

K. BENSALÉM

**Algorithme accéléré des plus proches voisins
suivant une hiérarchie de boules emboîtées**

Les cahiers de l'analyse des données, tome 9, n° 1 (1984),
p. 123-124

http://www.numdam.org/item?id=CAD_1984__9_1_123_0

© Les cahiers de l'analyse des données, Dunod, 1984, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

ALGORITHME ACCÉLÉRÉ DES PLUS PROCHES VOISINS SUIVANT UNE HIÉRARCHIE DE BOULES EMBOÎTÉES

[VOIS. BOUL. HIER.]

par K. Bensalem

L'algorithme de recherche des plus proches voisins après agrégation en boules de rayon borné [VOIS. BOUL.], n'est pleinement efficace que si le nombre de points I contenu dans chaque boule est peu élevé : ce nombre devrait être de l'ordre de grandeur du nombre des voisins cherchés, (i.e. 10 ou 20) et en tout cas inférieur à 100. Cependant l'algorithme d'agrégation [ALG. AGR. RAY.] lui-même, a un temps de rotation de $KI \times KN$: si donc le cardinal KI de l'ensemble des points s'élève, il est dangereux de faire croître simultanément KN (nombre de boules), car on s'expose à des temps de calculs en $(k(KI))^2$ si $(KN = k \times KI)$... Quel que soit KI, il semble qu'on ne doive pas demander à [ALG. AGR. RAY.] de construire plus de 100 boules. Dès lors, si l'on veut maintenir l'effectif du contenu de chaque boule à quelques dizaines (et en tout cas en dessous de 100, ainsi qu'on l'a dit), la seule solution semble être d'appliquer à chaque boule l'algorithme [ALG. AGR. RAY.], avec un rayon RAY/2 (ou RAY/4 si nécessaire) afin d'obtenir une subdivision en boules. Le procédé peut en principe être répété itérativement ; et même si avec les données usuelles (KI 10.000) deux niveaux de boules semblent suffire, on donne ici un algorithme adapté à un nombre quelconque de niveaux ; parce que, avec des notations convenables cet algorithme général n'est pas plus compliqué que l'algorithme à deux niveaux. En fait, vis-à-vis même de l'algorithme de [VOIS. BOUL.] à un seul niveau, les modifications sont minimales ; en sorte que, une fois fixées les notations, nous écrirons l'algorithme presque sans commentaire.

```
entier KI, KJ, KVMIN, KVMAX, KV, KVO, I, KZ, KNx, KxBO, KxB, Nx, Mx ;
réel DIS, DISO, DSU, CD, AD, RAX ;
entier tableau IV[1:KVMAX]; IDNx, IFNx, NxB[1:KN] ;
réel tableau POSIJ[1:KI, 1:KJ], PJ[1:KJ], DI[1:KI],
          CENx J [1:KNx, 1:KJ], DNx[1:KNx] ;
étiquette BOULIER, ETI1, ETI2, ETI3 ;
réel procédure DISTANCE(PJ1[1:KJ], PJ2[1:KJ]) ;
```

Commentaire : les déclarations qui précèdent diffèrent de celles de [VOIS. BOUL.] en ce que plusieurs sigles sont affectés de l'indice minuscule x. Cet indice varie de 1 à h, où h désigne le nombre des niveaux : le cas de [VOIS. BOUL.] est donc $h = x = 1$. Ici prenons pour fixer les notations le cas $h = 2$; il y a KN_2 boules du niveau supérieur (2), dont le rayon est RA_2 ; chacune de ces boules est réunion de boules du niveau (1) dont le rayon est RA_1 ; il y a au total KN_1 boules du niveau 1. Boules et individus I sont numérotés de telle sorte que le contenu de toute boule forme une séquence : la boule N_2 du niveau 2 est réunion des boules N_1 du niveau 1 dont l'indice varie de $IDN_2[N_2]$ à $IFN_2[N_2]$; la boule N_1 du niveau 1 comprend des

(1) Ingénieur informaticien de la Faculté des Sciences de Tunis.
Etudiant en 3^o cycle. Université Pierre et Marie Curie.

individus I dont l'indice varie de IDN[N1] à IFN1[N1]. Le centre de la boule Nx du niveau x est CENxJ[Nx,..] ; la distance de ce centre au point PJ est DNX[Nx]. Au cours de l'exécution du programme, on s'intéresse à un nombre variable KBx de boules du niveau x ; l'indice de la Mx-ème boule du niveau x qu'on a retenue, est donnée par NxB[Mx].

```
lire entier KI,KJ,KVMIN,KVMAX,{KNx|x = 1,..., h} ;
lire réel CD,AD,{RAX|x = 1,..., h} ;
lire entier tableau{IDNx,IFNx[1:KNx]}x=1,..., h} ;
lire réel tableau POSIJ[1:KI,1:KJ],PJ[1:KJ],
{CENxJ[1:1KNx,1:KJ]}x = 1,...,h} ;
```

Commentaire : Au lieu d'utiliser un indice minuscule x, on aurait pu créer un véritable indice supplémentaire IX, variant de 1 jusqu'au niveau supérieur IM = h ; ainsi par exemple la suite des entiers KNx occuperait un tableau KNX[1:IM] ; mais les différents étages IX du tableau des centres, auraient alors des dimensions inégales. Finalement on a préféré laisser en dehors de l'ALGOL l'expression de la profondeur du programme : le passage en FORTRAN pouvant se faire selon la nécessité dans le cas h = 2 ou éventuellement h = 3, voire au-delà. Ceci dit, le programme comprend une tête ; puis après l'étiquette BOULIER, une suite de h+1 blocs numérotés de h à 0 : les blocs extrêmes h et 0 sont invariants ; et on les a déjà vus (à des détails minimes près) dans [VOIS. BOUL.] (cas h = 1). Les blocs intermédiaires, (h-1 ; h-2 ; ... ; 1) tous identiques entre eux sont en nombre variable : nous écrirons ici un seul de ces blocs. Pour plus de clarté on a encadré les différentes parties du programme.

```
TETE pour Nh:=1 pas 1 jusqu'à Knf faire
      DNh[Nh]:=DISTANCE(PJ[.],CENhJ[Nh,.]) ;
DIS:=CD*RAI;DISO:=-RAh;K1BO:=K2BO:=.....:=KhBO:=0;KVO:=0
```

BOULIER ;

```
niveau h
      KhB:=KhBO ;
      pour Nh:=1 pas 1 jusqu'à KNh faire
        si DISO+RAH<DNh[Nh] DIS+RAH alors début
          KhB:=KhB+1;NhB[KhB]:=Nh fin ;
```

```
niveau x
on note y=x + 1
      KxB:=KxBO ;
      pour My:=1 pas 1 jusqu'à KyB faire
        pour Nx:=IDNy[NyB[My]] pas 1 jusqu'à IFNy[NyB[My]] faire début
          si KyBO<My alors DNX[Nn]:=DISTANCE(PJ[.],CENxJ[Nx,.]) ;
          si DISO+RAX<DNx[Nx]<DIS+RAX alors début
            KxB:=KxB+1;NxB[KxB]:=Nx fin fin
      KyBO:=KyB
```

```
niveau 0 et retour à BOULIER
      Kv:=KVO;DSU:=DIS ;
      pour M1:=1 pas 1 jusqu'à K1B faire
        pour I:=IDN1[N1B[M1]] pas 1 jusqu'à IFN1[N1B[M1]] faire début
          si K1BO<M1 alors DI[I]:=DISTANCE(PJ[.],POSIJ[I,.]) ;
          si DI[I]>DSU ou DI[I]<DISO aller à ETI3 ;
          si KV<KVMAX alors KV:=KV+1 ;
          IV[KV]:=I;KZ:=KV ;
          ETI1 ;
          si KZ<2 aller à ETI2 ;
          si DI[I]<DI[IV[KZ-1]] alors début
            IV[KZ]:=IV[KZ-1];IV[KZ-1]:=I;KZ:=KZ-1 ;
            aller à ETI1 fin ;
          ETI2 ;
          si KV=KVMAX alors DSU:=DI[IV[KV]] ;
          ETI3 fin ;
      si KV<KVMIN alors début
        DISO:=DIS;DIS:=AD*DIS;K1BO:=K1B ;
        aller à BOULIER fin
```