompresie, în defavoarea ratei de compresie.



Decompresia SHANNON-FANO

Pas 1: Se citeste informatia auxiliara intr-o ordine prestabilita (de ex. N , A^0 , $N(s_1), \dots, N(s_N)$).

Pas 2: Se construiesc arborele binar plecand de la informatia auxiliara.

Pas 3: Se decripteaza sirul noilor coduri (setul de date comprimat) citind informatia utila bit cu bit. Simultan se parcurge arborele de la radacina catre frunze si la atingerea unei frunze se emite codul original, pe 8 biti, al simbolului asociat, dupa care se va reincepe decriptarea de la radacina. Procesul de decriptare continua pana la epuizarea bitilor de pe fluxul de intrare.

Avantaje:

Algoritmul de decompresie nu necesita cunoasterea initiala a lungimilor codurilor emise in faza de compresie.

Noile coduri pot fi depuse pe fluxul de iesire in faza de compresie fara a mai fi separate intre ele, deci fara a adauga alte coduri auxiliare.

Decriptarea este exacta pentru ca modelul arborescent este constuit cu acelasi algoritm in ambele faze.