LES CAHIERS DE L'ANALYSE DES DONNÉES

J. JUAN

Programme de classification hiérarchique par l'algorithme de la recherche en chaîne des voisins réciproques

Les cahiers de l'analyse des données, tome 7, n° 2 (1982), p. 219-225

<http://www.numdam.org/item?id=CAD_1982__7_2_219_0>

Les Cahiers de l'Analyse des Données
Vol. VII - 1982 - n° 2 - p. 219-225

PROGRAMME DE CLASSIFICATION HIÉRARCHIQUE PAR L'ALGORITHME DE LA RECHERCHE EN CHAÎNE DES VOISINS RÉCIPROQUES [PROG. C.A.H. CHAÎNE RECIP.]

par J. Juan (1)

1 *Rappels et notations*

Le critère de la variance est utilisé pour construire la hiérarchie. Il permet de travailler directement sur la matrice des données. L'écart η entre deux éléments i et i' de l'ensemble à hiérarchiser sera donc défini par :

$$\eta(i,i') = (\rho_i \rho_{i'}/(\rho_i + \rho_{i'}))\, d^2(i,i')$$

où ρ_i est le poids de l'élément i et d^2 une distance euclidienne au carré.

Deux éléments qui s'agrègent sont remplacés par leur centre de gravité. Le poids du noeud ainsi créé est la somme des poids de ses deux successeurs. L'indice de niveau ν du noeud $g = i \cup i'$ est :

$$\nu(g) = \eta(i,i')$$

Les éléments i et i' seront agrégés si i est le plus proche voisin (au sens de η) de i' et réciproquement. Dans ce cas, i et i' sont appelés voisins réciproques.

2 *La recherche en chaîne des voisins réciproques*

2.1 Principe (d'après [C.A.H. CHAINE RECIP.]) : A partir d'un élément s_1 on construit une chaîne d'éléments $s_1 \to s_2 \ldots \to s_n$ telle que s_n (pour $n > 1$) soit un plus proche voisin (PPV) de s_{n-1}. On sait qu'on aboutit nécessairement à un élément s_k ($k > 1$) qui a comme PPV l'élément s_{k-1}. Dans ce cas, la paire $\{s_k , s_{k-1}\}$ est constituée de deux voisins réciproques que l'on doit agréger.

La recherche d'une nouvelle paire de voisins réciproques s'effectue en poursuivant la chaîne à partir de s_{k-2} (si $k \geq 3$) ou en la recommençant (si $k = 2$) depuis le noeud créé.

La classification est terminée lorsque tous les noeuds ont été formés. (Pour un ensemble de m individus, il y a m-1 noeuds à créer).

---

(1) *Boursier D.G.R.S.T. du laboratoire de statistique de l'université Pierre et Marie Curie (PARIS VI).*

2.2 Algorithme : Lors de la création d'un noeud ou de la mise en chaîne d'un voisin, on doit mettre à jour certains fichiers concernant les individus. Cette mise à jour peut s'effectuer simplement en utilisant une table qui permet de retrouver la position des invividus dans les fichiers. C'est cette solution que nous décrivons ici.

```
lecture des poids et coordonnées des individus
faire : JU=0 ; NR=NI; IL=1,
Pour I=1 jusqu'à NI:BO(I)=I
```

Commentaire : BO est la table des adresses des individus. IL est le nombre d'individus de la chaîne, JU et NR sont respectivement le compteur de noeuds et le nombre d'éléments restant à hiérarchiser.

```
1   DD = ∞
    Si IL>1 alors DD=D(BO(IL-1))
    Si (IL+1)>NR alors aller à 2
    INDIC=0
    Pour tout I=IL+1 jusqu'à NR faire :
       Calcul de l'écart DDK entre BO(IL) et BO(I)
       Si DD>DDK alors INDIC=I et DD=DDK
    Si INDIC>0 alors aller en 3
```

Commentaire : Ce bloc de programme cherche un PPV à l'élément BO(IL). On ne le cherchera pas parmi les individus BO(1),..., BO(IL-2) qui en possèdent déjà un. Ni lorsque la chaîne contient tous les individus car dans ce cas BO(NR) et BO(NR-1) sont nécessairement voisins réciproques. Le calcul de l'écart DDK (qui se fait en bref par somme des carrés de différences de coordonnées) est en fait interrompu dès que DDK>DD.DD est en fin d'étape l'écart entre BO(IL) et son PPV. Pour I<IL le réel D(BO(I)) mesure l'écart entre les éléments BO(I) et BO(I+1) appartenant à la chaîne. INDIC est positif lorsque le PPV de BO(IL) n'est pas BO(IL-1).

```
2   JU=JU+1; DI(JU)=DD
    agréger BO(IL) et BO(IL-1)
    Si JU≥NI-1 alors aller à 4
    mettre le noeud créé à la place de l'élément BO(IL-1)
    Si IL<NR alors BO(IL)=BO(NR)
    NR=NR-1; IL=MAX(1,IL-2)
    aller à 1
```

Commentaire : L'indice de niveau du noeud formé par l'agrégation de BO(IL) et BO(IL-1) est placé dans le tableau DI. Si la classification n'est pas terminée (i.e. si JU≠NI-1) on met à jour les fichiers des individus : le noeud prend la place de BO(IL-1) et BO(NR) celle de BO(IL) (sauf si IL=NR). La chaîne aura en fin d'étape le plus grand des deux entiers IL-2 et 1.

```
3   D(BO(IL))=DD; IL=IL+1
    permuter BO(IL) et BO(INDIC) sauf si INDIC=IL
    aller à 1
```

Commentaire : La chaîne est augmentée de l'élément BO(INDIC).

```
4   Trier les indices de niveaux du tableau DI
    écrire la hiérarchie
    fin
```

Commentaire : La classification étant terminée, il faut écrire les noeuds de la hiérarchie dans l'ordre de leurs indices de niveaux.

3 Les performances

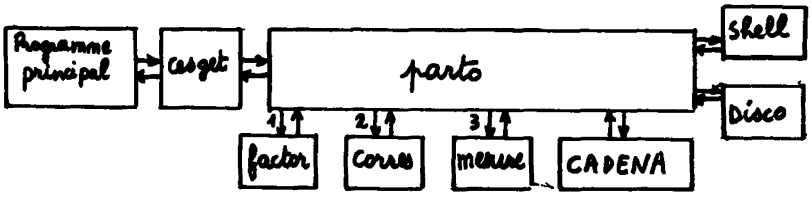
Les performances du programme CAVOR de recherche en chaîne des voisins réciproques sont comparées à celles de deux autres programmes : le premier utilise également l'algorithme des voisins réciproques (programme HIVOR) et le second est basé sur l'algorithme des graphes réductibles ("seuil de l'inertie"). (cf. [*PROG. C.A.H. RECIP.*] pour HIVOR et une description des graphes réductibles).

graphes réductibles "seuil inertie"		voisins réciproques prog. HIVOR		voisins réciproques (recherche en chaîne) prog. CAVOR	
Card I	Temps	Card I	Temps	Card I	Temps
25(8)	0,31	25(8)	0,25	25(8)	0,23
176(7)	3,3	176(7)	1,02	176(7)	0,97
188(26)	9,2	188(26)	2,5	188(26)	2,3
376(6)	13,1	376(6)	3,0	376(6)	3,1
564(6)	29,4	1128(6)	21,2	1128(6)	22,1
1785(6)	816,9	1880(6)	54,4	1880(6)	58,4

Comparaisons faites sur IBM 370/168. Les temps d'exécution sont en secondes et les nombres entre parenthèses indiquent le nombre de variables ou de facteurs. Les résultats observés (sur des tableaux provenant des mêmes données) montrent nettement la plus grande efficacité des algorithmes basés sur les voisins réciproques. On constate en outre que si le programme HIVOR est le plus performant pour les fichiers de grande taille (i.e. supérieurs à 300 individus), le programme CAVOR reste rapide (surtout s'il y a un grand nombre de variables. Il est d'autre part le seul à assurer un temps de calcul croissant en n^2, quelle que soit la structure de l'ensemble des n individus à classer (cf.[*C.A.H. CHAINE RECIP.*]).

4 Le programme CAVOR

La structure de CAVOR est exactement la même que celle de HIVOR:



Structure du programme CAVOR

Les seuls éléments différents par rapport à HIVOR (cf. [*PROG. C.A.H. RECIP.*]) sont le programme principal, PARTO et CADENA dont nous fournissons un listage.

```
C       PROGRAMME CAVOR.CLASSIFICATION ASCENDANTE BINAIRE,ALGORITHME
C       DES VOISINS RECIPROQUES PAR LA METHODE DE LA CHAINE.
C       STRATEGIE DE LA VARIANCE.
C       PAS DE FICHIER DE TRAVAIL.
C       LE FICHIER ILEC A HIERARCHISER EST UN FICHIER DE NI ELEMENTS; LE
C       PROGRAMME ADMET 3 TYPES DE FICHIER ILEC:
C       IOP=1 FICHIER DE FACTEURS
C       IOP=2 FICHIER DE CORRESPONDANCE
C       IOP=3 FICHIER DE MESURES CONTINUES.
C       SI IOP=3 ,LES NJ VARIABLES SONT CENTREES ET REDUITES.
C       POUR IOP=2 OU IOP=3 ON LIT SEULEMENT LES NJ VALEURS EN REELS.
C       POUR IOP=1 ON LIRA LE POIDS ET LES NJ COORDONNEES EN REELS. SI LE
C       FICHIER DES FACTEURS PROVIENT DE ADDAD , LE FORMAT A ECRIRE SERA
C       (40X,2E20.10/(4E20.10)).
C       LA HIERARCHIE EST ECRITE SUR LE FICHIER ISOR SELON LE FORMAT
C        (1X,4I5,E20.10) QUI CORRESPOND A:
C       NUM.DU NOEUD,NB ELEMENTS,NUM.AINE,NUM.BENJAMIN ET INDICE DU NOEUD
C
C       MEMOIRE :  9*NI+NI*NJ+2*NJ-4( VERSION HIERARCHIE USUELLE )
C
C       IL Y A 3 CARTES PARAMETRES:
C       1  TITRE( 20A4 )
C       2  NI,NJ,IOP,ILEC,ISOR( 5I5 )
C       3  FORMAT( 20A4 )
C
C       PROGRAMME DE J.JUAN ,22 FEVRIER 1982
C       REFERENCES : ARTICLES 'CAH.CHAINE.RECIP' ET 'PROG.CHAINE.RECIP'
C        DES CAHIERS DE L'ANALYSE DES DONNEES.DUNOD.
C
      EXTERNAL PARTO
      INTEGER TITRE(20),FMT(20)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      COMMON /IO/ ILEC,ISOR,FMT
      READ 1,TITRE
1     FORMAT(20A4)
      PRINT 2,TITRE
2     FORMAT(1X,20A4/)
      READ 3,NI,NJ,IOP,ILEC,ISOR
3     FORMAT(5I5)
      PRINT 4,NI,NJ,IOP,ILEC,ISOR
4     FORMAT(1X,'     NI     NJ    IOP   ILEC   ISOR'/1X,5I6//)
      READ 1,FMT
      PRINT 2,FMT
      NTOF=NI-1
      NTOT NTOF+NI
      MEMOIR=NTOT+4*NI+NI*NJ+2*NJ+3*NTOF
      CALL CESGET(PARTO,4,MEMOIR,85)
      STOP
5     CONTINUE
      PRINT 6,MEMOIR
6     FORMAT(1X,I7,'MOTS MINIMUMS A RESERVER.AUGMENTER ESPACE MEMOIRE')
      STOP
      END
      SUBROUTINE PARTO(V,MEMOIR)
      DIMENSION V(MEMOIR)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      LA=1+NI
      LB=LA+NI*NJ
      LC=LB+NTOT
      LD=LC+NI
      LE=LD+NI
      LG=LE+NI
      LN=LG+NJ
      LP=LN+NTOF
      LQ=LP+NTOF
      LR=LQ+NTOF
      IF(IOP.EQ.1) CALL FACTOR(V(1),V(LA))
      IF(IOP.EQ.2) CALL CORRES(V(1),V(LR),V(LA),V(LG))
      IF(IOP.EQ.3) CALL MESURE(V(1),V(LR),V(LA),V(LG))
      CALL CADENA(V(1),V(LA),V(LB),V(LC),V(LD),V(LE),V(LG),V(LN),V(LP),
     CV(LQ))
      CALL SHELL(V(LC),V(LQ))
      CALL DISCO(V(1),V(LB),V(LC),V(LN),V(LP),V(LQ))
      RETURN
      END
```

```
      SUBROUTINE CADENA(FI,FIJ,IP,SOM,D,BO,TAB,A,B,DI)
      INTEGER IP(NTOT),SOM(NI),BO(NI)
      INTEGER A(NTOF),B(NTOF)
      REAL FI(NI),FIJ(NI,NJ),D(NI),TAB(NJ),DI(NTOF)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
C         TABLEAUX UTILISES :
C     FI    : POIDS DES SUJETS A HIERARCHISER
C     FIJ   : DONNEES
C     IP    : CARDINAUX DES CLASSES
C     SOM   : NUMERO DES SOMMETS
C     D     : DISTANCES MINIMALES
C     BO    : NUMEROS DES SUJETS DU FICHIER..
C     A     : AINES
C     B     : BENJAMINS
C     DI    : INDICES DES NOEUDS
C     TAB   : COORDONNEES DU SUJET COURANT
C
C     AFINI : REEL PLUS GRAND QUE LA PLUS GRANDE DES PROXIMITES .
C     II    : NUMERO DU NOEUD VENANT D'ETRE CREE.
C     JU    : COMPTEUR DE NOEUD ET INDICE DES TABLEAUX A,B,DI
C     ANE   : INERTIE ( =SOMME DES INDICES DE NIVEAUX)
C     IL    : LONGUEUR DE LA CHAINE (=DERNIER ELEMENT DE CETTE CHAINE)
C     DD    : DISTANCE DE BO(IL) A SON PLUS PROCHE VOISIN.
C     NR    : TAILLE DU FICHIER DES ELEMENTS A HIERARCHISER.
      AFINI=1.E+50
      II=NI
      NR NI
      JU=0
      ANE=0.
      DO 30 I=1,NI
      IP(I)=1
      BO(I)=I
30    SOM(I)=I
C     CONSTITUTION DE LA CHAINE.
C     SI IL>1 LA CHAINE EST PROLONGEE.
C     SINON ELLE EST RECREE A PARTIR DE L'ELEMENT EN COURS.
C     LE CALCUL DE LA PROXIMITE DDK ENTRE LES ELEMENTS BO(IL) ET
C     BO(IR) POUR IR VARIANT DE IL+1 A NR EST ABANDONNE SI DDK > DD.
      IL=1
120   DD=AFINI
      IF(IL.LE.1) GOTO 125
      IVV=BO(IL-1)
      DD D(IVV)
125   ILL=BO(IL)
      PI FI(ILL)

      DO 2 J=1,NJ
2     TAB(J)=FIJ(ILL,J)
      IA=IL+1
      IF(IA.GT.NR) GOTO 128
      INDIC=0
      DO 130 IR=IA,NR
      I=BO(IR)
      DDK=0.
      PON PI*FI(I)/(PI+FI(I))
      DO 4 J=1,NJ
      RAP=TAB(J)-FIJ(I,J)
      DDK=DDK+PON*RAP*RAP
      IF(DDK.GE.DD) GOTO 130
4     CONTINUE
      INDIC=IR
      DD DDK
130   CONTINUE
      IF(INDIC.NE.0) GOTO 140
C     LES ELEMENTS BO(IL-1) ET BO(IL) SONT VOISINS RECIPROQUES.
C     ON LES AGREGE;AINES , BENJAMINS ,INDICE DE NIVEAU ET CARDINAL DU
C     NOUVEAU NOEUD SONT CONSERVES.ON MET A JOUR LE FICHIER DES DONNEES.
C
128   JU=JU+1
      A(JU)=SOM(ILL)
      B(JU)=SOM(IVV)
      DI(JU)=DD
      ANE=ANE+DD
      II II+1
      IP(II)=IP(SOM(ILL))+IP(SOM(IVV))
      SOM(IVV)=II
```

```
      IF(JU.GE.NTOF) GOTO 135
C     ON VIENT DE TESTER LA FIN DE LA CLASSIFICATION..
      PV=FI(IVV)
      PA=PI+PV
      DO 8 J=1,NJ
8     FIJ(IVV,J)=(PI*TAB(J)+PV*FIJ(IVV,J))/PA
      FI(IVV)=PA
C     ON A REMPLACE IVV PAR LE CENTRE DE GRAVITE DE IVV ET ILL.
C     IL RESTE A ENLEVER ILL DU FICHIER (I.E. DE BO).
      IF(IL.NE.NR) BO(IL)=BO(NR)
      NR NR-1
      IL-IL-2
      IF(IL.LE.0) IL=1
      GOTO 120
C     ETIQUETTE 140 TRAITE LE CAS OU BO(IL-1) ET BO(IL) NE SONT PAS
C     VOISINS RECIPROQUES.ON PLACE EN IL+1 LE VOISIN DE IL.
140   D(ILL)=DD
      IL=IL+1
      IF(IL.EQ.INDIC) GOTO 120
      IVV=BO(INDIC)
      BO(INDIC)=BO(IL)
      BO(IL) IVV
      GOTO 120
135   CONTINUE
      PRINT 27,ANE
27    FORMAT(1X///1X,'SOMME INDICES DE NIVEAUX: ',E10.5)
      RETURN
      END
      SUBROUTINE CORRES(FI,FJ,FIJ,TAB)
      REAL FI(NI),FJ(NJ),FIJ(NI,NJ),TAB(NJ)
      INTEGER FMT(20)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      COMMON /IO/ ILEC,ISOR,FMT
      DO 1 J=1,NJ
1     FJ(J)-0.
      T-0.
      DO 2 I=1,NI
      READ(ILEC,FMT) TAB
      FI(I) 0.
      DO 2 J=1,NJ
      A=TAB(J)
      FIJ(I,J)=A
      FI(I) FI(I)+A
      FJ(J)-FJ(J)+A
2     T=T+A
      DO 3 J=1,NJ
3     FJ(J)-SQRT(FJ(J)/T)
      DO 4 I=1,NI
      FI(I)=FI(I)/T
      DO 4 J=1,NJ
4     FIJ(I,J)=FIJ(I,J)/(T*FI(I)*FJ(J))
      RETURN
      END
      SUBROUTINE MESURE(FI,FJ,FIJ,TAB)
      REAL FI(NI),FJ(NJ),FIJ(NI,NJ),TAB(NJ)
      INTEGER FMT(20)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      COMMON /IO/ ILEC,ISOR,FMT
      DO 1 J=1,NJ
      FJ(J)=0.
1     TAB(J)=0.
      DO 2 I=1,NI
      FI(I)-1.
      READ(ILEC,FMT) (FIJ(I,J),J=1,NJ)
      DO 2 J=1,NJ
      A=FIJ(I,J)
      FJ(J)-FJ(J)+A
2     TAB(J)=TAB(J)+A*A
      AI-NI
      DO 3 J=1,NJ
      FJ(J)-FJ(J)/AI
      V=(TAB(J)-FJ(J)*FJ(J)*AI)/AI
3     TAB(J)=SQRT(V)
      DO 4 I=1,NI
      DO 4 J=1,NJ
4     FIJ(I,J)=(FIJ(I,J)-FJ(J))/TAB(J)
      RETURN
      END
      SUBROUTINE FACTOR(FI,FIJ)
      REAL FI(NI),FIJ(NI,NJ)
      INTEGER FMT(20)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      COMMON /IO/ ILEC,ISOR,FMT
```

```
      T 0.
      DO 1 I=1,NI
      READ(ILEC,FMT) FI(I),(FIJ(I,J),J=1,NJ)
1     T T+FI(I)
      DO 2 I=1,NI
2     FI(I) FI(I)/T
      RETURN

      END
      SUBROUTINE SHELL(IVO,DI)
      INTEGER IVO(NTOF)
      REAL DI(NTOF)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
C     TRI DU TABLEAU DI (INDICES DE NIVEAU ) PAR LA METHODE DE SHELL.
C     REFERENCES : KNUTH , SORTING AND MERGING (1973).
      DO 70 I 1,NTOF
70    IVO(I)=I
C     LE TABLEAU DES POINTEURS EST IVO.
      M=1
      NT=NTOF/3
71    M=3*M+1
      IF(M.LT.NT) GOTO 71
72    M=M/3
      IF(M.LE.0) GOTO 75
      K NTOF-M
      DO 74 I=1,K
      J=I
73    L=J+M
      IF(DI(IVO(L)).GE.DI(IVO(J))) GOTO 74
      KP=IVO(J)
      IVO(J)=IVO(L)
      IVO(L)=KP
      J=J-M
      IF(J.GT.0) GOTO 73
74    CONTINUE
      GOTO 72
75    RETURN
      END
      SUBROUTINE DISCO(NOM,IP,IVO,A,B,DI)
      INTEGER A(NTOF),B(NTOF),IP(NTOT),IVO(NTOF),NOM(NTOT),FMT(20)
      REAL DI(NTOF)
      COMMON /PAR/ NI,NJ,IOP,NTOT,NTOF
      COMMON /IO/ ILEC,ISOR,FMT
C     LES 2*NI-1 NUMEROS DES ELEMENTS DE LA HIERARCHIE TOTALE SONT
C     STOCKES DANS LE TABLEAU NOM.
      DO 2 I=1,NTOF
      NOM(I)=I
2     NOM(IVO(I)+NI)=NI+I
      NOM(NI)=NI
      DO 1 IR=1,NTOF
      I=IVO(IR)
      NOEU NI+IR
      ICAR IP(I+NI)
      IAIN=NOM(A(I))
      IBEN NOM(B(I))
      DNIV=DI(I)
      WRITE(ISOR,100) NOEU,ICAR,IAIN,IBEN,DNIV
1     CONTINUE
100   FORMAT(1X,4I5,E20.10)
      RETURN
      END
      SUBROUTINE CESGET(PARTO,LMOT,NBMOT,*)
C
C     CE SOUS PROGRAMME N'EST PAS A METTRE SI ON UTILISE LE CIRCE.
C    (IL SUFFIT ALORS DE METTRE LES CARTES DE CONTROLES SUIVANTES:
C              //GO.SYSLIB DD
C              //          DD DISP=SHR,DSN=SYS1.BIBLI.NIV1
C
C     SI ON UTILISE CE SOUS PROGRAMME ,L'ALLOCATION N'EST PAS DYNAMIQUE.
C
      DIMENSION S(10000)
      IF(NBMOT*LMOT*0.25.GT.10000) RETURN 1
      CALL PARTO(S,NNMOT)
      RETURN
      END
       SORTIES DE CAVOR
  HIERARCHIE DES SUJETS
    NI     NJ    IOP   ILEC   ISOR
   176      7     1     10     11
(40X,2E20.10/(4E20.10))
SOMME INDICES DE NIVEAUX: .18790E+01
```