Algoritmul SHANNON-FANO

Metoda SHANNON-FANO consta in constructia arborelului binar asociat setului de date, care se efectueaza dupa urmatorul algoritm:

Pas 1: Parcurgerea setului de date (D) in mod secvential, citind fiecare simbol in parte:

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

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

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

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

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

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:

- Sa se pastreze ordinea simbolilor
- Ponderile subalfabetelor sa fie cat mai apropiate

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

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

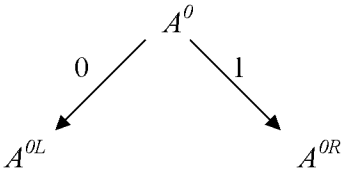
$$N(A^{0L}) = \sum_{s \in A^{0L}} N(s) \approx N(A^{0R}) = \sum_{s \in A^{0R}} N(s)$$

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.

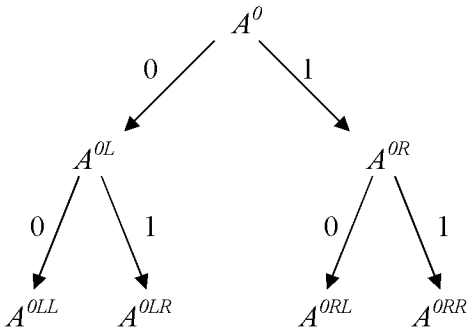
$$A^0 = \{s_1, s_2, \ldots\ldots, s_k, \ldots\ldots, s_{N-1}, s_N\}$$

A^{0L} (left) $\qquad$ A^{0R} (right)

Pas 4: Constructia arborelui binar



Pas 5: Se repeta pasii 3 si 4 pt 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 construieste arborele binar

Pas 2: Se construieste setul de date comprimat din:

- A^0
- N – numarul de simboli ai alfabetului A^0 . Fiecare simbol e reprezentat pe 8 biti.
- Tabela contoarelor asociate simbolilor alfabetului. Fiecare contoe va fi reprezentat pe maxim 32 de biti.
- Sirul noilor coduri, in ordinea in care se succed simbolii in setul original de date. Aceste coduri se iau din arborele binar construit.

Obs: Pe fluxul de iesire avem doua tipuri de date: utile (sirul noilor coduri) si auxiliare (date referitoare la alfabet si contoare). Algoritmul este eficient daca lungimea informatiei auxiliare este mult mai mica decat lungime informatiei utile. Informatia auxiliara este strict necesara in faza de decompresie, in defavoarea ratei de compresie.

Decompresia SHANNON-FANO

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

Pas 2: Se construieste 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 decripatea de la radacina. Procesul de decriptare continua pana la epuizarea bitilor de pe fluxul de intrare.

Avantaje:

- Algoritmul de decompresie nu necesita cunoasterea in avans a lungimilor codurilor emise in faza de compresie, datorita structurii arborescente a modelului statistic.
- 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.