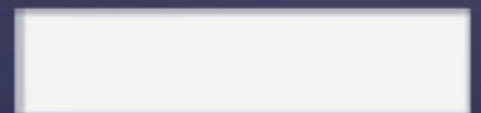
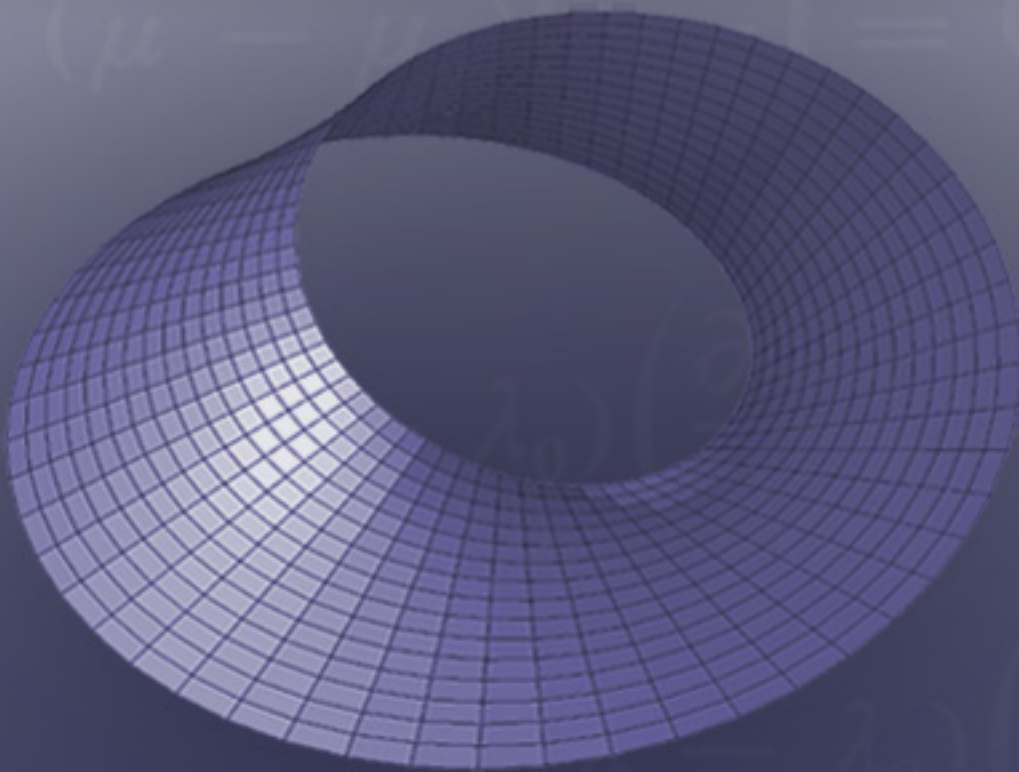


Michalis Meimaris Online Archive

Construction du code de huffman par
un algorithme se cah

[Cah Huffman]



CONSTRUCTION DU CODE DE HUFFMAN PAR UN ALGORITHME DE CAH

[CAH HUFFMAN]

par J.P. Benzécri*, M. Meimaris**

1 Code de Huffman et classification ascendante hiérarchique

La construction du code de Huffman rentre dans le cadre de la CAH avec un critère satisfaisant à l'axiome de la médiane (cf. R. Sedgewick ; Algorithms ; ch. 22 ; Addison Wesley 1983. Le critère est tout simplement la masse de la classe créée par réunion de a et b , soit :

$$\text{crit}(a,b) = m_a + m_b ; \text{ où } m_a = \sum \{m_i | i \in a\} ;$$

il existe pour ce critère une "formule de la médiane" : $\text{crit}(s, a \cup b) = (1/2)(\text{crit}(s,a) + \text{crit}(s,b) + \text{crit}(a,b))$. Pour Huffman, le tout est d'agréger de préférence les individus (des mots ou des lettres d'un alphabet) les moins usités, (la masse étant proportionnelle à la fréquence) ; ainsi chaque i se trouve dans l'arbre de la CAH à une profondeur d'autant plus faible (i.e. avec un nombre de prédécesseurs hiérarchiques d'autant plus petit) qu'il est plus lourd ; et un code i par un mot binaire spécifiant sa ligne de descendance (0 pour aînés et 1 pour benjamins) à partir du sommet S = ensemble des i = noeud n° (2CardI-1) ; soit e.g. :

$$\text{si } i = A(B(B(A(A(S)))))) ; \text{ le code est } 00110$$

le décodage d'un message se faisant en descendant l'arbre : e.g. dans un message commençant par 00110... on reconnaît que c'est précisément cette tranche de 5 bits qui désigne la première lettre parce que en descendant l'arbre à partir du sommet, en inclinant à gauche ou à droite selon que le chiffre est 0 ou 1, on aboutit à un individu, après cinq pas : i.e. AS, AAS, BAAS, BBAAS sont des noeuds, mais ABBAAS est un individu et il n'y a rien au-delà dans l'arbre ; donc les bits suivants servent à coder d'autres lettres etc. .

L'algorithme usuel de recherche en chaîne des couples de voisins réciproques permet de construire une CAH en un temps de l'ordre de n^2 (où $n = \text{Card I}$) ; mais ici un algorithme approprié permet d'aboutir en un temps de l'ordre de $n \log n$. L'objet de la présente note est de présenter l'algorithme de construction du code de H. en s'écartant le moins possible des voies ordinaires de la CAH. L'algorithme comprend un tri préalable (en $n \log n$) et des balayages (en n) seuls donnés explicitement ici.

(*) Professeur de statistique. Université Pierre et Marie Curie.

(**) Docteur-ès-Sciences.

Banque Agricole de Grèce.

2 L'algorithme de construction du code

```
entier CI,N,UR,UN;
réel tableau D[1:2*CI-1],VR[1:CI+1];
entier tableau IR[1:CI],A,B[CI+1:2*CI-1],PR[1:2*CI-2];
```

Commentaire : CI est le cardinal de l'ensemble des individus (lettres etc.) à classer; le tableau D donne en D[I] la masse (fréquence etc.) de l'individu I ; à partir de l'indice CI+1, le tableau D est mis initialement à l' ∞ (plus grand nombre réalisable) ; dans la suite, on mettra en D[N] le niveau (somme des masses des individus de la classe) du noeud N ; les valeurs contenues dans D[1:CI] se retrouvent rangées dans l'ordre croissant dans VR[1:CI] ; le tableau IR indique à quel indice I correspond la valeur dont le rang est R : VR[R] = D[IR[R]] ; nous supposons que les tableaux VR et IR ont été remplis par une procédure de tri usuel (cf. e.g. [PROB. ALG. TRI], in C.A.D. Vol VIII n° 1 pp 113-120 ; 1983) ; la valeur VR[CI+1] est une butée mise à l' ∞ (cf. *infra*) ; les tableaux A, B, PR donnent l'ainé, le benjamin et le père suivant les notations usuelles de la CAH ; N désigne le numéro du noeud en cours de création ; les indices UR et UN pointent respectivement dans VR[1:CI+1] et D[CI+1:2*CI-1] vers la masse du plus léger individu et du plus léger noeud non agrégés (quand tous les individus sont agrégés, UR pointe vers la butée CI+1) ; comme il apparaîtra au cours du déroulement de l'algorithme, ces pointeurs nous dispensent de constituer un tableau spécial pour numérotter les sommets : en effet les individus sont agrégés par ordre de masse croissante, du plus léger au plus lourd ; et de même pour les noeuds ; les sommets sont donc d'une part les individus IR[UR:CI] ; d'autre part les noeuds [UN:N-1].

```
procédure AVANCE(F) ; entier F; début
  si D[UN] < VR[UR]
    alors début F:=UN; UN:=UN+1 fin
    sinon début F:=IR[UR]; UR:=UR+1 fin;
  PER[F]:=N fin ;
```

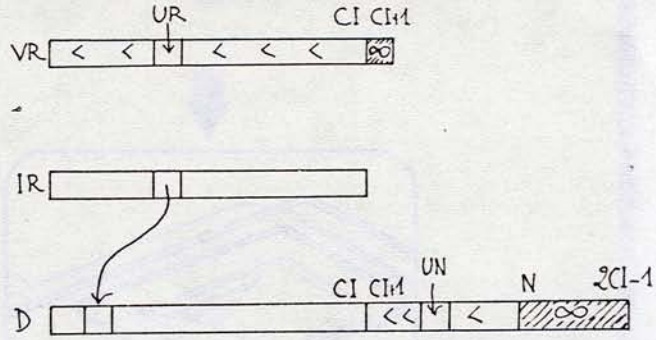
Commentaire : la procédure choisit pour fils F du noeud N qu'on crée, la classe la plus légère disponible : soit un noeud, désigné par le pointeur UN ; soit un individu dont le rang est désigné par le pointeur UR (le n° de l'individu étant IR[UR]) ; on avance le pointeur correspondant à la classe choisie ; la procédure est utilisée successivement deux fois pour construire l'ainé et le benjamin de N.

```
UR:=1; UN:=CI+1 ;
```

Commentaire : initialement, aucun noeud n'existe ; les individus sont tous des sommets, constituant l'intervalle [1:CI]=IR[1:CI] (numérotage initial et numérotage par rang).

```
pour N:=1 pas 1 jusqu'à 2*CI-1 faire début
  AVANCE(A[N]); AVANCE(B[N]);
  D[N :=D[A[N]]+D[B[N]] fin
```

Commentaire : Le graphique ci-dessous résume la procédure de recherche des fils suivant les pointeurs :



(les signes \ll , indiquent que les valeurs sont rangées dans l'ordre croissant de gauche à droite).