KLASS:
Estudi d'un sistema d'ajuda al tractament estadístic de grans bases de dades
(Master Thesis)

Karina Gibert

Report LSI–91–7

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament de Llenguatges i Sistemes Informàtics

KLASS

Estudi d'un sistema d'ajuda al tractament estadístic de grans bases de dades[1]

TESI DE LLICENCIATURA

Karina Gibert Oliveras

Febrer 1991

[1]**Aquest treball ha estat parcialment finançat pel projecte ACRE - CAYCIT B-0145.**

Index

Agraïments

Vull començar aquest treball agraint a tothom l'ajuda que m'ha prestat. En especial a l'Ulises, amb qui he treballat amb tota llibertat, i sense la guia del qual aquesta tesina mai hauria pres cos. Doblement agraïda per l'atenció constant que m'ha dedicat. A en Tomàs, que m'ha assessorat en les qüestions de caire estadístic. A l'"*equip-Ulises*", (Enric, J.M., Javi, Lluís...), pel suport moral i intel·lectual que hi he trobat. A la Marta, que m'ha ajudat amb les esponges. A la meva mare, que en tot moment m'ha encoratjat a arribar fins al final. I, en definitiva, a tots aquells que amb el seu granet de sorra han contribuït al desenvolupament de la meva tesi de llicenciatura.

Capítol 1

INTRODUCCIÓ

"La diferència és el principi general de la multitud".
"La diferència és la forma que divideix el gènere en moltes espècies".
R.LLULL

1.1 La classificació

La classificació és la tècnica d'ús més comú per a separar dades en grups.

Moltes vegades, qui fa recerca s'enfronta a grans quantitats d'informació que serien pràcticament intractables a menys que es podessin agrupar en classes que, d'alguna forma, fossin susceptibles de ser considerades com a unitats.

En altres ocasions, simplement interessa esbrinar l'estructura interna de les dades per a utilitzar-la com a base de tasques posteriors, bé sia amb fins predictius o d'altres naturaleses. Per exemple, classificant un conjunt de malalties en funció dels símptomes que les identifiquen i assignant a cada grup un tractament, el diagnòstic d'un malalt pot reduir-se a veure a quina classe pertany.

En qualsevol cas, l'objectiu de les tècniques de classificació (clustering) és obtenir una representació esquemàtica i simple d'una matriu rectangular de dades, que en conservi el màxim d'informació. En general, es busca una jerarquia, un seguit de particions encaixades, cada cop més fines, sobre el conjunt d'observacions inicial.

La representació ideal és l'arbre jeràrquic, que té: en els nodes interns les diferents subdivisions establertes sobre els individus, a les fulles del corresponent

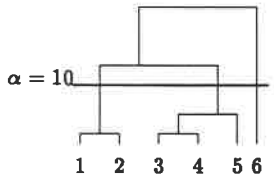


Figura 1.1: Arbre jeràrquic amb α-tall a nivell 10.

subarbre, els elements que formen part de cada subclasse, i branques de diferents longituds, que situen els nodes interns a diferents nivells respecte de l'horitzontal sobre la que es disposen els individus. El nivell dels nodes, normalment xifrat, indica el grau de semblança dels seus fills, i està relacionat directament amb la distància, entre aquests, en l'espai de les variables. Quant més gran sigui la longitud de la branca que uneix dos fills al seu pare, menys s'assemblen els nodes que representen.

El tall horitzontal de l'arbre donat un determinat nivell, anomenat α-partició (α-cut), proporciona una certa partició de la mostra en classes. Fent variar el nivell del tall, obtenim diverses classificacions dels mateixos individus, de diferents graus de generalitat.

El que es pretén amb la classificació és obtenir les classes *naturals* del domini estudiat, que abans de començar l'estudi són desconegudes, i això s'aconsegueix amb un estudi *a posteriori* de l'arbre de classificació. Així, el tall de la figura 1.1 particiona la mostra en tres classes: la primera conté els elements 1 i 2, la segona els 3, 4 i 5, i la tercera el 6.

1.2 Panoràmica històrica de la classificació

Classificar gent en tipus és un costum que s'ha vingut practicant des fa molt de temps.

Ja els Hindús usaven el sexe, les característiques físiques i el comportament per a classificar la gent en sis tipus que designaven amb noms d'animals.

Els primers físics de la Grècia i Roma clàssica desenvoluparen diverses tipologies basant-se en les variacions que diferents combinacions de quatre humors

bàsics de l'organisme humà produïen en les característiques físiques de l'individu. D'entre elles, la més famosa és la de Galen, que definia nou tipus temperamentals relacionats amb la susceptibilitat de les persones, diferents malalties i diferències individuals en el comportament.

Tot el coneixement real que tenim depèn dels mecanismes pels quals l'home distingeix la semblança de la diferència. Quan més gran sigui el nombre de distincions que aquest mecanisme sigui capaç de captar, més clara és la nostra idea de les coses. Quant més gran sigui el nombre d'objectes que tractem de reconèixer, més difícil serà establir aquests mecanismes, i també més necessari.

Els primers treballs de classificació es desenvolupaven, en la seva majoria, en els camps de la Biologia i la Zoologia, on es sol designar amb el terme Taxonomia als arbres de classificació.

Al segle XVIII, un dels més grans classificadors de la història, Linneus, efectua una classificació dels éssers vius encara vigent avui dia en el món de les ciències naturals.

En principi, la construcció d'una Taxonomia era més aviat un procés artístic, amb una forta intervenció del *sentit comú* del seu autor, i quedava un xic lluny del que habitualment entenem per treball científic. És a rel de les taxonomies numèriques, basades en les idees d'Andanson (s. XVIII), que comencen a desenvolupar-se tècniques més objectives. Durant la primera meitat del nostre segle, hi ha hagut nombroses temptatives de racionalitzar el procés mental que Linneus va emprar en el seu treball. Entre altres, Zubin i Thorndike [THOR53] apliquen tècniques de classificació numèrica a camps que no són les ciències naturals.

Però no és fins que la Informàtica pren cos a l'entorn universitari que apareixen els primers algorismes automatitzant aquest procés. Efectivament, la consolidació d'aquesta ciència ofereix medis per a la còmoda manipulació de les grans quantitats d'informació que es maneguen en el món de l'anàlisi de dades, i permet efectuar de forma ràpida la gran quantitat de càlculs que requereixen els processos de classificació.

Actualment, la base matemàtica d'aquesta àrea de l'Estadística encara està poc desenvolupada, i no permet elegir un algorisme amb avantatges indiscutibles sobre la resta. És per això que en els darrers anys hi ha hagut intents de formular models matemàtics més precissos i rigorosos de l'anàlisi de

classificació, entre els quals són de destacar els treballs de Wolfe [WOLF70] o Jardine & Sibson [JARD71].

Ja que els fonaments matemàtics dels mètodes de classificació no són molt sòlids, l'estructuració d'un conjunt d'individus en classes requereix un cert procés de raonament. Això situa la classificació dins el camp d'una altra àrea de coneixement, que també la fa objecte del seu estudi: la Intel·ligència Artificial.

De fet, diversos sistemes experts famosos, MYCIN[SHORT76], INTERNIST, PLANT/ds[MICH82], no són més que classificadors. Donada una certa entrada, es determina a quina classe pertany segons un cert subconjunt de les seves característiques, i les regles corresponents implementades en forma d'arbre de decisió.

Sota una aproximació clàssica, és l'expert el responsable de decidir quines característiques dels individus jugaran en la classificació i, per tant, en la formulació de regles. La limitació que aquest enfoc presenta, ja assenyalada per alguns autors (ex: [CLAN81], [HAYE84], [BAIM88],...), ha estimulat el desenvolupament de sistemes que poden decidir quan un atribut és potencialment útil o no, alliberant d'aquesta responsabilitat a l'usuari. Aquesta selecció d'atributs rellevants s'ha fet mitjançant mètodes d'inferència inductiva automàtica, que posin de manifest les estructures o relacions existents entre les dades, que poden intervenir en la posterior formulació de les regles de classificació. Aquesta aproximació també s'utilitza en el reconeixement de formes (**pattern recognition**), combinant-la, essencialment, amb mètodes estadístics [LÓPE81].

En l'àrea de l'aprenentatge automàtic (**machine learning**), s'han desenvolupat diversos mètodes *heurístics* d'avaluació d'atributs en termes de la seva utilitat potencial (identificada amb la seva rellevància) que permeten decidir a quin grup pertany cada un, basant-se, en la seva majoria, en la teoria de la informació clàssica (ex: [QUIN82], [SCHL86], [UTGO87], [BAIM88], [VELD89], i [LÓPE90]).

La classificació ha estat, doncs, àmpliament estudiada per diferents disciplines des d'èpoques remotes. Aquest treball neix de la idea de posar en contacte idees estadístiques i inductives tractant de compaginar, en la mesura del possible, ambdós punts de vista.

1.3 Situació actual

Donada una mostra per classificar, la primera intenció que hom té és d'estudiar totes les classificacions possibles i triar la millor, en base a algun criteri que mesuri com s'ajusta la jerarquia obtinguda a les dades. De fet, donat un grup d'individus, el conjunt de tots els arbres binaris que es puguin construir amb aquests individus com a nodes terminals defineix un espai de cerca que conté la classificació òptima. Però la cerca de l'òptim en aquest espai es submergeix en el món de la complexitat algorísmica, on, naturalment, naufraga, ja que el nombre de jerarquies possibles esdevé intractable quan el nombre d'individus a classificar encara no és gaire gran.

No podent fer un estudi exhaustiu, es recorre als criteris heurístics, que redueixen aquest espai de cerca. Disminuint el nombre de jerarquies que s'estudiaran, es genera molt més ràpidament una solució, que, en contrapartida, pot no ser l'òptima, encara que sí es pot assegurar que sigui *raonable*, és a dir, suficientment propera a l'òptim real com per que no valgui la pena invertir més esforç computacional en calcular-lo.

Actualment podem agrupar els heurístics existents en tres grans blocs:

Mètodes de particions: Cerquen una partició òptima del conjunt que s'estudia en un nombre prefixat de classes k. En tenim de dos tipus també:

- **Mètodes de particions directes:** Les classes que es formin seran disjuntes.
- **Mètodes de particions en classes solapades:** Les classes poden solapar-se, com el seu nom indica. És a dir, un mateix objecte pot pertànyer simultàniament a més d'una classe.

Mètodes de classificació jeràrquica: Busquen l'arbre que reflexa l'estructura jeràrquica de les dades. Segons el nivell pel que es talli l'arbre obtindrem una partició més o menys precissa del conjunt objecte d'estudi. L'avantatge d'aquest grup respecte de l'anterior és que no cal avançar el nombre de classes que es volen obtenir al final, sinó que un cop construïda la taxonomia es pot decidir a quin nivell la tallem horitzontalment, obtenint així

una partició del refinament més adient als objectius de l'estudi. Principalment hi ha dues tècniques:

- **Classificació ascendent jeràrquica:** Es construeixen les jerarquies fent agregacions successives dels objectes — i després dels grups formats—, en funció de les distàncies entre objectes — i eventualment grups—.
- **Classificació descendent jeràrquica:** Opera en sentit invers a l'anterior en base a diferents dicotomies. El conjunt total dels individus es comença per escindir en dos, i es procedeix subdividint cada part.

Mètodes de conjunts difusos: Es centren en la detecció de les zones de forta densitat del núvol de punts. Portant una mica a l'extrem la filosofia de la classificació jeràrquica, esbrinen per si mateixos el nombre més adient de classes.

És a dir, mentre que en el primer grup tota la responsabilitat recau sobre l'investigador, que ha de decidir quantes classes vol, en el segon, si bé també ha de fer-ho, el mètode li proporciona informació sobre com són les dades que l'ajudi a prendre aquesta decisió. En el tercer grup, les pròpies dades *parlen*, indicant en quines classes es volen agrupar.

Donada la feblesa de bases teòriques de TOTS aquests algorismes, hom no hauria de fiar-se totalment dels resultats de qualsevol d'aquests processos, sinó que s'hauria de fer una validació *a posteriori* de la qualitat de la classificació obtinguda.

1.4 Estructura del treball

L'objectiu d'aquest treball es centra en oferir una eina de validació dels resultats obtinguts en la classificació de LINNEO [MART90], programa que efectua una classificació heurística basant-se en tècniques d'Intel·ligència Artificial. KLASS, el classificador que ocupa aquest treball, empra únicament criteris estadístics. Es tractarà, doncs, de confrontar els resultats obtinguts per un i altre. És d'esperar que KLASS sigui més lent que LINNEO, donat l'elevat nombre de càlculs que comporta el procediment que implementa, a canvi de proporcionar resultats estadístics més solvents.

L'organització del treball es detalla tot seguit. El capítol que obre aquesta obra és un capítol introductori, en el que s'ofereix una ràpida visió del que constitueix la classificació de dades.

El segon capítol conté el desenvolupament teòric que serveix de base per KLASS, que és un classificador ascendent jeràrquic per a dades descrites sobre espais de variables de qualsevol naturalesa, tant contínues com categòriques.

El tercer capítol està dedicat a la implementació. En ell es fa referència al disseny de l'algorisme a nivell funcional, així com als aspectes més interessants del nivell intern.

Tot seguit, es presenten alguns bancs de dades que s'han utilitzat per a provar i validar el classificador, i s'analitzen els resultats obtinguts en l'execució del programa, efectuant la comparació, abans esmentada, amb els resultats que, per a les mateixes dades, proporciona LINNEO.

Per acabar hi ha un apartat de conclusions, i les directrius del treball que queda per fer com a continuació d'aquest estudi.

En els apèndixs s'hi poden trobar els llistats tant dels fitxers de dades usats com a prova com de les parts més rellevants del codi del classificador.

Capítol 2

Desenvolupament teòric de KLASS

"És impossible que d'éssers contraris
se segueixi concordància alguna".
R.LLULL

2.1 Introducció

Aquest treball es centra en l'estudi de tècniques que aprofitin el coneixement que els experts tenen sobre la classificació estadística de grans bancs de dades, amb l'objectiu de facilitar tant com es pugui aquest tipus de procediments a l'usuari.

Una de les majors dificultats que presenten els mètodes de classificació és que el desconeixement de l'estructura real de les dades fa impossible una valoració de la qualitat de la classificació obtinguda, ja que no es pot, *a priori*, comparar amb res. Per tant, el que es fa és *interpretar* els grups resultants, i a partir d'aquesta interpretació, validar les classes. Es tractarà aquí de minimitzar la feina que correspongui a l'usuari i el nivell d'experiència requerit per a treballar.

L'objectiu de la classificació de dades és, partint d'una mostra d'individus pertanyents a una població (o poblacions) de la qual se'n desconeix l'estructura, formar grups el màxim de diferents entre si amb individus el màxim d'homogenis, segons les variables que els descriuen.

La informació inicial que es disposarà serà, precisament, la descripció d'una mostra d'individus d'una certa població en termes d'un conjunt de variables de

naturaleses diverses (veure 2.2), que se sol representar en forma de matriu ($\mathcal{M}$).

S'ha utilitzat un mètode de classificació ascendent jeràrquica de les dades que consisteix a classificar les pròpies classes d'individus i repetir el procés fins formar un arbre binari. Amb això, hom tindrà classificacions del mateix collectiu a diferents nivells d'abstracció, i es podrà elegir la que més s'ajusti als propòsits de l'expert.

Successives fusions[1] d'individus o subgrups en noves classes generaran l'arbre de classificació. Aquest procés requereix, però, que s'especifiqui com determinar les entitats a fusionar en cada pas, així com la forma d'efectuar aquesta fusió.

L'algorisme dels **veïns recíprocs** utilitza un concepte propi per a determinar els individus que s'agreguen:

> S'anomena **veïns recíprocs** als individus i, i' si i és l'objecte més proper a i' en la mostra donada, i i' és també el més proper a i.

En aquest algorisme doncs, sempre s'agreguen parelles de veïns recíprocs, i això requereix la definició d'una mètrica sobre l'espai de les variables que permeti de calcular la distància entre dos individus, per tal d'identificar quines són les parelles d'elements més propers.

Quant al procés d'agregació, hi ha diverses possibilitats, que es detallen en l'apartat 2.4 i que donen lloc a diferents representacions de les classes. És a dir, segons la forma com es faci l'agregació, la característica rellevant d'una classe variarà. Entre altres, podem utilitzar com a descriptor de la classe el seu centre de gravetat, o la distància entre els dos individus més *separats* d'entre els de la classe, o la dels més *propers*, ..., sempre en termes de la mètrica definida.

L'estratègia consisteix a detectar les parelles de veïns recíprocs que es poden fusionar i anar construint l'arbre d'agregacions. Per a la implementació es pot aprofitar el fet que la relació de veïnatge és local, i no queda afectada pel procés d'agregació. Això es contempla en l'algorisme dels **veïns recípocs encadenats**, que d'altra banda és de fàcil tractament recursiu, i per tant, molt adient per a ser implementat en LISP[2].

Aquesta versió parteix d'un objecte de la mostra, busca el seu veí més proper, i va saltant de més proper en més proper fins que detecta dos individus per a

[1] En endavant s'utilitzaran indistintament els termes *fusió* i *agregació*.
[2] S'utilitza el VAX-COMMONLISP 2.2.

agregar. En aquest punt s'efectua la fusió i es torna a classificar a partir del darrer element de la cadena de més propers que s'ha anat construint (d'aquí el nom de l'algorisme), però considerant ara la nova classe com un altre objecte. Els seus components desapareixen com a individus i ja no s'utilitzaran més en la classificació.

Per a fer-ho, resulta molt útil de mantenir una matriu de distàncies entre individus, la gestió de la qual es detalla a l'apartat 3.5.

A continuació es presenta l'esquema algorísmic corresponent als veïns recíprocs encadenats.

```
veins reciprocs(nuvol, element)
    vei:=mes_proper(element)
    vei_reciproc:=mes_proper(vei)
    Si element=vei_reciproc
    llavors
       classe:=agregar(element, vei)
       nuvol_aux:=treure((element, vei), nuvol)
       nuvol':=insertar(classe, nuvol)
       element':=anterior(element)
       veins_reciprocs(nuvol', element')
    sino
       anteriors:=insertar(element, anteriors)
       veins_reciprocs(nuvol, vei)
    fsi
```

La funció **agregar** actualitza totes les estructures d'acord amb la formació d'una classe amb *element* i *vei*.

El que es farà en aquest capítol és, a part de descriure breument quin tipus de dades es tracten, desenvolupar les justificacions teòriques dels criteris d'agregació que s'han implementat, i les diferents mètriques sobre les que es treballa.

2.2 Les variables

Els individus a estudiar estan descrits per una sèrie de variables que defineixen un espai n-dimensional en el qual ubicar-los com a núvol de punts.

Aquestes variables, o descriptors, podran ser **quantitatives o qualitatives**. Les primeres són sempre conceptes mesurables i s'expressen en forma numèrica, bé en nombres reals (p.ex.: alçada) o naturals (p.ex.: edat, nombre d'habitacions d'una vivenda). Les segones, en canvi, corresponen a altres tipus de qualitats dels individus, que s'expressen mitjançant adjectius (p.ex.: mida — petita, mitjana, gran ...—, color, ...). D'entre elles cal destacar les variables **binàries**, que codifiquen amb 0 i 1 l'absència o presència d'una certa propietat respectivament — (p.ex.: referent a una mostra d'animals, es podria considerar la propietat "tenir cua") —.

En el món de l'estadística, on sovint s'ha treballat en FORTRAN, el que es fa per a tractar aquest tipus de dades és un preprocés de recodificació de la informació, que generi una matriu de dades íntegrament numèriques per a estudiar, generalment denotada amb la lletra $\mathcal{X}$.

No obstant, en moltes ocasions resulta més senzill descriure en forma qualitativa els objectes, fins i tot fent referència a conceptes clarament mesurables, que no pas fer-ho de forma numèrica, procés que requereix força més informació, precissió i exactitud. Per exemple, sense saber del cert l'edat d'una persona, hom la podria relacionar ràpidament amb el seu grup d'edat (i.e.: menor, adult, gran), segurament amb un risc d'error molt petit.

És aquest fet, que qualificar resulti més natural per al ser humà que quantificar, que ens porta a estudiar la manera de tractar directament les dades simbòliques, introduint-nos, per aquest motiu, en un entorn LISP.

2.3 Les mètriques

Per a identificar les parelles de veïns recíprocs, cal definir la distància entre dos individus descrits segons una sèrie de variables. S'utilitzen fonamentalment dues mètriques, l'euclídia i χ^2, i es farà en funció del tipus de variables que es considerin.

En aquells casos en que es tinguin J variables contínues, es farà servir la

mètrica euclídia, on la distància entre dos individus i i i' es defineix com

$$d^2(i,i') = \sum_{j=1}^{J}(x_{ij} - x_{i'j})^2$$

Ara bé, quan les variables són d'ordres de magnitud molt diferents, l'efecte de cada una d'elles sobre la distància pot ser molt desigual. El concepte de proximitat entre individus depèn de les unitats de mesura de les variables. Això no hauria de ser així, i el que es fa per a evitar-ho és normalitzar el càlcul de les distàncies dividint per la desviació tipus empírica l'efecte de cada variable:

$$d^2(i,i') = \sum_{j=1}^{J}\left(\frac{x_{ij} - x_{i'j}}{s_j}\right)^2$$

De cares a la implementació, aplicant una petita transformació matemàtica s'obté una expressió en funció de la variància, que es pot calcular directament a partir de les observacions.

$$d^2(i,i') = \sum_{j=1}^{J}\frac{(x_{ij} - x_{i'j})^2}{s_j^2}$$

Si les variables són categòriques, no es poden fer aquests càlculs ja que els elements de la matriu de dades són les diferents modalitats de la variable, i, per tant, són alfanumèrics.

Generalment, en estadística es tracta aquest tipus de variables desdoblant-les en paquets de variables binàries que representen cada una de les seves modalitats, i es mesura la distància entre individus en la mètrica de χ^2.

Així, si la variable k té c categories o valors possibles $k_1, \ldots, k_c$, es transforma en c columnes de la matriu de dades $x_1, \ldots, x_c$, tals que, per a cada individu i

$$x_{ij} = \begin{cases} 1, & \text{si } i_k = k_j \text{ en la matriu original de dades brutes} \\ 0, & \text{altrament} \end{cases}$$

anomenant *taula disjuntiva completa* al resultat.

Després d'aquesta recodificació, es considera x_j el nombre d'individus de la mostra que pertanyen a la modalitat j, x_i la suma de la fila i, i es defineix com a distància entre individus la de χ^2 :

$$d^2(i,i') = \sum_{j=1}^{J}\left(\frac{x_{ij}}{x_i} - \frac{x_{i'j}}{x_{i'}}\right)^2 / x_j \qquad (2.1)$$

Degut al fet que cada individu pertany a una única modalitat per a cada variable, la recodificació que s'ha fet és tal que, per a cada variable categòrica inicial, es genera un grup de columnes de la matriu transformada que contenen un i només un 1 a cada fila, essent nuls els altres elements. Per tant, la fila corresponent a l'individu i tindrà en total tants 1's com variables es consideren, d'on resulta que $x_i = n, \forall i$.

Operant sota aquesta hipòtesi l'expressió 2.1

$$d^2(i,i') = \frac{1}{n^2}\sum_{j=1}^{J}\frac{(x_{ij} - x_{i'j})^2}{x_j}$$

Ara bé, LISP és un llenguatge orientat a la manipulació simbòlica i es pot intentar de treballar directament sobre la matriu sense transformar ($\mathcal{M}$).

Donat que les columnes de la matriu transformada ($\mathcal{X}$) resulten de l'expansió binària de les variables inicials, es poden tractar per blocs de columnes derivades d'una mateixa variable:

$$d^2(i,i') = \frac{1}{n^2}\sum_{k=1}^{K} d_1(i,i',k)$$

on $d_1(i,i',k)$ és la distància en la variable j entre i i i'

$$d_1(i,i',k) = \sum_{i=k_1}^{K_c}\frac{(x_{ij} - x_{i'j})^2}{x_j}$$

Fent un estudi en una única variable, sense pèrdua de generalitat, ens veiem forçats a distingir dos casos:

1. Si i i i' estan en la mateixa categoria es recodificarien, tradicionalment, en el mateix subvector, i per tant $d_1(i,i',k) = 0$.

2. En el cas més general, i i i' tindran l'1 en columnes diferents, sien j i j', i la resta de termes seran 0

   $$d_1(i,i',k) = \frac{(1-0)^2}{x_j} + \frac{(0-1)^2}{x_{j'}} = \frac{1}{x_j} + \frac{1}{x_{j'}}$$

   i aquesta és la *contribució* de la variable k a la distància entre i i i'.

   Per tant,

   $$d_1(i,i',k) = \begin{cases} 0, & \text{si } i_k = i'_k \\ \frac{1}{x_j} + \frac{1}{x_{j'}}, & \text{en cas contrari} \end{cases}$$

Aquesta és la forma en que el programa calcula la distància entre dos individus descrits en un espai de variables qualitatives, i de cares a optimitzar càlculs, es manté una estructura amb les x_j calculades (veure apartat 3.10).

Ara bé, degut al mecanisme d'agregació, apareixen durant el transcurs de l'algorisme nous objectes que no se situen enterament en una modalitat de la variable, sinó distribuïts en diverses d'elles de forma més o menys irregular.

És a dir, l'individu i pot tenir, per a la variable k, un valor del tipus $(f_1, \ldots, f_c)$ on f_j és la proporció sobre la categoria k_j[3].

S'anomena *objectes en forma extesa* als que es representen així i *compactes* als altres. Per a aquest tipus d'objectes, es pot utilitzar la fòrmula general per a calcular distàncies

$$d_1(i,i',k) = \sum_{j=1}^{c} \frac{(f_{ij} - f_{i'j})^2}{x_j} \qquad (2.2)$$

El pas de forma compacte a forma extesa coincideix amb la recodificació que en estadística s'ha vingut fent tradicionalment [ROUX85], esmentada a l'inici d'aquest apartat. La distància entre un objecte en forma compacta i un en forma extesa es calcula extenent la representació del primer i aplicant la fòrmula general 2.2.

Considerant però que el canvi de representació depèn directament del número de categories de la variable i que aquesta és una operació que es pot haver de repetir un nombre força elevat de vegades, val la pena d'implementar directament aquest cas particular:

$$d_1(i,i',k) = \frac{(f_{j'} - 1)^2}{x_{j'}} + \sum_{\substack{j=1 \\ i \neq j'}}^{c} \frac{f_j^2}{x_j}$$

on i està en forma extesa, i' en forma compacta i j' és la columna corresponent a la modalitat d'i' en la variable k. Proves empíriques demostren que la introducció d'aquesta modificació en l'algorisme redueix el temps de CPU considerablement.

Efectivament, l'execució del programa sobre diversos bancs de dades amb o sense aquesta petita optimització genera resultats força diferents. La taula 2.3 ho mostra.

[3]Pel fet que les f_j són proporcions, satisfan que $f_j \geq 0$, $\forall i$, i a més, $\sum_{j=1}^{c} f_j = 1$;.

PROVA	dades	mètrica	criteri	temps de CPU			
				Carregat		Gastat	
				abans	després	abans	després
Compositors	Normalitz.	Euclídia	Centroide	20.33	17.83	1:41.73	36.48
Paisred	No Norm.	χ^2	Centroide i Ward	30.78	34.14	4:31.42	47.42
Paispet	No Norm.	Mixte	Centroide	17.98	19.43	38.32	28.98

Taula 2.1: Taula de temps d'execució.

La primera de les proves fa referència a una matriu de variables de vuit directors d'orquestra, contínues totes elles . Les dues restants són extractes d'una matriu més gran que conté dues variables categòriques i dues de contínues descrivint la situació i tendència político-econòmica d'un nombre considerable de països de tot el món. En la segona només s'estudien les variables categòriques, mentre que en la tercera es mantenen totes, però només es tracten els països del continent Americà.

Queda, per últim, estudiar aquest darrer cas que combina descriptors quantitatius i qualitatius. En el cas mixte, el que es fa és treballar en la mètrica euclídia per als descriptors numèrics i en la de χ^2 per als simbòlics, operant component a component.

Així, si les J_1 primeres columnes de la matriu són les variables quantitatives i les restants qualitatives, es podria definir la distància entre objectes com:

$$d^2(i,i') = \sum_{j=1}^{J_1} (x_{ij} - x_{i'j})^2 + \frac{1}{n^2} \sum_{j=J_1+1}^{J} d_1(i,i',j)$$

i cas que els rangs de les variables numèriques fóssin molt diferents de [0,1] es tindria una forta desproporció entre les influències de la part contínua i categòrica sobre la distància.

És per aquest motiu que, per una banda, es normalitzen les components contínues de la distància, i per una altra, s'introdueixen dos factors que equilibrin la influència d'espai continu i categòric. Si a més es prescindeix de l'ordre en que venen les variables i es denota per C el conjunt de totes les variables contínues que intervenen en la matriu, i per Q el de totes les qualitatives:

$$d^2(i,i') = \alpha \sum_{\forall j \in C} \frac{(x_{ij} - x_{i'j})^2}{s_j^2} + \beta \frac{1}{n^2} \sum_{\forall j \in Q} d_1(i,i',j)$$

A partir d'ara, s'anomenarà *subdistància* a cada una d'aquestes components d'una distància mixta.

Així doncs, es pot expressar la distància mixta com a suma ponderada de les subdistàncies contínua i categòrica.

$$d^2(i,i') = \alpha d_C + \beta d_Q \tag{2.3}$$

Queda per veure com efectuar el càlcul d'α i β, (secció 2.5).

2.4 Criteris d'agregació

En aquest treball s'ha implementat, per una banda, el criteri del centroide, també anomenat del baricentre, que és el que més s'assembla al mètode de classificació heurística implementat en LINNEO, i per altra el criteri de Ward, que té un bon comportament, quant a la qualitat dels resultats que genera fa referència, en aquells casos on tingui sentit ponderar les dades.

2.4.1 El criteri del centroide

Segons el criteri del centroide, el resultat de fusionar dos individus és un nou individu que té com a coordenades les del centre de gravetat dels seus components, que es pot interpretar com a la caracterització promig de la classe en conjunt, i que no té perquè coincidir amb cap element real.

Per tant, per una mostra on només es contemplen variables numèriques, el nou objecte o, que és el resultat d'agregar els individus $i_1,\ldots,i_c$, té com a components $(o_1,\ldots,o_n)$, on, en general

$$o_j = \frac{1}{c}\sum_{i=1}^{c} x_{ij} \tag{2.4}$$

Ara bé, el procés de classificació, encara que va formant classes cada cop més nombroses, mai agrega directament c individus, sinó que ho fa de dos en dos, construint subclasses intermèdies.

Per tant, per a evitar el càlcul directe de centres de gravetat d'un nombre creixent de punts, s'ha desenvolupat un procés incremental, partint dels centres de les subclasses que s'agreguen.

Així doncs, en agregar dues subclasses s_1 i s_2, compostes de c_1 i c_2 individus respectivament[4], els centres de gravetat de les quals són $(s_{11},\ldots,s_{1n})$, i $(s_{21},\ldots,s_{2n})$, es pot demostrar que el centre de gravetat de la classe englobant és, en notació vectorial,

$$o = \frac{c_1 s_1 + c_2 s_2}{c_1 + c_2} \tag{2.5}$$

observant la forma en que prèviament s'havien calculat s_1 i s_2 (2.4).

La implementació d'aquest càlcul en LISP és directe si es disposa del nombre d'individus de cada classe, i aquesta informació es té en la llista *Lnro_nodes* , l'estructura de la qual s'explica en l'apartat 3.1.

Les variables categòriques necessiten aquí un tractament especial, donat que no es poden fer, directament, operacions numèriques amb símbols.

Per una variable categòrica, la manera més natural de caracteritzar un grup d'individus és detallant la forma com es distribueixen en les diferents modalitats de la variable, bé sia mitjançant proporcions[5], o percentatges, o contingents absoluts.

Considerant que en tota classificació jeràrquica es succeeixen nivells creixents d'abstracció, i que en algun moment les subclasses seran tractades com objectes ordinaris, la solució de les proporcions sembla la més adequada, ja que permet fer la següent interpretació:

> *El centre de gravetat d'un conjunt d'individus en un espai de variables qualitatives és una entitat fictícia que no té una modalitat definida per a cada variable, sinó que es reparteix en les diferents categories de cada una de forma proporcional a com ho fan els objectes que el generen.*

Per exemple:

> Suposi's un núvol de punts definit sobre l'espai de les variables FORMA i COLOR, amb les modalitats *(quadrat, triangular, rodó)* i *(blanc, vermell, negre)* respectivament, i dos individus d'aquest

[4] Eventualment, c_1 i/o c_2 podran ser la unitat.

[5] En termes més tècnics, quan parlem de proporcions ens estem referint ni més ni menys que a la distribució marginal de cada variable.

núvol definits de la forma i = *(quadrat,blanc)*, i'= *(quadrat,negre)* el resultat d'agregar i i i' és

o = *((1 quadrat) (0 triangular) (0 rodó), (1/2 blanc) (1/2 negre))*,

és a dir, un objecte quadrat, i aproximadament de color gris. Efectivament, la propietat additiva dels colors és particularment apropiada per a interpretar aquest cas i ofereix una idea molt intuïtiva del significat que té l'agregació de dos individus en termes de variables qualitatives.

Com que el centre de gravetat es calcula component a component, es pot fer l'estudi en una única variable k de categories $k_1, \ldots, k_c$ sense pèrdua alguna de generalitat. L'anterior operació és directament extensible a l'agregació d'$n > 2$ objectes, i, anàlogament a com s'ha fet per al cas continu, s'estudia ara la possibilitat d'aprofitar càlculs intermitjos en la generació de la caracterització de les successives classes.

En el cas més general, i seguint un raonament semblant al del cas continu, l'agregació d's_1 i s_2, subclasses de n_1 i n_2 individus repectivament, i amb centres de gravetat $((f_{11}\,k_1)(f_{12}\,k_2)\ldots(f_{1c}\,k_c))$ i $((f_{21}\,k_1)(f_{22}\,k_2)\ldots(f_{2c}\,k_c))$ és

$$o = ((f_1 k_1)(f_2 k_2)\ldots(f_c k_c)) \text{ , on } f_j = \frac{n_1 f_{1j} + n_2 f_{2j}}{n_1 + n_2}$$

Ara bé, la casuística inherent a aquest tipus d'agregacions permet d'introduir certes optimitzacions en el càlcul del descriptor de la classe.

Per una banda, si s'agreguen dues entitats que rauen en la mateixa modalitat de la variable, no cal construir la distribució marginal, ja que el centre de gravetat descansarà, en la seva totalitat, sobre aquesta mateixa categoria. Així, es pot seguir representant el nou objecte pel símbol de la modalitat en la qual es troba:

$$o = k_j \text{ si } s_1 = s_2 = k_j$$

Per l'altra, si els objectes no són de la mateixa categoria, la distribució resultant únicament tindrà dues modalitats no nul·les, i les freqüències respectives seran:

$$f_{j1} = \frac{n_1}{n_1 + n_2}, \text{ i } \quad f_{j2} = \frac{n_2}{n_1 + n_2}$$

essent k_{jl} la modalitat a la que pertany s_l.

Per últim, si s'agreguen un objecte compacte i un en forma extesa es té que

$$f_j = \begin{cases} \frac{n_1 f_{1j}}{n_1+n_2}, & s_2 \neq k_j \\ \frac{n_1 f_{1j}+n_2}{n_1+n_2}, & s_2 = k_j \end{cases}$$

cas que val la pena implementar, si es considera que serà relativament freqüent, i que estalvia un producte per a cada component i una suma per totes menys 1; més si es pensa que, en definitiva, aquest procés caldrà repetir-lo per a cada una de les variables categòriques que hi hagi.

Per últim, comentar que, en espais on es combinen variables categòriques i contínues, el càlcul es fa component a component aplicant una o altre relació segons el tipus de la variable corresponent.

2.4.2 El criteri de Ward

El criteri de Ward [ROUX85], a diferència de l'anterior, no decideix la fusió de dues classes basant-se en una noció de distància, sinó en la inèrcia de cada una de les classes. Quan menor sigui la inèrcia d'un núvol de punts, més apinyats estaran els seus integrants al voltant del seu centre de gravetat, i per tant, més homogènia serà la classe, perquè els elements seran molt semblants.

La inèrcia d'un conjunt de punts està directament relacionada amb la dispersió d'aquest conjunt respecte del seu centre de gravetat. Sempre que s'afegeixen elements a un núvol de punts augmenta la dispersió d'aquest núvol, i, en aquest sentit, l'agregació de dues subclasses[6] representa un augment en el moment d'inèrcia, que és el que tracta de minimitzar el criteri de Ward. Calculat l'augment de dispersió de totes les possibles fusions, s'agregaran les dues subclasses que enregistrin el menor de tots.

Considerant que l'augment d'inèrcia juga el paper d'una pseudodistància entre classes, a cada pas s'agregaran les dues classes més properes, i això ens situa molt a prop del que s'ha vingut fent fins ara. Des del punt de vista dels veïns recíprocs doncs, no cal més que substituir la matriu de distàncies entre classes per una matriu on s'hi recull l'augment d'inèrcia que es produiria si s'efectués la fusió de cada dues classes, i operar en la forma ordinària.

[6] Que suposa fer un núvol gran de dos de petits, i per tant, amb més punts.

El moment d'inèrcia d'un conjunt de punts $\mathcal{I}$ és el seu moment centrat de segon ordre, o moment respecte del centre de gravetat g, i es calcula com:

$$M^2(\mathcal{I}/g) = \sum_{i \in \mathcal{I}} m_i \, d^2(i,g) \tag{2.6}$$

essent m_i la massa del punt i, entès com l'efectiu de la subclasse en qüestió.

La matriu inicial es construeix considerant que cada punt representa una classe d'un únic individu — i per tant, de massa 1 —. L'element ii' de la matriu representa l'augment de dispersió que es tindria si efectuéssim la fusió dels objectes i i i'. Aplicant la fòrmula (2.6) pel cas $\mathcal{I} = \{i, i'\}$ i massa unitat es té:

$$d^2_{ii'} = M^2(\mathcal{I}, g) = d^2(i,g) + d^2(i',g) \tag{2.7}$$

Per ser g el centre de gravetat d'I, es pot calcular les seves coordenades:

$$\left.\begin{array}{l} i = (x_1 \dots x_n) \\ i' = (x'_1 \dots x'_n) \end{array}\right\} g = (\frac{x_1 + x'_1}{2}, \dots, \frac{x_n + x'_n}{2})$$

En la mètrica euclídia:

$$\begin{aligned} d^2(i,g) &= \sum_{j=1}^{n} \left(x_j - \frac{x_j + x_{j'}}{2} \right)^2 \\ &= \sum_{j=1}^{n} \left(\frac{2x_j - x_j - x_{j'}}{2} \right)^2 \\ &= \sum_{j=1}^{n} \frac{(x_j - x_{j'})^2}{2^2} \\ &= \frac{d^2(i,i')}{4} \end{aligned}$$

Substituint en l'expressió (2.7)

$$d^2_{ii'} = \frac{d^2(i,i')}{4} + \frac{d^2(i,'i)}{4} = 2\,\frac{d^2(i,i')}{4}$$

Per tant, a l'inici de l'algorisme cal fer:

$$\mathcal{D} = (d_{ii'}) \text{ , amb } d^2_{ii'} = \frac{1}{2}\, d^2(i,i')$$

En un primer pas, s'agreguen els punt i, i' que tinguin la menor pseudodistància entre si. En les passes successives, s'identificarà la nova agregació després d'actualitzar la matriu $\mathcal{D}$ en la forma convenient.

En una passa qualsevol, la hipotètica agregació de dos objectes q i q', bé sien individus o subclasses, generaria l'augment de dispersió:

$$d_{qq'} = M^2(q \cup q', g) = m_q d^2(q,g) + m_{q'} d^2(q',g) \tag{2.8}$$

que es pot posar en funció dels centres de gravetat de cada classe i els seus efectius únicament:

$$M^2(q \cup q', g) = \frac{m_q m_{q'}}{m_q + m_{q'}} d^2(q,q')$$

Demostrem-ho: En el cas general, els centres de gravetat de cada classe són de la forma:

$$q = \frac{x_1 + \dots + x_{m_q}}{m_q} \text{ , i } q' = \frac{x'_1 + \dots + x'_{m_{q'}}}{m_{q'}}$$

i el centre de gravetat de la classe englobant seria

$$g = \frac{x_1 + \dots + x_{m_q} + x'_1 + \dots + x'_{m_{q'}}}{m_q + m_{q'}}$$

Calculem $d^2(q,g)$:

$$d^2(q,g) = \sum_{n}^{j=1} \left(\frac{x_{1_j} + \dots + x_{m_q j}}{m_q} - \frac{x_{1_j} + \dots + x_{m_q j} + x'_{1_j} + \dots + x'_{m_{q'} j}}{m_q + m_{q'}} \right)^2 =$$

fent denominador comú:

$$\sum_{n}^{j=1} \left(\frac{1}{m_q(m_q + m_{q'})} \left(\begin{array}{l} m_q(x_{1_j} + \dots + x_{m_q j}) + m_{q'}(x_{1_j} + \dots + x_{m_q j}) - \\ m_q(x_{1_j} + \dots + x_{m_q j}) - m_q(x'_{1_j} + \dots + x'_{m_{q'} j}) \end{array} \right) \right)^2 =$$

factoritzant i anul·lant termes:

$$\left(\frac{1}{m_q(m_q + m_{q'})} \right)^2 \sum_{n}^{j=1} \left(m_{q'}(x_{1_j} + \dots + x_{m_q j}) - m_q(x'_{1_j} + \dots + x'_{m_{q'} j}) \right)^2 =$$

$$\frac{(m_q m_{q'})^2 \, d^2(q,q')}{(m_q\,(m_q + m_{q'}))^2}$$

ja que la distància entre els centres de les dues classes:

$$\begin{aligned} d^2(q,q') &= \sum_{j=1}^{n} \left(\frac{x_{1_j} + \dots + x_{m_q j}}{m_q} - \frac{x'_{1_j} + \dots + x'_{m_{q'} j}}{m_{q'}} \right)^2 \\ &= \sum_{j=1}^{n} \left(\frac{m_{q'}(x_{1_j} + \dots + x_{m_q j}) - m_q(x'_{1_j} + \dots + x'_{m_{q'} j})}{m_q + m_{q'}} \right)^2 \end{aligned}$$

Així doncs,

$$d^2(q,g) = \frac{m_{q'}^2\, d^2(q,q')}{(m_q + m_{q'})^2}$$

, i de forma anàloga (2.9)

$$d^2(q',g) = \frac{m_q^2\, d^2(q,q')}{(m_q + m_{q'})^2}$$

Substituint a (2.8) les expressions (2.9)

$$\begin{aligned} M^2(q \cup q', g) &= m_q \frac{m_{q'}^2\, d^2(q,q')}{(m_q + m_{q'})^2} + m_{q'} \frac{m_q^2\, d^2(q,q')}{(m_q + m_{q'})^2} \\ &= \frac{d^2(q,q')}{(m_q + m_{q'})^2} (m_q m_{q'}^2 + m_{q'} m_q^2) \\ &= \frac{d^2(q,q')}{(m_q + m_{q'})^2} m_q m_{q'} (m_{q'} + m_q) \end{aligned}$$

d'on, finalment, s'obté

$$d_{qq'} = M^2(q \cup q', g) = \frac{m_q m_{q'}}{(m_q + m_{q'})} d^2(q,q') \quad \blacksquare \qquad (2.10)$$

Ara bé, amb això, es necessita la distància entre les dues classes que s'agreguen per a calcular la nova matriu, la qual cosa implica, o bé mantenir una estructura auxiliar amb les distàncies entre individus, o bé guardar aquestes distàncies i modificar la forma de calcular l'element més proper d'un donat, que ja no és el mínim de la corresponent fila de la matriu sinó que cal fer els càlculs de l'expressió (2.10). El que realment seria útil des del punt de vista de l'algorisme és una forma recurrent de calcular les noves pseudodistàncies en funció de les actuals, que no utilitzés aquesta distància entre classes que no volem guardar:

En un pas intermedi de l'algorisme, les noves pseudodistàncies que cal calcular són les corresponents a la darrera classe que s'ha format — suposi's $\mathcal{I} = \{i, i'\}$ —, amb cada un dels objectes restants del núvol de punts — sia k—. Partint de l'expressió (2.10):

$$d_{\mathcal{I}k} = \frac{m_{\mathcal{I}} m_k}{(m_{\mathcal{I}} + m_k)} d^2(\mathcal{I}, k) = \frac{(m_i + m_{i'}) m_k}{(m_i + m_{i'} + m_k)} d^2(\mathcal{I}, k)$$

Conegudes les coordenades del centre de gravetat de la nova classe (expressió 2.5) i denotant per m el terme $m_i + m_{i'} + m_k$,

$$d_{\mathcal{I}k} = \frac{(m_i + m_{i'})\, m_k}{m} \sum_{j=1}^{n} \left(\frac{m_i i_j + m_{i'} i'_j}{m_i + m_{i'}} - k_j \right)^2$$

fent denominador comú i factoritzant

$$d_{\mathcal{I}k} = \frac{(m_i + m_{i'})\, m_k}{m(m_i + m_{i'})^2} \sum_{j=1}^{n} (m_i i_j + m_{i'} i'_j - m_i k_j - m_{i'} k_j)^2$$

Reagrupant termes i desenvolupant el quadrat

$$d_{\mathcal{I}k} = \frac{m_k}{m(m_i + m_{i'})} \sum_{j=1}^{n} \left(m_i^2 (i_j - k_j)^2 + m_{i'}^2 (i'_j - k_j)^2 + 2 m_i m_{i'} (i_j - k_j)(i'_j - k_j) \right)$$

Sumant i restant els termes $m_i m_{i'}(i-k)^2$ i $m_i m_{i'}(i'-k)^2$ en l'interior del sumatori

$$\begin{aligned} d_{\mathcal{I}k} &= \frac{m_k}{m(m_i + m_{i'})} \sum_{j=1}^{n} (m_i (m_i + m_{i'})(i_j - k_j)^2 + m_{i'} (m_i + m_{i'})(i'_j - k_j)^2 \\ &\quad + 2 m_i m_{i'} (2 i_j i'_j - i_j k_j - i'_j k_j + k_j^2) \\ &\quad - m_i m_{i'} (i_j^2 - 2 i_j k_j + k_j^2 + i_j^2 - 2 i'_j k_j + k_j^2)) \end{aligned}$$

Distribuint els sumatoris convenientment

$$\begin{aligned} d_{\mathcal{I}k} &= \frac{m_k}{m(m_i + m_{i'})} [m_i (m_i + m_{i'})\, d(i,k)^2 + m_{i'} (m_i + m_{i'})\, d(i',k)^2 + \\ &\quad \sum_{j=1}^{n} m_i m_{i'} (2 i_j i'_j - 2 i_j k_j - 2 i'_j k_j + 2 k_j^2 - i_j^2 + 2 i_j k_j - 2 k_j^2 - i_j^2 + 2 i'_j k_j)] \\ &= \frac{m_k}{m(m_i + m_{i'})} \begin{pmatrix} m_i (m_i + m_{i'}) d(i,k)^2 \\ + m_{i'} (m_i + m_{i'}) d(i',k)^2 \\ - m_i m_{i'} d(i,i')^2 \end{pmatrix} \end{aligned}$$

Fent factor comú i simplificant

$$d_{\mathcal{I}k} = \frac{1}{m} \left(m_i m_k d(i,k)^2 + m_{i'} m_k d(i',k)^2 - m_k \frac{m_i m_{i'}}{m_i + m_{i'}} d(i,i')^2 \right)$$

Multiplicant i dividint per factors constants

$$= \frac{1}{m} \begin{pmatrix} (m_i + m_k) \frac{m_i m_k}{m_i + m_k} d(i,k)^2 \\ + (m_{i'} + m_k) \frac{m_{i'} m_k}{m_{i'} + m_k} d(i',k)^2 \\ - m_k \frac{m_i m_{i'}}{m_i + m_{i'}} d(i,i')^2 \end{pmatrix}$$

i s'obté la fòrmula recurrent buscada:

$$d_{\mathcal{I}k} = \frac{1}{m} \begin{pmatrix} (m_i + m_k) d_{ik} \\ + (m_{i'} + m_k) d_{i'k} \\ - m_k d_{ii'} \end{pmatrix}$$

Amb aquesta relació s'actualitzaran les pseudodistàncies d'un pas a l'altre, i a la matriu s'hi tenen les quantitats que ens determinen *directament* quins nodes agregar en el següent pas.

2.5 Càlcul de les constants α i β de ponderació de les distàncies

Aquests pesos estan destinats a equilibrar les components qualitativa i quantitativa de la distància entre dos individus (veure 2.3). És a dir, interessa multiplicar per un factor gran distàncies petites i vice-versa. Es tracta, doncs, de definir α i β de tal forma que siguin *inversament* proporcionals al rang de la subdistància que multipliquen.

Sia $\mathcal{D}_\mathcal{C} = (d_{C_{ij}})$ la matriu de subdistàncies contínues entre els individus a estudiar, i $\mathcal{D}_\mathcal{Q} = (d_{Q_{ij}})$ la de subdistàncies qualitatives. Llavors

$$d_{C_{ij}} = \sum_{\forall k \in \mathcal{C}} d_1(i,j,k) \;\; ; \;\; d_{Q_{ij}} = \sum_{\forall k \in \mathcal{Q}} (x_{ik} - x_{jk})^2$$

Prenent un element de $\mathcal{D}_\mathcal{C}$ — sia d_C—, i un de $\mathcal{D}_\mathcal{Q}$ — sia d_Q— es pot definir

$$\alpha = f(\frac{1}{d_C}) \text{ , i } \beta = f(\frac{1}{d_Q}) \tag{2.11}$$

En principi, d_C i d_Q podrien ser els màxims elements, en valor absolut, de les repectives matrius, però això no seria massa robust respecte dels valors aberrants, i podria generar justament el fenòmen que es tracta d'evitar, és a dir, un efecte descompensat de les diferents components[7]. Volent treballar a un nivell de fiabilitat del 95%, no es triarà el màxim element de cada matriu, sinó que es menyspreará almenys el 5% dels elements de més gran valor absolut d'aquestes matrius.

És a dir, que calculades $\mathcal{D}_\mathcal{V}, \mathcal{V} \in \{\mathcal{Q},\mathcal{C}\}$, s'ordenaran els seus elements de major a menor en dos vectors $\mathcal{V}_\mathcal{V}$ respectivament, i es triarà com a d_V l'element que ocupi la $i-sima$ posició d'aquest vector, essent i tal que deixa el 5% del total d'elements de $\mathcal{V}_\mathcal{V}$ a l'esquerra:

$$d_V = \mathcal{V}_\mathcal{V}(i) \text{ , tq } i = \left\lfloor 0.05 \tfrac{n(n-1)}{2} \right\rfloor$$

ja que $\mathcal{D}_\mathcal{V}$ té $\frac{n(n-1)}{2}$ elements, n és el nombre d'individus a estudiar, i indexem $\mathcal{V}_\mathcal{V}$ des de 0.

[7]Efectivament, la distància de tot valor aberrant a qualsevol altre objecte de la mostra serà anormalment gran respecte al conjunt, i produïrà uns valors ponderats extremadament petits per a la resta d'elements de la matriu.

Un segon aspecte a considerar és que potser no sigui molt adient equilibrar ambdues components de les distàncies (fent f la identitat) perquè la importància d'un i altre bloc de variables pugui no ser la mateixa. És a dir, portant-ho a l'extrem, si, per exemple, tenim moltes més variables qualitatives que quantitatives, sembla raonable que el terme qualitatiu intervingui amb més força en la distància, ja que està representant la major part de la descripció dels individus. Segurament, dos individus s'assemblaran més si tenen properes la quasi totalitat de les seves variables.

Per això, definim α i β proporcionalment a la dimensió del subespai al que estan associades. I per tant, essent $n_V = card(V)$

$$f(x_V) = n_V x_V \tag{2.12}$$

De les expressions (2.11) i (2.12) es desprèn la forma de les constants de ponderació, tal com s'utilitzaran en l'algorisme per a equilibrar les subdistàncies:

$$\alpha = \frac{n_C}{d_C} \text{ , i } \beta = \frac{n_Q}{d_Q} \tag{2.13}$$

2.6 Tractament de valors mancants

Els valors mancants són observacions desconegudes i representen sempre una dificultat a l'hora de fer una anàlisi, ja que són font de files incomplertes de la matriu de dades, i no es poden tractar directament per falta d'informació.

Quan una fila té un valor mancant, les alternatives són múltiples, entre d'altres solucions hi ha:

- Obviar totes aquelles files de la matriu de dades que contenen algun valor mancant, amb la conseqüent pèrdua d'informació.
- Recodificar tots els valors mancants a alguna constant tractable des del punt de vista de l'algorisme.
- Substituir els valors mancants per algun valor que no pertorbi la columna, estadísticament parlant.

KLASS substitueix els valors mancants de les variables quantitatives pel seu valor mig. En particular, si es treballa amb variables normalitzades segons la

mètrica dels rangs, es substitueixen per 0.5. Aquest és el valor que utilitza LINNEO en qualsevol cas, i que coincideix amb el valor mig de la variable si aquesta ha estat reescalada a l'interval [0,1], la qual cosa s'assoleix dividint precisament per la inversa dels rangs.

Quant a les variables qualitatives, una solució ràpida consisteix a substituir el valor mancant per la modalitat central de la variable. És a dir, si tenim una variable d'n modalitats $(m_1, \ldots, m_k)$, els valors perduts són substituïts per $m_{\lfloor \frac{1+k}{2} \rfloor}$.

A l'apartat de conclusions, on també es parla de les línies obertes d'aquest treball es proposa un tractament alternatiu de valors mancants que s'està implementant per a un futur immediat.

KLASS interpreta una observació com a dada mancant, si el seu valor és '-1' ó '?'.

Un cop revisades les qüestions de caire més teòric, es pot passar a l'estudi de l'algorisme que les implementa.

Capítol 3

Disseny i implementació

Aquest capítol parla del disseny de KLASS profunditzant en l'estructura i particularitat de cada mòdul en el que el programa s'organitza.

3.1 Representació interna de la informació

El programa ha estat dissenyat per a complementar un entorn ja existent basat en classificació heurística, anomenat LINNEO [MART90] i que s'ha desenvolupat en el departament de Llenguatges i Sistemes Informàtics de la Universitat Politècnica de Catalunya. LINNEO és una eina d'ajut en tasques d'adquisició de coneixements en l'entorn dels sistemes experts, i estructura les dades en els següents fitxers:

<nom_fitxer>.dat Conté la matriu de dades per files. Per a cada objecte hi ha una llista amb les coordenades que el defineixen en cada variable.

<nom_fitxer>.pro Conté metainformació referent a les variables, a les *propietats* segons les quals s'ha descrit els individus. Per a cada variable es té, apart del seu nom i un índex numèric associat, les següents dades:

1. Tipus de la variable:
   - **Q** : Qualitativa
   - **C** : Contínua
2. Pes: Per a estudis en els que calgui ponderar les propietats.

3. Nombre d'objectes que comparteixen aquesta propietat. Això permet treballar amb conjunts de propietats que no necessàriament han de ser aplicables a tots els individus de la mostra objecte d'estudi. Per exemple, tractant persones, no té sentit el nombre de fills d'un nen, i aquesta seria una variable a estudiar únicament sobre un subconjunt dels individus.

A més, per a cada variable

qualitativa es té la llista de modalitats en que es particionen els individus.

quantitativa es té el rang en que es mouen els valors de la variable.

Les variables apareixen ordenades alfabèticament.

<nom_fitxer>.obj Caracterització dels objectes de la mostra. Com en el cas de les propietats, hi ha un identificador de l'objecte, i un índex associat, així com també la llista de propietats que el descriuen, que sempre és un subconjunt de les que formen la matriu de dades.

Es disposa d'una interfície que carrega aquests fitxers creant les estructures i símbols necessaris , i l'entorn resultant és el punt de partida d'aquest programa.

Fonamentalment, el que s'obté de la càrrega és la matriu de dades i dues llistes on hi ha la informació associada a les variables i objectes respectivament (i.e. *l_obj_creados* i *l_props_defs*).

El classificador treballarà amb les següents estructures globals:

- Estructures estàtiques, que no es modifiquen en tot el procés, i són útils per a fer recorreguts de la matriu, bé sia per files o columnes:

  l_objectes: Llista dels objectes de la mostra.

  l_propietats: Llista de variables que s'estudien.

- Estructures dinàmiques que contenen la informació necessària per fer l'anàlisi i agilitzar càlculs:

  l_nro_nodes: Estructura de fàcil gestió que dóna el número d'individus que s'agrupen en cada una de les classes. S'inicialitza amb els objectes de la mostra, màxim nivell de desagregació possible, on cada objecte

representa una classe d'un únic integrant. En fer una fusió, el nombre de nodes de la nova classe és la suma del de llurs fills.

l_mes_propers: Conté l'element més proper de cada objecte. S'inicialitza amb tripletes de la forma (i, i', ∞) i es modifica tal com es detalla en (3.7) cada cop que s'actualitzen distàncies.

l_efectius: Nombre d'objectes pertanyents a cada modalitat per a cada variable categòrica. La seva gestió es duu a terme com s'indica a l'apartat (3.10).

matriu: Conté les coordenades de cada individu en l'espai de les variables objecte d'estudi. Inicialment té n files útils, però a mesura que s'avança en la classificació apareixen noves classes, el centre de gravetat de les quals passa a engrandir les files d'aquesta matriu[1]. Com que es forma un arbre binari a partir dels objectes de la matriu inicial (n-dimensional), es tindran en total $\frac{n(n-1)}{2}$ nodes (entre interns i terminals), i aquesta és la dimensió de *Matriu*, que es pot definir de forma exacta gràcies a la gestió dinàmica que LISP fa de l'espai.

matriu de distàncies: de la qual se'n parla àmpliament als apartats (3.5) i (3.6).

A més a més, a cada variable quantitativa se li associarà la seva mitjana i variància, a cada objecte, la posició que ocupa en la matriu de dades, per tal de facilitar l'accés, així com l'índex de nivell (veure 1.1) que li correspon, per a representar després la jerarquia, i si s'està en un cas mixte, amb variables d'ambdós tipus, es crearà a part α, β, i les dues matrius de subdistàncies numèrica i categòrica, de les quals se'n parla a l'apartat (3.4).

Es pot considerar que el programa KLASS té dos subprocessos fonamentals:

- Un de cerca de veïns recíprocs, que es basa principalment en la consulta de la llista de més propers, i

- Un d'agregació dels nodes identificats en el procés anterior, que és, en definitiva, una fase de modificació i actualització d'estructures.

[1] Es conserva la informació referent a totes les classes intermèdies perquè és *a posteriori* que es decideix el nivell de refinament de la classificació definitiva.

Així doncs, definides les inicialitzacions de les variables i els mecanismes d'actualització davant d'una agregació queda totalment determinat el comportament d'aquestes estructures al llarg de tot el procés.

El resultat del programa és l'arbre de classificació jeràrquica, que no es contrueix explícitament, sinó que es disposa de tota la informació necessària per a representar-lo i estudiar-lo en les següents estructures:

l_obj_creados: Conté la llista de tots els nodes de l'arbre començant per l'arrel en l'ordre invers a com s'han anat creant els nodes interns. Els darrers elements de la llista són les fulles.

matriu: Conté les coordenades dels centres de gravetat de cada classe en l'ordre com apareixen a *l_obj_creados*, en el benentès que el centre de gravetat d'una classe formada per un únic punt s'indentifica directament amb les coordenades d'aquest punt.

Cada classe queda representada per un símbol al qual s'ha associat:

1. Una llista amb els identificadors dels dos fills de la classe.
2. La distància entre els seus dos fills en la mètrica corresponent. Aquesta quantitat està directament relacionada amb l'índex de nivell de cada node, i la longitud de les branques que uneixen un pare als seus fills serà proporcional a aquest índex de nivell.

El següent pas consisteix a la visualització d'aquest arbre que es pot fer via algun preprocés que calculi les coordenades de cada element d'*l_obj_creados* en un espai rectangular, de dimensions $[0, index_arrel]$x$[1, nro_individus]$, i després els ordeni per files, i columnes en cada fila començant per les fulles, reajustant el rectangle a la pantalla, o a les dimensions del suport sobre el que s'hagi de representar l'arbre.

Les coordenades d'un punt són funció del seu índex de nivell per la component horitzontal, i del nombre de fills que té per la vertical. Aquesta darrera quantitat es pot accedir via la llista *l_nro_nodes* de forma directa, a partir de l'identificador del node, i la primera està directament associada a cada node, via la seva llista de propietats.

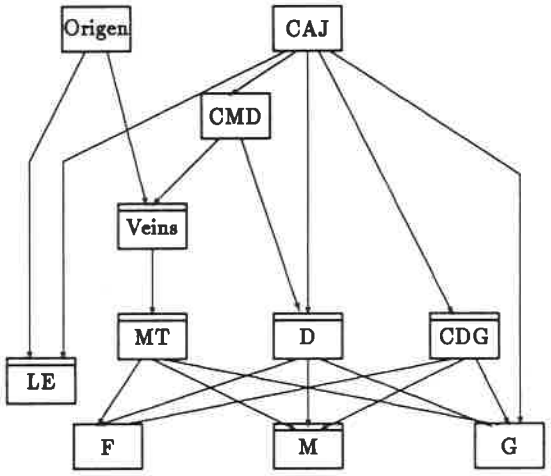


Figura 3.1: Diagrama de mòduls

3.2 Disseny funcional

El programa s'ha estructurat en una sèrie de mòduls de funcions més o menys específiques, alguns dels quals contenen estructures de dades pròpies i d'altres no. No obstant, pels mòduls amb TAD's, l'accés a l'estructura no es realitza, en alguns casos, mitjançant interfície funcional, sinó accedint directament a la implementació. Hi intervenen en aquesta decisió raons òbvies d'eficiència i la consideració que, si aquest programa està orientat a tractar grans quantitats d'informació i és recursiu, convé no carregar la pila amb crides a funcions trivials que, d'altra banda, són molt utilitzades, i a més retarden el procés de l'execució.

En la figura 3.1 es presenta un diagrama dels mòduls que componen el programa. Mòduls les especificacions dels quals apareixen en els apartats subsegüents d'aquest texte. Els mòduls en qüestió són els següents:

Origen: Origen (veure 3.4).

Caj: Classificació Ascendent Jeràrquica (veure 3.3).

CMD: Construir Matriu de Distàncies (veure 3.6).

Veïns: Veïns(veure 3.7).

MT: Matriu Triangular de Distàncies (veure 3.5).

D: Distàncies (veure 3.8).

CDG: Centre de Gravetat (veure 3.9).

LE: Llista d'Efectius (veure 3.10).

F: Fòrmules (veure 3.12).

M: Matriu (veure 3.11).

G: General (veure 3.13).

A part, s'han definit una funció que calcula el valor de **metrica** explorant el tipus de variables que tenim, i una altra per a recórrer l'arbre jeràrquic proporcionant la informació rellevant de cada node.

A grans trets es té un mòdul d'inicialització, un amb l'algorisme general, i els TAD's corresponents a les principals estructures de la implementació, fent ús, quasi totes elles de funcions de caire més general, agrupades en els dos mòduls inferiors a modus de biblioteca.

3.3 Caj: Classificació Ascendent Jeràrquica.

Representa el cos d'aquest treball. És el mòdul que dirigeix, per a dir-ho d'alguna manera, el fluxe de l'algorisme de classificació.

3.3.1 Interfície

La funció de crida és

(veins_reciprocs llista td alfa beta criteri)

i la semàntica dels paràmetres es detalla tot seguit:

llista és la llista d'objectes a classificar

td és la mètrica segons la qual calcular les distàncies entre individus. Hi ha quatre mètriques implementades (2.3), i els valors del paràmetre que designen cada una d'elles són:

- euclidia
- eucl_norm
- chi_2
- mixte

alfa, beta són opcionals i nominals[2]. Tenen sentit si la mètrica de treball és la mixte, i indiquen els valors amb que cal ponderar les components contínua i categòrica de les distàncies respectivament. El valor que tenen per defecte és 1.

criteri també és opcional i nominal, i indica el criteri d'agregació a seguir en la classificació. Pot prendre dos valors:

- centroide
- ward

i si no s'especifica aquest paràmetre, la classificació s'efectua sota el criteri del centroide.

3.3.2 Precondicions

Aquesta funció suposa l'existència de certes estructures abans de començar, que són les que inicialitza la funció *carregar*, que forma part del mòdul d'inicialitzacions [3.4], i que principalment són:

- La constant *infinit* definida amb un valor real prou gran. El defecte és 99999.
- La matriu d'observacions.
- Dues llistes que permetin indexar aquesta matriu tant per files com per columnes *L_objectes* i *L_propietats*.

[2]Un paràmetre nominal és aquell que, cas que formi part de la crida a la funció, cal especificar-ne el nom. És a dir, no n'hi ha prou amb subministrar el valor de l'argument, sinó que cal prefixar-lo pel seu nom. Per exemple :alfa.

- Les llistes de proximitat, amb el nombre de nodes, i els efectius degudament inicialitzades[3].
- Variàncies i mitjanes de les variables qualitatives calculades.
- Si estem en mètrica mixte, dues submatrius triangulars a partir de les que es podrà construir una matriu de distàncies, i de les que se'n parla a (3.6.5). Una constant que indiqui quant és el 5% de les dades, i les constant α i β per a ponderar les distàncies.

D'acord amb això, el valor habitual del paràmetre **llista** en la crida inicial és *L_objectes*.

3.3.3 Postcondicions

El resultat de fer la classificació és el símbol de l'arrel de l'arbre jeràrquic. Degut a l'estructuració de les dades que s'ha fet, a partir d'aquest únic símbol es pot accedir a tot l'arbre, ja que el valor assignat al propi símbol és la llista dels seus fills. Aquesta és la forma més condensada de representar un arbre, i mitjançant una simple avaluació del node, s'obtenen els fills, en estructura de llista directament, que és la més pròpia per a fer tractaments recursius, seqüencials o exhaustius en LISP. Això ha fet que no s'optés per la implementació clàssica d'arbre binari via *apuntador* a fill dret i esquerra, que requereix un accés individualitzat a cada un dels fills d'un node.

3.3.4 Accés a altres mòduls

L'algorisme de classificació utilitza funcions dels mòduls *Efectius*, *Minims*, *General*, *Matriu*, i *Fòrmules*.

3.3.5 Comentaris generals

A part de la funció principal, que és la que porta el control de l'execució, els dos nuclis bàsics del classificador són els que gestionen l'agregació i la creació d'un nou grup.

[3]Contindran únicament la informació trivial referent a cada individu. En el procés de classificació s'hi aniran incorporant les subclasses que es creïn.

El primer vé implementat per una funció que:

- Crea un node. Per a fer-ho, s'utilitza el generador de símbols de LISP, i es crea el símbol **#:(classe-n)** com a identificador del nou grup, on n és un enter que s'assigna seqüencialment a cada classe començant per 1.
- Elimina la informació referent als fills de la matriu de distàncies.
- Calcula la nova fila d'aquesta matriu corresponent al node acabat de crear.

El procés de creació d'un nou node suposa:

- L'associació de la llista dels nodes que s'agreguen al símbol que representa la nova classe.
- La modificació de la llista de proximitats.
- La modificació de la llista amb el nombre d'individus de cada node de la jerarquia. Únicament cal tenir present que el nombre d'individus de la nova classe és $n = n_1 + n_2$ si s'agregaven c_1 i c_2 i n_i és el nombre d'individus de la classe c_i.
- La inserció de la nova fila de la matriu de dades.
- La inserció del nou node a la llista *L_obj_creados*.
- La modificació de la llista d'efectius.

3.4 Origen

Aquest és el mòdul que s'encarrega de la inicialització d'aquelles estructures que es tracten de forma més o menys global en el procés de classificació i de preparar l'entorn que el classificador necessita per a treballar.

3.4.1 Interfície

La funció de crida és

(carregar fitxer modus metrica objectes)

i el significat dels paràmetres és el que s'especifica a continuació:

fitxer És el nom, i eventualment directori, de la família de fitxers (**fitxer.cla**, **fitxer.obj**, **fitxer.pro**), que conté les dades a analitzar i la metainformació associada (3).

modus Indicarà si cal llegir les dades de la matriu amb algun tipus de transformació de normalització o no. De fet, pot ser útil de definir alguna mètrica sobre l'espai dels individus per tal que s'uniformitzi, en certa manera, la importància que cada variable pugui tenir en l'anàlisi. Els valors que pot prendre aquest paràmetre són:

- no : En aquest cas no es fa cap transformació de les dades, sinó que es treballa directament amb les originals.
- continuo: Per aquesta opció, la mètrica que es defineix sobre els individus és la inversa dels rangs, la qual cosa es materialitza en la divisió de totes les observacions pel rang de la variable en qüestió[4]. Aquesta és una solució alternativa a la utilització d'una distància euclídia normalitzada quan totes les variables són quantitatives, amb la salvetat que la distància euclídia normalitzada correspon a definir com a mètrica sobre els individus la inversa de la desviació tipus. Efectivament, igual ens és treballar en la mètrica euclídia normalitzada, on la distància entre individus és $\sum_{\forall j} \frac{(x_{ij}-x_{i'j})^2}{s_j^2}$, que normalitzar les dades, prèviament a l'anàlisi, dividint-les per la seva desviació tipus, i treballar en la mètrica euclídia ordinària, on es calcularien les distàncies de forma equivalent al primer cas com $\sum_{\forall j} \left(\frac{x_{ij}}{s_j} - \frac{x_{i'j}}{s_j}\right)^2$.
- binario: Es transformen totes les variables en binàries en funció d'un cert límit que biparteix les observacions en un grup d'*ítems* inferiors al límit , que es codifiquen amb 0, i un de superiors a ell, que es codifiquen amb el valor 1. El resultat és una matriu de variables 0/1.

metrica Paràmetre opcional amb valor per defecte

[4]Si $max_\mathcal{V}$ i $min_\mathcal{V}$ són, respectivament, els valors màxim i mínim de les observacions per a la variable $\mathcal{V}$, es divideixen els valors d'aquesta variable per $(max_\mathcal{V} - min_\mathcal{V})$.

indiferent

que només és rellevant si té com a valor

mixte

perquè és aquest l'únic cas en que cal calcular les matrius de subdistàncies, i les constants de ponderació α i β.

objectes Paràmetre opcional amb valor per defecte

tots

que indica quins objectes intervindran en la classificació d'entre els que apareixen en els fitxers d'entrada. Permetrà, més endavant, estudiar la influència que un cert objecte té en la classificació.

3.4.2 Precondicions

La crida d'aquesta funció requereix únicament l'existència dels tres fitxers que contenen les definicions i dades que es preten estudiar. Si aquesta família de fitxers, o algun d'ells, no és al directori de treball, cal indicar el camí que condueix al directori que la conté, bé sia en relatiu o absolut[5].

Si l'especificació de la família de fitxers és errònia, no es poden instanciar les dades.

També cal tenir cura d'indicar la mètrica corresponent a les dades que es tenen: Si es vol treballar en la mètrica mixte, cal que el nombre de variables qualitatives sigui no nul, per tal que en dividir per aquest nombre no apareixin errors d'aritmètica. Si es dóna aquest cas, es passarà automàticament a la mètrica euclídia normalitzada.

3.4.3 Postcondicions

Als efectes d'aquest treball, les estructures que crea aquesta funció són les següents:

- S'inicialitzen les constants *infinit* à 99999, i *ntot* amb el nombre de propietats.

[5]Segurament, la possibilitat d'indicar el camí relatiu depengui de la plataforma sobre la que corri el programa, i potser també de la implementació de LISP que s'utilitsi.

- Es prepara el generador de símbols de LISP per tal que les classes que es creïn tinguin una nomenclatura estàndard i seqüencial.
- Es construeix la llista d'objectes, associant a cada objecte l'índex de nivell 0 que es tindrà en la jerarquia per a cada classe individual.
- Es construeix la llista de propietats, associant a cada variable quantitativa la següent informació:
  1. La mitjana aritmètica
  2. La variància
- S'inicialitza la llista *l_nro_nodes* amb un parell de la forma $(i,1)$ per a cada individu i de la mostra.
- S'inicialitza la llista *l_obj_creados*, que més tard serà la font de visualització de l'arbre de classificació.
- S'inicialitzen les llistes *l_mes_proper* i *l_efectius* segons s'indica als respectius mòduls.
- Es construeix la matriu de dades efectuant la transformació expressada pel paràmetre **modus**.
- Si el paràmetre **metrica** té el valor mixte es calculen les constants de ponderació α i β i les matrius de subdistàncies segons el procediment que s'explica més endavant en aquesta mateixa secció.

3.4.4 Accés a altres mòduls

Per a inicialitzar les llistes d'efectius i proximitats es fa accés als mòduls *Veïns* i *Efectius*, i per a carregar les dades i calcular variàncies i mitjanes de les variables s'ha accedit al fitxer *Interfase*.

3.4.5 Comentaris generals

Referent a α i β:

S'havia deduït en l'apartat (2.5) les expressions per a α i β. Quant a la implementació, α i β es poden calcular amb un preprocés a partir de la construcció

paral·lela de les matrius de subdistàncies per una banda ($\mathcal{D}_V$) i els vectors amb les subdistàncies ordenades de major a menor ($\mathcal{V}_V$), tant pel cas continu ($\mathcal{C}$) com pel categòric ($\mathcal{Q}$) , i fent selecció de l'$i-sim$ element de $\mathcal{V}_V$, un cop conegut i. Això suposa la utilització de dos vectors i dues submatrius d'$\frac{n(n-1)}{2}$ elements.

Aquesta és una primera aproximació, on s'està guardant cada distància per duplicat (i.e. en les matrius per una banda, i ordenada per l'altra), la qual cosa representa una ocupació d'espai elevadíssima[6], en termes d'informació totalment redundant.

Com a primer intent de reduir aquest espai, observant que només interessa l'$i-sim$ element dels vectors $\mathcal{V}_V$, no sembla necessari mantenir l'ordenació de tots els elements de la matriu.

De forma anàloga a com, calculant el màxim d'un vector, només es recorda el màxim en curs, en aquest cas només es tindrà memòria dels i màxims elements en curs, a fi de triar el més petit al final, que és el que determinarà el factor de ponderació.

Per una banda, això redueix l'espai a quasi la meitat, ja que enlloc de mantenir dos cops les $\frac{n(n-1)}{2}$ distàncies, es guarden una única vegada, repetint-ne, en el vector d'ordenació corresponent, només el 5%.

Per l'altra, el procés d'ordenació ara és local a un 5% dels elements de la matriu, la qual cosa millorarà molt el temps de cerca, ja que tractem un volum d'informació reduït en un 95%.

I encara més, si enlloc d'ordenar aquest 5% de les distàncies de major a menor, es fa al revés, en calcular una nova distància, es pot esbrinar si pot pertànyer al 5% d'interés o no a través d'una simple consulta del primer element de la llista, que és precisament el que sortirà del vector donat el cas. En detall:

> Sien $(v_1,\ldots,v_n)$ els elements de màxim valor absolut trobats en un cert instant (un 5% de les distàncies calculades fins al moment), i ordenats de menor a major. El càlcul de la nova distància $d_{ii'}$ pot modificar aquest vector únicament si $d_{ii'} > v_1$, i en aquest cas, s'insertarà ordenadament en el vector $(v_2,\ldots,v_n)$, ja que v_1 no pot aportar més informació a l'algorisme i hom en pot prescindir[7].

[6] Cal recordar que, en general, es tindrà grans matrius de dades.

[7] Aquest és un procediment d'implementació LISP molt simple i natural, on només s'utilitza un accés al *car* de la llista i una comparació per a discriminar en quin cas s'està i si cal manipular

En acabar, l'element que interessa per a calcular el factor corresponent es troba ni més ni menys que al cap de la llista, la qual cosa resulta extremadament oportuna. Així doncs, la solució que hom adopta definitivament passa per calcular el nombre d'elements a retenir ordenats ($i = 5\%(n)$), calcular $\mathcal{D}_C$ i $\mathcal{D}_Q$ alhora que s'actualitzen els corresponents $\mathcal{V}_\mathcal{V}$, vectors d'i components únicament, i per últim fer

$$\alpha = \frac{n_C}{\mathcal{V}_C(1)} \text{ , i } \beta = \frac{n_Q}{\mathcal{V}_Q(1)}$$

on recordar que els numeradors representen el nombre de variables numèriques i simbòliques que hi ha respectivament.

L'algorisme generarà doncs, si és el cas, les constants i estructures:

α : Constant de ponderació de les subdistàncies quantitatives.

β : Constant de ponderació de les subdistàncies qualitatives.

d_quant : Matriu de subdistàncies quantitatives ($\mathcal{D}_C$).

d_quali : Matriu de subdistàncies qualitatives ($\mathcal{D}_Q$).

Alliberació d'espai:

Un cop efectuat el càlcul d'α i β, els vectors $\mathcal{V}_\mathcal{V}$ no tenen cap més utilitat, i es podria tractar de recuperar l'espai que ocupen. Si s'assigna NIL a les variables que representen aquests vectors abans d'acabar, quan es dispari el recollector de deixalles (**garbage collector**) de LISP, s'alliberarà l'espai corresponent a $\mathcal{V}_C$ i $\mathcal{V}_Q$ (que vé a ser un 10% dels elements), ja que la reassignació dels símbols convertirà en referències penjades els valors que abans tenien, fent-los susceptibles de recuperació per part del recollector de deixalles.

Aspectes Tècnics de la construcció de certes llistes:

Aprofitant que es construeix la llista d'individus s'associa a cada símbol una sèrie de característiques com a elements de la seva llista de propietats[8]. Amb això l'accés es fa per nom de forma molt ràpida.

la cua de la llista o no.

[8]Les llistes de propietats són també estructures que proporciona el propi LISP, juntament amb les operacions de consulta i modificació dels seus elements.

El que passa és que en tot procés funcional es pot modificar una única estructura en cada crida, i el fet de tractar diverses característiques alhora, a fi d'aprofitar el mateix recorregut seqüencial dels objectes, requereix la introducció d'efectes laterals. Una de les estructures es tracta de forma natural per la pròpia funció (p.ex. la construcció de la llista d'objectes constitueix el cos de la funció **crear_l_objectes**). Les altres es manipulen via efectes laterals, bé a base d'usar la forma LISP **progn**, bé introduint nous paràmetres a les crides de la funció, que si es fan opcionals no afecten a la invocació que d'aquesta funció es faci des de capes superiors del codi (En la funció **crear_l_objectes** hi ha paràmetres d'aquest tipus per a assignar l'índex de posició que s'ha mencionat abans per a cada objecte. Veure apèndix A.).

3.5 Matriu de distàncies

En el procés de classificació per veïns recíprocs les distàncies entre els individus de la mostra juguen un paper fonamental. La forma més intuïtiva d'emmagatzemar aquestes distàncies és mitjançant una matriu quadrada de dimensió n (nombre d'individus a classificar), en la qual d_{ij} representa la distància entre els elements i i j en la mètrica que s'hagi definit. Considerant que molt sovint es tracta amb grans matrius de dades, és a dir, amb n gran, i que la matriu de distàncies tindrà n^2 elements, cal gestionar de forma compacta i eficient el seu càlcul, reduint el temps que l'algorisme inverteix en aquesta tasca i l'espai que ocuparà aquesta matriu.

Com que tota distància és commutativa, per definició de mètrica, aquests tipus de matrius sempre són simètriques, i amb diagonal 0. És per això, que es pot emmagatzemar únicament la part sobretriangular de la matriu. La resta, o bé és informació redundant, o bé es pot tractar implícitament, com és el cas de la diagonal, estalviant així la meitat de l'espai.

S'ha implementat, doncs, la matriu de distàncies com una llista de subllistes que contenen la part sobrediagonal de cada fila i les operacions que facin eficient la manipulació d'aquesta estructura. És a dir, internament, la matriu

$$\mathcal{D} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ \vdots & \ddots & & \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix} \text{ es representa com } \mathcal{D} = \begin{pmatrix} (d_{12} & d_{13} & \dots & d_{1n}) \\ & (d_{23} & \dots & d_{2n}) \\ & & \ddots & \\ & & & (d_{n-1n}) \end{pmatrix}$$

3.5.1 Interfície

En aquest mòdul hi ha operacions orientades a les diferents etapes del procés de classificació, i a les diverses formes de fer-la que determinen els valors d'alguns paràmetres. Les funcions que s'utilitzen fora d'aquest mòdul són:

1. **(Submatrius l_obj Ml Mt)** Crea en paral·lel dues matrius sobretriangulars que contenen respectivament les subdistàncies qualitativa i quantitativa entre els individus d'**l_obj** [2.3 i 2.5]. La funció retorna la qualitativa. L'altra, es tracta via efecte lateral. Quant als paràmetres:

   l_obj És la llista d'objectes que intervindran en la construcció de la matriu.

   Ml, Mt Són la part de matriu creada en el moment de cada crida.

2. **(Eliminar_f_c i m fila nom_fila criteri)** Donada la matriu **m** n'elimina la fila i columna **i**, deixant, eventualment, la fila eliminada al símbol **nom_fila**.

   i Número de la fila que es vol eliminar. La indexació de files i columnes comença en 1.

   matriu Part sobretriangular de la matriu que es vol modificar.

   fila Part de fila que ja s'ha eliminat.

   nom_fila Símbol que contindrà la fila eliminada si s'opera sota el criteri de **Ward**. Si no, no es retorna aquesta fila.

   criteri Indica quin criteri d'agregació s'utilitza [2.4 i 3.3.1].

3. **(pos_matr objecte llista_objectes n)** És una funció d'accés, que retorna la posició en la matriu de distàncies d'**objecte** suposant que s'indexa a partir d'**n** (0 ó 1).

4. **(distancies_agregades element vei llista_obj metrica m_dist)** Retorna la matriu de distàncies corresponent a eliminar del núvol de punts els dos individus que s'agreguen.

   element,vei Individus que formen la nova classe.

   llista_obj Llista d'objectes del núvol, a excepció d'**element** i **vei**.

   metrica Indica la mètrica en que es treballarà.

   m_dist Matriu de distàncies corresponent als individus de **llista_obj**.

5. **(minim fila matriu)** Donat l'índex numèric d'una fila de la matriu de distàncies en calcula el mínim.

   fila Índex de la fila en qüestió.

   matriu Part sobretriangular de la matriu.

3.5.2 Precondicions

Submatrius

Per a aquesta funció cal que existeixi la llista d'objectes a classificar, així com la matriu amb les dades d'aquests objectes.

També han d'existir la llista d'efectius, i una constant amb el nombre de variables qualitatives que intervenen en l'anàlisi, que s'ha de dir **n_quali**, per al càlcul de la matriu qualitativa.

La crida es fa com **(submatrius l_objectes nil nil)**, segons les estructures que la funció *Carregar* genera.

Eliminar_f_c:

Si es fa una classificació sota el criteri de **Ward**, i després d'executar la funció es vol tenir la fila que s'ha eliminat, cal que la crida s'efectüi amb el paràmetre **fila** a **NIL** i que **nom_fila** contingui el símbol que designarà aquesta fila. **M** serà la matriu de distàncies entre individus.

Pos_matr

La precondició d'aquesta funció és l'existència d'una llista d'objectes ordenats segons les files de la matriu de dades. És a dir, si la fila *i* de la matriu correspon a l'objecte *o*, aquest serà l'*i*–sim element d'*l_objectes*.

Com que s'ha suposat que la matriu s'indexava des de 1, la crida serà

(pos_matr objecte l_objectes 1)

Distancies_agregades

Es crida amb certes estructures semi-modificades de forma coherent amb l'agregació que s'està efectuant, perquè únicament és una crida intermitja que facilita l'actualització de la llista de proximitats [3.7]. Per això, en cridar-la, cal que la informació referent als objectes que s'agreguen no figuri ni a la llista d'objectes ni a la matriu de distàncies. Per tant, la crida es fa després d'eliminar fila i columna d'element i vei de la matriu, i els seus identificadors d'l_objectes.

Minim

Pel que fa a la funció mínim cal cridar-la amb la matriu de distàncies com a paràmetre i un número de fila que estigui dins del rang correcte, és a dir, estrictament positiu i per sota del nombre d'objectes a classificar.

3.5.3 Postcondicions

Funció Submatrius:

Les variables *Dquali* i *Dquant* contenen les dues matrius de subdistàncies respectivament. Aquestes matrius han de servir després pel càlcul d'α i β.

Eliminar_f_c

Retorna la matriu sobretriangular resultant d'eliminar la fila i columna i de la matriu m. La creació d'una nova classe durant la classificació significa l'aparició d'un objecte en el núvol de punts, i la desaparició dels seus dos integrants. Per a poder continuar la classificació, s'ha de calcular la distància d'aquest nou objecte a tots els altres. Això significa que la matriu de distàncies haurà d'incloure una fila més, que es pot afegir còmodament pel cap (motiu pel qual no s'ha definit l'operació **insertar_primera_fila**). Però la informació corresponent als dos objectes que s'acaben d'agregar ja no fa cap falta, i és per aquest motiu que es pot eliminar de la matriu qualsevol referència als objectes fusionats. Aquest és el propòsit d'aquesta funció.

Amb el criteri de **Ward**, però, la fila corresponent a la nova classe es calcula recurrentment a partir de les files dels dos elements que la formen, és a dir, les

que es volen eliminar de la matriu. Per això hi ha paràmetres, en la crida a la funció, que permeten generar un símbol amb la fila (o columna, ja que la matriu és simètrica) eliminada, a la qual es tindrà accés posteriorment, per al càlcul de la fila a insertar en la matriu.

De fet, en cada agregació es crida dos cops a aquesta funció, una per cada fill que es fusiona, amb els símbols *fila_elem* i *fila_vei* per a contenir les files corresponents a aquests individus.

Pos_matr

Retorna la posició que ocupa la informació corresponent a objecte en la matriu de dades. S'utilitza cada cop que cal fer un accés a la matriu ja que LISP no té accés simbòlic, sinó per posició, als vectors.

Distancies_agregades

Retorna la mateixa matriu de distàncies que s'ha passat en la crida, i com a efecte lateral, modifica la llista de proximitats deixant-la en coherència amb la desaparició dels dos individus que es fusionen [3.7]. Just després caldrà afegir la informació corresponent al nou objecte tant a la matriu com a la llista d'objectes.

Minim

El resultat de (**minim i M**) és el parell $(o_j\ d_{ij})$ on d_{ij} és el valor mínim de la fila i d'$\mathcal{M}$, i o_j és l'índex en la llista *L_objectes* de l'objecte al qual fa referència la columna (o fila) j de la matriu $\mathcal{M}$.

Aquesta funció es fa servir per a calcular l'element més proper d'i en la matriu $\mathcal{M}$, quan ha desaparegut del núvol de punts l'objecte que jugava aquest paper, per haver-lo agregat amb un altre.

Això motiva que la funció no solament retorni el mínim valor de la fila en qüestió, sinó també informació suficient per a saber quin és l'objecte que ha determinat aquesta distància mínima, de forma que després es pugui actualitzar la llista de proximitats.

3.5.4 Accés a altres mòduls

Al mòdul *Matriu* s'hi accedeix per a obtenir les files de la matriu de dades. S'utilitza també el mòdul *Fòrmules* per a calcular les subdistàncies.

3.5.5 Comentaris generals

Càlcul de les dues matrius de subdistàncies

Comporta també la construcció paral·lela dels vectors *dmaxquali* i *dmaxquant* amb els elements de major valor de la mostra ordenats, en un nombre igual al 5% del total. El cos principal de la funció és un procediment de creació de la matriu qualitativa per files, similar al que s'utilitza en **matriu_distancies** on, per cada subdistància que es calcula s'actualitza, com a efecte lateral, el vector de màxims, en la forma en que s'ha explicat en l'apartat [3.4.5].

De forma absolutament anàloga, però tractada tota ella com a efecte lateral, es desenvolupa la construcció de la matriu quantitativa, que treballa substituint el mecanisme del pas de paràmetres per l'ús de símbols globals auxiliars.

La funció **d_quali_2** crea els símbols *tij* i *lij* amb les subdistàncies quantitativa i qualitativa entre els individus i i j respectivament [2.3].

La funció **fila_quali** construeix les files de les matrius corresponents en *Tij* i *Lij*.

La funció **submatrius** construeix *Dquali* i *Dquant* amb un procediment recursiu pel cap, al igual que **fila_quali**, i no per la cua (com és habitual en LISP), motiu pel qual cal passar com a paràmetre la part de matriu que s'ha construït en cada cas, per tal que al final del procés es pugui fer l'assignació definitiva símbol-valor. Aquests paràmetres s'inicialitzen sempre a NIL. En l'apèndix A s'hi troba el codi referent a aquestes funcions.

Eliminar_f_c:

Per a cada agregació insertem una nova fila a la matriu de distàncies i n'eliminem dues (dues crides a **eliminar_f_c**). Amb això, la matriu de distàncies va reduint les seves dimensions a mesura que es generen noves classes, fins que, al final de la classificació, desapareix. A títol d'il·lustració, suposi's que s'agreguen els

individus i i i' formant o. La matriu de distàncies original

$$\mathcal{D} = \begin{pmatrix} d_{11} & \dots & d_{1i} & \dots & d_{1i'} & \dots & d_{1n} \\ & & & & \vdots & & \\ d_{i1} & \dots & d_{ii} & \dots & d_{ii'} & \dots & d_{in} \\ & & & & \vdots & & \\ d_{i'1} & \dots & d_{i'i} & \dots & d_{i'i'} & \dots & d_{i'n} \\ & & & & \vdots & & \\ d_{n1} & \dots & d_{ni} & \dots & d_{ni'} & \dots & d_{nn} \end{pmatrix}$$

s'ha de transformar en

$$\mathcal{D}' = \begin{pmatrix} d_{o1} & \dots & d_{oi-1} & d_{oi+1} & \dots & d_{oi'-1} & d_{oi'+1} & \dots & d_{on} \\ & & & & & \vdots & & & \\ d_{11} & \dots & d_{1i-1} & d_{1i+1} & \dots & d_{1i'-1} & d_{1i'+1} & \dots & d_{1n} \\ & & & & & \vdots & & & \\ d_{i-11} & \dots & d_{i-1i-1} & d_{i-1i+1} & \dots & d_{i-1i'-1} & d_{i-1i'+1} & \dots & d_{i-1n} \\ d_{i+11} & \dots & d_{i+1i-1} & d_{i+1i+1} & \dots & d_{i+1i'-1} & d_{i+1i'+1} & \dots & d_{i+1n} \\ & & & & & \vdots & & & \\ d_{i'-11} & \dots & & & & & & & \\ d_{i'+11} & \dots & & & & & & & \\ & & & & & \vdots & & & \\ d_{n1} & \dots & d_{ni-1} & d_{ni+1} & \dots & d_{ni'-1} & d_{ni'+1} & \dots & d_{nn} \end{pmatrix}$$

Com que només es té la part so' etriangular de la matriu, hom podria pensar que n'hi ha prou esborrant la fila corresponent a l'individu que *desapareix*. Però també la columna associada s'ha d'eliminar, i per a aconseguir-ho caldrà tenir en compte que l'element m_{ij} ocupa la posició $j - i$ de la i–sima subllista de la matriu que s'ha emmagatzemat[9]. Així doncs, la funció **eliminar_f_c** comença traient l'element $j = i - 1$[10] de la primera fila d'm, per després repetir l'operació amb la cua de la matriu i la columna $j - 1$, fins arribar a la fila i, que desapareix sencera i s'acaba el procés. La submatriu que queda per sota de la fila i no es veu, en absolut, alterada. En la figura 3.5.5 apareixen requadrats els elements que cal eliminar de $\mathcal{D}$.

[9] Considerant que s'indexa la matriu a partir de 0.

[10] Cal restar la unitat a l'índex perquè s'ha obviat la diagonal en la representació interna de la matriu.

$$\mathcal{D} = \begin{pmatrix} (d_{12} & \dots & \dots & d_{1i-1} & \dots & \boxed{d_{1i}} & \dots & d_{1i+1} & \dots & d_{1n}) \\ & (d_{23} & \dots & d_{2i-1} & \dots & \boxed{d_{2i}} & \dots & d_{2i+1} & \dots & d_{2n}) \\ & & & & & \vdots & & & & \\ & & & & & (\boxed{d_{i-1i}} & \dots & d_{i-1i+1} & \dots & d_{i-1n}) \\ & & & & & & \dots & (\boxed{d_{ii+1}} & \dots & \boxed{d_{in}}) \\ & & & & & & & \ddots & & \\ & & & & & & & & & (d_{n-1n}) \end{pmatrix}$$

Pos_matr

Utilitza l'estructura *l_objectes*, que és paral·lela a la matriu de distàncies per a calcular l'índex corresponent a l'objecte en la matriu, i s'evita l'accés directe a la mateixa. També serviria per a calcular la posició d'un objecte en la matriu de dades si es cridés amb **n** valent 0.

Distancies_agregades

Aquesta funció és un artilugi per a utilitzar efectes laterals i fer la modificació simultània de diverses estructures en determinat ordre, del que ja se n'havia parlat amb anterioritat en aquest texte (3.4.5). Algunes d'elles es modifiquen aprofitant el mecanisme de pas de paràmetres, en la crida a la funció, i d'altres en el propi cos de la funció, de forma que al final l'entorn resulta coherent, encara que el resultat que es retorna sigui idènticament un dels paràmetres que es passen: La modificació no s'ha fet en la funció, sinó just en la crida a la mateixa. En l'apèndix A es troba el codi corresponent.

Mínim

Donada la implementació de la matriu de distàncies, de la qual tan sols se'n guarda la part sobrediagonal, només es té explícitament la segona meitat de cada fila d'aquesta matriu. En el càlcul del mínim d'una fila cal, per tant, inferir l'altra meitat de la fila de la qual se'n volen el mínim abans d'iniciar el càlcul.

La solució és més o menys trivial: Quan una matriu és simètrica, la disposició

dels seus elements té el següent aspecte (ja que $d_{ij} = d_{ji}$):

$$\begin{pmatrix} d_{11} & d_{12} & \boxed{d_{13}} & d_{14} & \dots & d_{1n} \\ d_{12} & d_{22} & \boxed{d_{23}} & d_{24} & \dots & d_{2n} \\ \boxed{d_{13}} & \boxed{d_{23}} & \boxed{d_{33}} & d_{34} & \dots & d_{3n} \\ \vdots & & & & \ddots & \\ d_{1n} & d_{2n} & d_{3n} & d_{4n} & \dots & d_{nn} \end{pmatrix}$$

i per tant, la part subdiagonal de la fila i correspon a la columna sobrediagonal del mateix índex.

Si es considera que no s'emmagatzema la diagonal de la matriu, l'element de la columna j de la primera fila ocuparà la posició $j-1$ en la representació interna de la matriu.

Per a les files subsegüents, cada cop la sobrediagonal té un element menys, i la columna j ocupa una posició d'índex decreixent segons el número de fila en el vector que representa internament la sobrediagonal d'aquesta fila.

En definitiva, $m_{ij} = m[i][j-i]$, $\forall i > j$.

Així doncs, els elements de la part subdiagonal de la fila i $m_{i1}, \dots, m_{ii-1}$ s'obtenen com m_{ij-i}, $j = 1, \dots, i+1$.

El procediment de càlcul del mínim d'una fila i es fa en dues passes: En primer lloc es recupera la part subdiagonal de la fila, alhora que se'n calcula el mínim valor, construint el parell $(oinf_j, dinf_{ij})$, essent $dinf_{ij}$ aquest mínim i $oinf_j$ l'objecte que indexa la columna j. El segon pas consisteix a calcular el mínim de la part sobrediagonal $(osup_j, dsup_{ij})$. El mínim de tota la fila vé determinat pel mínim dels mínims locals calculats, essent (o_j, d_{ij}), on $d_{ij} = \min(dinf_{ij}, dsup_{ij})$, i

$$o_j = \begin{cases} oinf_j & , si \quad d_{ij} = dinf_{ij} \\ osup_j & , si \quad d_{ij} = dsup_{ij} \end{cases}$$

En principi, aquests dos subprocessos es poden implementar per separat. Inicialitzant cada càlcul amb un parell del tipus $(-, \infty)$, s'obtenen els mínims locals a cada subvector, i tot seguit s'utilitzen per a calcular el mínim global a la fila.

$$\left.\begin{array}{l} \left.\begin{array}{l} (-,\infty) \\ Subdiagonal(\mathcal{D}_i) \end{array}\right\} \quad (oinf_j, dinf_{ij}) \\ \left.\begin{array}{l} (-,\infty) \\ Sobrediagonal(\mathcal{D}_i) \end{array}\right\} \quad (osup_j, dsup_{ij}) \end{array}\right\} \quad (o_j, d_{ij})$$

Optimitzant una mica, si el resultat d'un d'aquests subprocessos s'utilitza com a valor inicial de l'altre, p. ex.:

$$\left.\begin{matrix}(-.\infty)\\ Subdiagonal(\mathcal{M}_i)\end{matrix}\right\} \quad (oinf_j, dinf_{ij}) \quad \left.\begin{matrix}\\ Sobrediagonal(\mathcal{M}_i)\end{matrix}\right\} \ (o_j, d_{ij})$$

sense alterar cap d'aquests procediments, el resultat del darrer que s'executa ja no és local a la subfila corresponent, sinó que és directament el mínim global de tota la fila.

Una última particularitat que presenta aquesta funció és que obvia el tractament de l'element diagonal que, per ser 0, sempre determinaria el mínim de la fila, quan aquest fet no aporta cap informació útil al classificador[11].

3.6 Construir matriu de distàncies

Aquest mòdul conté el procediment de creació paral·lela de la matriu de distàncies i de la llista de proximitats.

3.6.1 Interfície

La funció que construeix aquestes dues estructures és

(matriu_distancies metrica l_obj α β criteri)

i construeix una matriu triangular de distàncies, en la mètrica i criteri indicats, per als individus l'**l_obj**.

metrica Afecta al càlcul de les distàncies entre individus. Pot prendre els quatre valors especificats en [3.3.1].

l_obj Objectes que configuraran la matriu de distàncies.

α, β Constants de ponderació de subdistàncies. Només tenen sentit si es treballa en la mètrica mixte.

criteri Indica com s'agreguen els individus. Pot prendre dos valors, que es detallen en [3.3.1].

[11] Ja se sap que tot objecte es té com a més proper a si mateix, però això no té cap interés per a determinar parelles de veïns recíprocs.

3.6.2 Precondicions

Per a cridar la funció, la llista *L_objectes* i la matriu de dades han d'existir. Si **metrica**= mixte, calen també les dues matrius de subdistàncies construïdes.

Normalment, la crida es fa amb *L_objectes* i $\alpha = \beta = 1$ si no s'està en mètrica mixte, en el qual cas, es fa $\alpha = alfa$ i $\beta = beta$, constants calculades pel propi sistema en executar *Carregar*, utilitzant el paràmetre opcional **metrica**.

Aquesta matriu es fa servir únicament per a averiguar quin és l'element més proper a l'individu que s'està intentant classificar en cada pas, i que vé determinat pel mínim de la fila corresponent en la matriu complerta de distàncies.

El càlcul del valor mínim de cada fila es pot fer en paral·lel a la construcció d'aquestes matrius, i evitar un recorregut seqüencial d'ella mateixa *a posteriori*. És per això que, abans de cridar aquesta funció, cal haver inicialitzat la llista de més propers també.

Per últim, en mètrica mixte o χ^2 també cal la llista amb els efectius de les variables categòriques, per al càlcul de les distàncies.

3.6.3 Postcondicions

Aquesta funció es crida un sol cop, abans de començar la classificació, amb la finalitat de calcular la matriu que serà la base per a identificar les parelles de veïns recíprocs.

Com a efecte lateral s'actualitza la llista de proximitats en funció dels valors de les distàncies que es calculen. Inicialitzada al començament de l'algorisme amb elements de la forma (i, i, ∞) per a cada objecte del núvol de punts, en calcular la distància entre i i i' s'actualitzen els parells corresponents a aquests dos elements en la llista, guardant sempre el de distància mínima.

3.6.4 Accés a altres mòduls

La funció **matriu_distàncies** accedeix al mòdul *Veïns* per a actualitzar la llista de proximitats, i el *Distàncies* per a calcular les distàncies entre individus.

3.6.5 Comentaris generals

El procés de creació de la matriu de distàncies

Es basa en el càlcul per files de les distàncies d'un element a la resta del núvol (o part d'ell). De fet, per a un element i, es calcularan les distàncies $d(i,j)$ per $j = i+1,\ldots,n$.

Ara bé, en la mètrica mixte, les distàncies es calculen com

$$d^2(i,i') = \alpha d_{C_{ii'}} + \beta d_{Q_{ii'}}$$

En aquest cas, la funció *carregar* ha generat els símbols **alfa**= α i **beta**= β, i per a fer-ho s'han construït les dues matrius de subdistàncies $\mathcal{D}_C$ i $\mathcal{D}_Q$ sobre les variables *Dquant* i *Dquali* respectivament. Com que ja s'havien calculat, es poden aprofitar per a inicialitzar la matriu de distàncies en aquest cas i fer:

$$\mathcal{D} = \alpha\mathcal{D}_C + \beta\mathcal{D}_C$$

Un cop integrades en la matriu de distàncies, es pot assignar **NIL** a les variables *Dquant* i *Dquali*, per a que el recollidor de deixalles alliberi l'espai que ocupen, ja que no fan cap més falta.

A partir d'aquí, el tractament pel cas mixte i els altres convergeixen en un únic algorisme general.

3.7 Veïns

L'algorisme dels veïns recíprocs està íntimament lligat a la identificació de l'element més proper a un donat en el núvol de punts a classificar. Això determinarà quines agregacions fer, i la jerarquia resultant.

En diversos punts d'aquest document s'ha esmentat la conveniència de mantenir una estructura de més o menys fàcil accés amb el veï més proper de cada punt del núvol que s'estudia.

Aquesta estructura és la llista d'associació *l_mes_proper* que es gestiona de forma simultània a la matriu de distàncies entre individus. Es composa d'elements de la forma *(i j d)*, indicant que l'element més proper d'i és j, i que la distància entre ells és d.

3.7.1 Interfície

Les funcions que gestionen la llista de proximitats són:

1. **(ini_l_mes_proper l_obj)** Inicialitza la llista de proximitats.

   l_obj Llista d'objectes que intervenen en la construcció de l'estructura que ens ocupa.

2. **(mes_proper node)** Aquesta és la funció d'accés a la llista de proximitats, i calcula el veí més proper d'un objecte del núvol de punts.

   node És l'identificador de l'objecte del que se'n vol saber el més proper.

3. **(actual_mes_proper element llista distància node1 node2)** Actualitza una llista de proximitats a partir de la distància existent entre dos dels objectes que hi intervenen.

   element Terme de la llista de proximitats corresponent, susceptible de ser canviat.

   llista Llista de proximitats

   distancia Distància existent entre **node1** i **node2**.

   node1, node2 Identificadors dels dos objectes del núvol de punts.

4. **(modif_mes_proper l_obj elem vei td f_actual mat_d obj_orden l_prox)** Modifica la llista de proximitats considerant que **elem** i **vei** acaben de desaparèixer del núvol de punts a classificar (pel fet que s'hagin agregat en una nova classe).

   l_obj Llista d'objectes que queden per tractar.

   elem, vei Objectes que s'han agregat.

   td Mètrica en la que s'opera.

   f_actual Índex de la fila de la matriu de dades que conté la descripció de la nova classe formada per **element** i **vei**.

   mat_d Matriu de distàncies que ja no conté les files (ni columnes) corresponents a **element** i **vei**.

   obj_orden Índex simbòlic de la matriu de distàncies.

   l_prox Llista de veïns més propers.

3.7.2 Precondicions

Ini_l_mes_proper

La crida a aquesta funció es fa amb la llista d'objectes a classificar **l_objectes**, que la funció *carregar* inicialitza degudament.

Mes_proper

No cal cap condició especial. Si es crida amb un objecte que no és a la llista *l_mes_proper* (la qual cosa mai passa segons el flux del programa) retorna NIL.

Modif_mes_proper

Per a que la crida a **modif_mes_proper** retorni el resultat que s'espera, cal que **l_obj** i **mat_d** ja no continguin cap referència a **element** i **vei**. Els elements de **obj_orden** han d'aparèixer en l'ordre en que estan les files de la matriu de distàncies. D'altra banda cal que el centre de gravetat de la nova classe c ja hagi estat calculat i insertat a la matriu de dades, i també que la llista de proximitats contingui el terme corresponent a aquest nou objecte[12]. La crida s'efectua amb una llista de proximitats que ja no conté els termes corresponents a elem i vei, i amb **f_actual** a 1, ja que insertem els nous elements pel cap de la matriu de dades.

3.7.3 Postcondicions

La funció ini_l_mes_proper

Es crida un sol cop a l'inici del procés de classificació, en l'etapa d'inicialitzacions. El resultat és la llista

$$l = ((i\ i\ \infty), \forall i \in \text{l\_obj})$$

[12] Abans de la crida, s'inserta a *l_mes_propers* el terme inicialitzat com (c c ∞). Aquesta és una modificació de la llista de proximitats que es fa directament accedint a la implementació, perquè no comporta cap tipus de complicació. Quan es completi la matriu de distàncies amb la nova fila, quedarà actualitzat correctament, seguint el procediment ordinari.

Mes_proper

Si la llista conté un element de la forma (*node vei dist*), la crida (**mes_proper node**) retorna el símbol **vei**.

Aquesta funció servirà per a construir la cadena de més propers de l'algorisme i identificar les parelles de veïns recíprocs.

En essència, el que fa el cos principal del procediment és cridar encadenadament aquesta funció des del primer objecte de la llista *o1*:

(mes_proper o1) = o2

(mes_proper o2) = o3

⋮

fins identificar dos veïns recíprocs, moment en el qual s'atura la cadena, i es fa l'agregació corresponent, per a prosseguir amb la seva construcció a partir del darrer element encadenat.

Actual_mes_proper

El resultat és la llista actualitzada en el terme **element** segons la nova distància calculada. Ja s'ha comentat com l'actualització d'aquesta llista és un nou procés que es fa en paral·lel amb la construcció de la matriu de distàncies entre individus, per una banda, i la seva modificació per l'altra. En construir la fila j de la matriu $\mathcal{D}$ es calculen les distàncies entre els individus a classificar. Cada distància que es calcula és nova informació que pot modificar la llista de proximitats. Com que la distància $d_{jj'}$ fa referència als dos objectes j i j', són els termes corresponents a ells dos els susceptibles de ser modificats. És, per tant, dues vegades, amb elements (j p_j d_{jp_j}) i (j' $p_{j'}$ $d_{j'p_{j'}}$) respectivament, que es crida aquesta funció cada cop que es calcula una distància.

Modif_mes_proper

El resultat que genera aquesta funció és la llista de proximitats corresponent al núvol que no conté els dos elements fusionats. Els elements que quedaran afectats són els de la forma *(i,i',d)* on *i'* és algun dels objectes que acaben de desaparèixer. És en aquest, i només en aquest cas, que cal recalcular directament sobre la matriu reduïda l'element més proper als individus que es vegin afectats per aquesta situació, i això és el que es fa accedint a la funció *Minims* del mòdul *Matriu de distàncies.*

3.7.4 Accés a altres mòduls

Aquest és un mòdul que accedeix al de *Formules*, al *General* i a *Matriu de distàncies.*

3.7.5 Comentaris generals

La inicialització

Després d'inicialitzar la llista de proximitats, i abans de començar la classificació, la llista estarà modificada i, per a cada element, contindrà el seu veí més proper, segons la matriu de dades observades. Com que es busquen els elements més propers entre si, és a dir, un mínim, cal inicialitzar cada parell amb un valor positiu molt gran, per tal que amb la primera comparació hi hagi actualització i es mantingui la invariant del cos iteratiu[13].

Actual_mes_proper

Amb la crida a aquesta funció canviarà el terme element si $\mathbf{node_2}$ és més proper a $\mathbf{node_1}$ que el veí que tenim a la llista i_p. Així, el càlcul de la distància $d_{jj'}$ transforma *l_nro_nodes*, de termes tipus $(i_l\, i_{p_l}\, d_{lp_l})$, en:

$$l\_nro\_nodes = ((i_1\, i_{p_1}\, d_{1p_1}) \ldots (i_j \left\{ \begin{array}{ll} i_{p_j} & d_{jp_j} \quad \text{, si } d_{jp_j} \leq d_{jj'} \\ i_{j'} & d_{jj'} \quad \text{, sino} \end{array} \right) \ldots\ldots (i_n\, i_{p_n}\, d_{np_n}))$$

[13] A la passa k tenim com a l_mes_propers els parells de veïns més propers del núvol format pels k primers elements de la llista.

Modif_mes_proper

S'efectua un recorregut seqüencial dels objectes per als quals, si a *l_mes_proper* hi tenim $(i\ f\ d)$, amb $f \in \{element, vei\}$, es parteix de 0 i es calcula sobre la corresponent fila de la matriu de distàncies l'element més proper a i d'entre els que conformen el nou núvol.

3.8 Distàncies

En aquest mòdul es gestiona tot el càlcul de distàncies entre individus.

La semàntica d'alguns paràmetres es mantindrà constant al llarg de totes les funcions definides dins d'aquest mòdul:

1. **metrica** Indica la mètrica de treball. Pot tenir els valors habituals especificats amb anterioritat.
2. α i β Són les constants de ponderació de les parts contínua i discreta de la distància quan es treballa en mètrica mixte.
3. **criteri** Especifica el criteri d'agregació que s'utilitza en la classificació.

3.8.1 Interfície

De les funcions implementades en aquest mòdul, només la primera de les que es detallen a continuació és accessible des de fora:

1. **(noves_distancies metrica pseudonode arbre dev α β criteri prearbre pos2)** Calcula les distàncies entre el node resultant d'una agregació i la resta de nodes del núvol de punts. El càlcul es realitza en funció de la metrica de treball segons s'explica en l'apartat [2.3], i depenent del criteri d'agregació que s'estigui fent servir: Si el criteri és **Ward**, es substitueixen les distàncies entre punts per l'augment d'inèrcia que suposaria agregar **Pseudonode** amb cada un dels punts restants. Quant als paràmetres:

   pseudonode Node resultant de l'agregació més recent (entre element i vei).

   arbre Núvol de punts que encara queda per classificar.

dev Distància existent entre **element** i **vei**.

prearbre Llista de nodes després d'haver eliminat **element**.

pos2 Índex numèric de les files de la matriu de distàncies corresponents als elements agregats. Aquest paràmetre s'utilitza sota el criteri de **Ward**.

2. **(distancies metrica node llista_nodes α β criteri)** Funció que calcula les distàncies de **node** a cada element de **llista_nodes** segons mètrica i criteri especificats en la crida mitjançant els paràmetres corresponents.

   En aquest treball s'han implementat dos criteris, un dels quals no actualitza les distàncies per aquesta via, i per tant, a excepció feta de l'etapa d'inicialitzacions, el criteri que figuri en la crida d'aquesta funció tindrà un únic valor (**centroide**). El significat dels paràmetres és:

   node Identifica la nova classe que s'ha creat.

   llista_nodes Nodes que constitueixen el núvol de punts a classificar arribat aquest punt de l'algorisme.

3. **(pseudodistancies pseudonode nounode llista_nodes m_1 m_2 d_1 d_2 d_{12})** Calcula les pseudodistàncies entre **pseudonode** i cada element de **llista_nodes**. Quant als paràmetres:

   pseudonode Símbol de la classe procedent d'agregar **element** i **vei**.

   nounode Llista dels dos nodes agregats.

   m_1 Nombre d'individus que integren la primera del les subclasses agregades, és a dir, la identificada per **element**.

   m_2 *Id.* però per a la segona de les classes que s'agreguen, **vei**.

   d_1 Vector de distàncies entre **element** i la resta de nodes pendents de classificar.

   d_2 *Id.* però per a l'element **vei**.

   d_{12} Distància entre **element** i **vei**.

3.8.2 Precondicions

Noves_distancies

Es pot fer la crida després d'agregar **element** i **vei** en **Pseudonode** i haver actualitzat la matriu de dades insertant pel cap la fila corresponent al centre de gravetat de la classe que s'acaba de crear.

3.8.3 Postcondicions

Noves_distancies

El resultat és una nova fila de la matriu de distàncies corresponent a l'objecte **Pseudonode** i aquesta funció forma part del procés de modificació d'estructures que té lloc en agregar dos nodes en un de sol.

La funció distancies

Retorna el vector $(d_{n1} \dots d_{nm})$ on d_{ni} és la distància existent, en la mètrica indicada, entre **node** i l'$i-sim$ element de **llista_nodes**.

Pseudodistancies

D'aquesta funció s'obté una fila preparada per a ser insertada com a primer element de la matriu de distàncies entre individus on per a cada element del núvol de punts hi ha la pseudodistància a la nova classe [2.4.2]. S'utilitza aquesta funció si el criteri d'agregació és **Ward**.

3.8.4 Accés a altres mòduls

Des d'aquí únicament hi ha accés a les funcions del mòdul de fòrmules matemàtiques, al de funcions de propòsit general, i a *Matriu* per a accedir a les files de la matriu que figuren com a operants dels càlculs de distàncies.

3.8.5 Comentaris generals

Referent a α i β

Al llarg de tot aquest mòdul els paràmetres α i β només tenen sentit si es treballa en mètrica **mixte**. I és només en aquesta ocasió que el sistema els calcula o els demana a l'usuari. En cas contrari, s'assigna un valor per defecte a aquests paràmetres que és la unitat, i que correspon al cas en que no s'efectua ponderació alguna.

Noves_distàncies

En aquesta funció únicament es determina si cal utilitzar distàncies o pseudodistàncies en funció del criteri d'agregació que s'utilitzi. Si el criteri és el del **centroide**, les distàncies del nou node a la resta es calculen en funció de les coordenades que aquest nou node — i els restants — tenen en l'espai de les variables a estudiar. Si el criteri és **Ward**, però, el que s'aplica és una funció sobre les files de la matriu de distàncies corresponents als objectes que s'acaben d'agregar. Aquestes files s'obtenen en reduir la matriu de distàncies com a resultat de la funció **eliminar_f_c**. La seqüència d'aplicacions que es fa d'aquesta matriu és la següent:

```
(eliminar_f_c
            elem pos2 ...
            (eliminar_f_c vei ...pos1 matriu  'fila_vei)
            'fila_elem
)
```

la qual cosa fa que **fila_vei** tingui un element més que **fila_elem**. El paràmetre **pos2** és el que indica quin és l'element de **fila_vei** que cal eliminar per a tenir dues files amb les components apariades. El càlcul de les pseudodistàncies s'efectuarà després d'haver eliminat el **pos2** – *sim* element de **fila_vei**.

Les funcions distàncies i pseudodistàncies

Estan implementades com un procés recursiu per la cua, on en cada crida es calcula la distància entre **node** i el primer element de **llista_nodes**, fins que

aquesta llista queda buida.

Així com les distàncies es calculen directament a partir de la matriu de dades, es disposa d'una forma recorrent de calcular les pseudodistàncies a partir de la pròpia matriu de pseudodistàncies, i és per això que en aquest darrer cas calen com a paràmetres les files d'aquesta matriu corresponents als dos elements que es fusionen. El resultat ja no contindrà informació referent a aquests dos elements.

KLASS té dues funcions internes que calculen la distància o pseudodistància entre dos elements donats, i que són utilitzades per les anteriors per a desfer la recursió en cada pas. Ambdues actualitzen la llista de veïns més propers en paral·lel a la construcció d'aquests vectors.

Les pseudodistàncies es calculen de forma totalment independent a la mètrica de treball iterant sobre una matriu inicial. És en el càlcul d'aquesta primera matriu on intervé la mètrica que emmarca la classificació. La matriu s'inicialitza amb la meitat de les distàncies entre els punts a classificar, utilitzant les funcions de distàncies que es detallen més avall.

En el càlcul de les distàncies, el paràmetre **metrica** discrimina la funció que cal utilitzar per a fer-ho. N'hi ha una per a cada mètrica implementada:

(d_euclidia o1 o2 norm criteri) Calcula la distància euclídia entre dos objectes normalitzant-la o no segons el valor del paràmetre **norm** que pot ser *si* o *no*. La normalització de la distància s'implementa com una postoperació que es composa amb el càlcul de la distància euclídia ordinària quan el paràmetre de normalització està actiu. Per a treballar amb les distàncies normalitzades, no cal res més que calcular amb un preprocés la mitjana i variància de cada columna de la matriu de dades, ja que, si bé les agregacions canvien el núvol de punts, com que sempre subtitueixen individus pel seu centre de gravetat, els moments de primer i segon ordre es mantenen invariants.

(d_chi_2 node1 node2 criteri) Calcula la distància de $\mathbf{node}_1$ a $\mathbf{node}_2$ en la mètrica de χ^2 segons el criteri d'agregació indicat.

(d_mix node1 node2 α β criteri) Calcula la distància mixta entre els nodes indicats pels dos primers paràmetres, utilitzant les constants de ponderació α i β.

3.9 Centre de gravetat.

En aquest mòdul hi ha les funcions necessàries per a calcular el centre de gravetat de la classe que apareix com a conseqüència d'una agregació.

3.9.1 Interfície

Una única funció constitueix la interfície d'aquest mòdul:

(centredg node1 node2 metrica)

i calcula el centre de gravetat de la classe resultant de l'agregació de **node1** i **node2**. Els paràmetres:

node1, node2 són els símbols de les subclasses que s'agreguen.

metrica indica la mètrica en que es treballa. Segons sigui, variarà el càlcul del centre de gravetat.

3.9.2 Precondicions

El paràmetre **metrica** ha de tenir un dels següents valors:

- euclidia
- eucl_norm
- chi_2
- mixte

En un altre cas, la funció no retonarà més que el valor **NIL**.

3.9.3 Postcondicions

Retorna el centre de gravetat dels individus que formen les classes $\mathbf{node}_1$ i $\mathbf{node}_2$ segons la mètrica indicada per **metrica**, i a partir del *c.d.g.* de cada subnúvol.

Aquesta funció s'utilitza quan es produeix una agregació i crea la nova fila de la matriu de dades, que després s'hi incorporarà com a primer element.

3.9.4 Accés a altres mòduls

Cal accedir a les files de la matriu de dades corresponents a **node1** i **node2**, i per tant al mòdul *Matriu*, així com al mòdul *Fòrmules* per a efectuar els càlculs matemàtics.

3.9.5 Comentaris generals

Encara que **metrica** pot prendre quatre valors diferents, només es distingeixen tres casos a l'hora de calcular el centre de gravetat dels dos núvols **node1** i **node2**, donat que la forma de fer-ho en mètrica euclidia normalitzada s'identifica amb la que es descriu per a la mètrica euclídia.

El càlcul que s'aplica per a cada cas és el resultat del raonament teòric que s'efectua en l'apartat [2.3]. Suposant que tenim n variables:

Cas euclidi: $oe_i = \frac{c_1 e_1 + c_2 e_2}{c_1 + c_2}$, $\forall i = 1, \ldots, n$

Cas chi_2: $oc_i = \begin{cases} \text{agregar}(c_{1i}, c_{2i}) & c_{1i}, c_{2i} \text{ simbòlics} \\ \text{combinar}(c_{1i}, c_{2i}) & c_{1i} \text{ simbòlic, i } c_{2i} \text{ forma extesa} \\ \text{combinar}(c_{2i}, c_{1i}) & c_{2i} \text{ simbòlic, i } c_{1i} \text{ forma extesa} \\ \text{subsum}(c_{1i}, c_{2i}) & c_{1i}, c_{2i} \text{ forma extesa} \end{cases}$

on agregar, combinar i subsum són les funcions que figuren al mòdul de fòrmules matemàtiques.

Cas mixte: $o_i = \begin{cases} oe_i & \text{si la variable } i \text{ és contínua} \\ oc_i & \text{si la variable } i \text{ és categòrica} \end{cases}$

3.10 La llista d'efectius

L'estructura de dades que gestiona aquest mòdul té, per a cada variable qualitativa k de modalitats $m_1, \ldots, m_c$ un element de la forma:

$$(k\ (m_1\ e_1) \ldots (m_c\ e_c))$$

on e_i és el contingent de la mostra que pertany a la modalitat m_i de la variable. Per a implementar-la, s'utilitza una llista d'associació, que permet obtenir de forma fàcil la informació a partir del nom de la variable.

3.10.1 Interfície

El tipus abstracte de dades *llista_efectiu* precissa d'operacions constructores i consultores per a ser manipulat. No obstant, en la interfície d'aquest mòdul només hi ha operacions pertanyents al primer d'aquests grups. D'una banda, la implementació del tipus descansa sobre una estructura no elemental de LISP, per a la qual el propi llenguatge proporciona funcions de tractament. De l'altra, l'accés a la informació és (via aquestes funcions), a més de freqüent, trivial. És per això que no s'han emmascarat aquestes crides sota la definició d'operacions consultores del tipus, que manquen en aquesta interfície.

Les dues operacions implementades permeten inicialitzar i modificar la llista:

1. **(crear_comptatge l_props n)** Crea una llista d'efectius comptant quants elements de la matriu de dades inicial pertanyen a cada una de les categories de cada variable qualitativa. Pel que fa als paràmetres:

   l_props Llista de totes les propietats a tractar.

   n Paràmetre opcional que representa l'índex, en la llista anterior, de la propietat que es tracta en cada crida[14]. El seu valor per defecte és 0.

2. **(modif_l_efec obj1 obj2 cdg l_props llista_efe)** Actualitza l'estructura després d'una agregació. Detallant el significat dels paràmetres:

   obj1 i obj2 Són els noms dels dos objectes que s'han fusionat.

   cdg És el símbol que identifica la nova classe que els comprèn.

   l_props És la llista de propietats que queden per examinar.

   llista_efe És la part de llista d'efectius corresponent als elements qualitatius d'*l_props*.

[14] El motiu d'aquest paràmetre és que la funció és recursiva, i a més no genera tants elements com té la llista de propietats, que també pot contenir variables contínues. S'incrementa a cada iteració.

3.10.2 Precondicions

Crear_comptatge

Aquesta funció pressuposa l'existència de :

1. La matriu de dades
2. La llista de propietats

objecte d'estudi, que es generen en les primeres passes de **carregar**. La crida es fa com a **(crear_comptatge l_objectes 0)**.

Modif_l_efec

S'executa just en el procés d'agregació de dos nodes; cal haver definit ja el símbol identificador i haver calculat també les coordenades del centre de gravetat de la nova classe.

3.10.3 Postcondicions

Crear_comptatge

Genera la llista d'efectius que conté la informació corresponent al núvol de punts a classificar. Es crida un únic cop en tot el procés, en la fase d'inicialitzacions.

Modif_l_efec

Aquesta funció, a diferència de l'anterior, és cridada cada cop que s'efectua una agregació i es crea una nova classe. El resultat és la llista d'efectius totals modificada (veure l'apartat de comentaris generals).

3.10.4 Accés a altres mòduls

Únicament es fa accés al fitxer d'interfície *Interfase* per a calcular els diferents efectius.

3.10.5 Comentaris generals. El procés de modificació de la llista

En agregar dos objectes en un de sol, el núvol de punts es redueix en una unitat, i per tant, els efectius de cada variable es modificaran: El que cal fer és eliminar els objectes fusionats i afegir el seu centre de gravetat en la forma convenient. Així doncs, pel cas d'una variable, eliminar o afegir un objecte en forma compacta [2.3] és trivial, ja que es redueix a sumar o restar 1 a l'efectiu de la modalitat corresponent, segons estem afegint o eliminant respectivament.

Ara bé, quan l'objecte està en forma extesa, aquesta unitat no recau sobre una única categoria de la variable, i cal modificar els efectius de les diverses modalitats segons la distribució de l'objecte que s'extreu o incorpora. El procés és independent per cada modalitat de la variable, i el que es fa és un recorregut de les modalitats modificant els seus efectius segons s'indica a continuació:
Per a cada modalitat m_i

1. Es resta la contribució a aquesta modalitat dels dos objectes que es fusionen.
2. Se suma la contribució de l'objecte fusió.

La contribució d'un objecte en forma extesa $o = ((k_1\ e_1)\dots(k_c\ e_c))$ a una certa categoria k_j és e_j directament[15], mentre que si l'objecte està en forma compacta, $o = k_i$ és

$$e_j = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}$$

En definitiva, per a modalitat m_j de cada variable categòrica k_i, s'actualitza l'efectiu e_j com

$$e_j = e_j - c_1 - c_2 + c_{12}$$

si c_1 i c_2 són les contribucions dels elements que es fusionen, i c_{12} la del que els engloba com a nova classe.

[15]Per això ha interessat representar via la distribució marginal el centre de gravetat d'una classe. En un altre cas, la modificació de la llista d'efectius requeriria més càlculs.

3.11 Matriu

En tot estudi estadístic les dades es disposen en una matriu cada element de la qual representa la mesura d'un individu de la mostra en una certa variable. LISP proporciona l'estructura *n-dimensional array*, que es presta molt bé a la implementació aquesta matriu de dades.

El mòdul *Matriu* ofereix les operacions que serviran per a gestionar aquesta estructura, des del punt de vista de les necessitats del programa.

3.11.1 Interfície

Aquestes operacions es redueixen a dues, a estudiar per separat:

1. **(fila_matriu objecte)** Per accedir a una fila de la matriu. En una fila de la matriu hi ha la informació referent a un cert individu respecte de totes les variables que intervenen en l'anàlisi. Respecte de l'únic paràmetre que té:

   objecte és el símbol de l'objecte del qual es vol la descripció en termes de les variables a estudiar.

2. **(inserta_fila_cap fila matriu)** Per afegir-li a la matriu la primera fila.

   fila És la fila a insertar en la matriu.

   matriu És la matriu a la qual afegirem la fila.

3.11.2 Precondicions

Per a fer una crida a la primera de les funcions abans esmentades, cal tenir comptades les variables qualitatives que s'estan tractant en la variable *ntot*.

3.11.3 Postcondicions

En aquest mòdul no hi ha cap efecte lateral, i per tant, com a postcondició es compta únicament amb el valor que cada funció retorna.

Fila_matriu

Funció que retorna la fila de la matriu referent a l'individu que simbolitza objecte. Aquesta és una funció àmpliament cridada en el càlcul de distàncies entre individus.

Inserta_fila_cap

Funció que inserta el primer paràmetre com a primera fila de la matriu que figura com a segon paràmetre de la crida i troba la seva utilització en les actualitzacions que es fan de les estructures de dades després de crear una nova classe.

3.11.4 Accés a altres mòduls

Matriu és un mòdul de baix nivell, que no accedeix a cap altre mòdul.

3.11.5 Comentaris generals

La funció fila_matriu

Forma part de la interfície que comunica **LINNEO** amb **KLASS** i no apareix físicament en el aquest mòdul sinó en el fitxer que agrupa totes les funcions d'interfície.

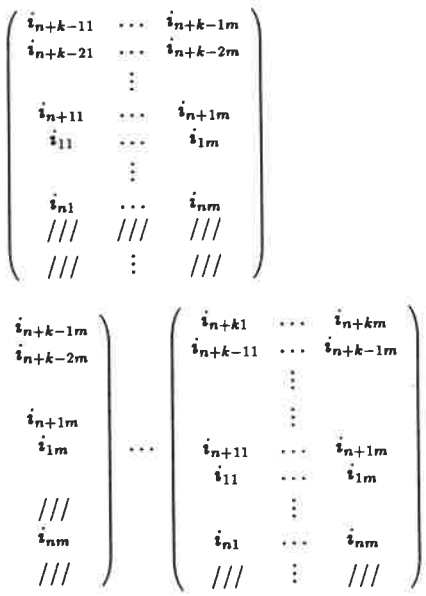
Donat el símbol de l'objecte del qual se'n vol la descripció, el primer que cal fer és calcular quin índex numèric li correspon com a fila de la matriu de dades, ja que a les matrius de **LISP** s'hi accedeix per posició, i no de forma simbòlica. Per a això, cal recórrer seqüencialment la llista d'objectes i efectuar un comptatge dels elements que es passen abans de trobar el que es demana.

Sembla, doncs, eficient d'associar a cada objecte l'índex numèric que li correspon com a fila de la matriu. El que passa és que en crear noves files de la matriu i col·locar-les les primeres, els índexos de tots els objectes restants canvien, i caldria anar-los actualitzant. Per això s'ha implementat la primera solució en que es precalcula l'índex numèric de l'objecte abans de cada accés. Considerant que s'ha minimitzat en tot el possible l'accés a aquesta matriu, aquesta operació no serà gaire freqüent.

La funció inserta_fila_cap

Treballa sobre una matriu que des del començament del programa té reservat l'espai per a totes les files que es crearan durant l'execució. La inserció d'un nou vector a la primera fila de la matriu suposa el desplaçament de les files restants a la següent. El que cal fer és començar sempre per la darrera fila útil i copiar-la a la primera fila no usada, i repetir l'operació per les files anteriors, fins arribar a la primera fila que es copiaria sobre la segona, i deixaria l'espai a la nova fila que cal insertar.

Suposi's la iteració k d'una classificació d'n individus descrits per m variables. Gràficament, el procés d'inserció del $k - sim$ element a la matriu de dades seria:



3.12 Fòrmules matemàtiques

Aquest mòdul s'estructura com una mena de biblioteca de funcions matemàtiques específiques d'aquest programa, agrupant totes aquelles funcions que implementen càlculs aritmètics a nivell més baix.

Les funcions que conformen aquest mòdul són:

(d_euclidia2 o1 o2 norm) Calcula la distància euclídia al quadrat entre els dos vectors **o1** i **o2**, d'igual dimensió. Opcionalment es pot normalitzar aquesta distància a través del paràmetre **norm**.

$$d\_euclidia2(o_1,\ o_2,\ norm) = \sum_{j=1}^{J} \begin{cases} (o_{1j} - o_{2j})^2, & \text{si } norm = no \\ \frac{(o_{1j}-o_{2j})^2}{s_j^2}, & \text{si } norm = si \end{cases}$$

(d_chi_2_2 llista_efec v1 v2) Calcula la distància de χ^2 entre els dos vectors de dimensió n, v_1 i v_2.

$$d\_chi\_2\_2(llista\_efec,\ v_1,\ v_2) = \sum_{j=1}^{J} \begin{cases} 0, \text{si } v_{1j} = v_{2j} \\ contrib\_dist(v_{1j}, v_{2j}, llista\_efec_j) \end{cases}$$

amb la funció interna al mòdul *contrib_dist*, que calcula el terme amb que una certa variable categòrica intervé en aquest sumatori, i que es defineix de la següent forma:

$$\left.\begin{array}{l} contrib\_dist(c_1, c_2, \\ ((m_1\ ef_1), \ldots, (m_k\ ef_k))) \end{array}\right\} = \begin{cases} \frac{1}{ef_{i_1}} + \frac{1}{ef_{i_2}}, & \text{si } c_1 = m_{i_1} \\ & \text{i } c_2 = m_{i_2} \\ \frac{(f_{1i}-1)^2}{ef_{i_1}} + \sum_{j\neq i}^{K} \frac{f_{2i}^2}{ef_j}, & \text{si } c_1 = m_i, \text{ i} \\ & c_2 = ((m_j\ f_j)_{j=1}^{k}) \\ \sum_{j=1}^{K} \frac{(f_{2i}-f_{1i})^2}{ef_j}, & \text{altrament} \end{cases} \tag{3.1}$$

(d_mix_2 l_efec l_pro o1 o2 nvcat α β) Calcula la distància en mètrica mixte entre els objectes **o1** i **o2**:

$$\left.\begin{array}{l} d\_mix\_2(l\_efec,\ l\_pro, \\ o1,\ o2,\ nvcat,\ \alpha,\ \beta) \end{array}\right\} = \sum_{\forall i} \begin{cases} 0, & \text{si } o_{1i} = o_{2i} \\ \frac{\beta}{nvcat} contrib\_dist(o1, o2), & \text{si } v_i \text{ és} \\ & \text{categòrica} \\ \alpha \frac{(o_{1i}-o_{2i})^2}{s_i^2}, & \text{altrament} \end{cases}$$

(agrega_categ llista_mod c1 c2 f1 f2) Donada una variable categòrica de modalitats $(m_1 \ldots m_k)$ i dos objectes en les categories c_1, c_2, i amb pesos

f_1 i f_2 respectivament, questa funció calcula la coordenada (simbòlica o composta) corresponent al *c.d.g.* entre c_1, c_2.

$$agrega\_categ((m_1 \ldots m_k),\ c_1,\ c_2,\ f_1,\ f_2) = \begin{cases} c_1, & \text{si } c_1 = c_2 \\ (p_1 \ldots p_k), & \text{altrament} \end{cases}$$

$$\text{on } p_i = \begin{cases} (c_1 \frac{f_1}{f_1+f_2}), & \text{si } c_1 = m_i \\ (c_2 \frac{f_2}{f_1+f_2}), & \text{si } c_2 = m_i \\ (m_i\ 0), & \text{altrament} \end{cases}$$

(subsum_categ c ((c1 e1) ... (ck ek)) f1 f2) Aquesta funció té ús quan s'està construint el centre de gravetat de dos nodes que s'han agregat per a formar una nova classe un dels quals té representació compacta, mentre l'altre està en representació extesa. Calcula la component del *c.d.g.* corresponent a la variable categòrica en curs.

$$o_i = \begin{cases} (c\ \frac{f_1 e_{1i} + f_2}{f_1+f_2}), & \text{si } c_i = c \\ (c_i\ \frac{f_1 e_i}{f_1+f_2}), & \text{altrament} \end{cases}$$

(combina_categ ((c1 e11)...(ck e1j)) ((c1 e21)...(ck e2j)) f1 f2) Contempla el cas en que tots dos operands estan en representació extesa, i retorna el valor segons la fòrmula general:

$$o_i = (c_i\ \frac{f_1 e_{1i} + f_2 e_{2i}}{f_1 + f_2})$$

(contrib_euc c1 c2 f1 f2) Essent c_1 i c_2 components en una certa variable contínua, calcula el terme de distància corresponent a aquesta variable.

$$contrib\_euc(c_1,\ c_2,\ f_1,\ f_2) = \frac{c_1 f_1 + c_2 f_2}{f_1 + f_2}$$

(augment_inercia m1 m2 mk d1k d2k d12) Augment d'inèrcia que es produïria en agregar la classe formada pels punts 1 i 2 amb la classe k. Els paràmetres són :

mi: massa de la classe i. És a dir, nombre d'individus que la formen.

dij: distància existent entre les classes i i j.

La fòrmula matemàtica que implementa aquesta funció és:

$$\frac{(m_1 m_k)d_k + (m_2 + m_k)d_{2k} - m_k d_{12}}{m_1 + m_2 + m_k}$$

(d_quali_2 lefec lpro o1 o2 nvcat 2 dquali dquant) Calcula les subdistàncies contínua i categòrica d'un parell de punts o_1 i o_2 en paral·lel, retornant, com a valor de la funció, la subdistància qualitativa, i en el símbol global **dquant** la quantitativa:

$$\sum_{\forall i \in C}(o_{1i} - o_{2i})^2 \; ; \; \sum_{\forall i \in Q} contrib\_dist^{16}(o_{1i}, o_{2i}, lefec)$$

3.13 Funcions de propòsit general

En aquest mòdul hi ha funcions de propòsit general que s'utilitzen amb freqüència al llarg de tot el programa. Les funcions són les següents:

1. Les funcions **retorna_dist** i **retorna_sim** retornen el primer paràmetre tal com els hi arriba, i serveixen per a implementar alguns efectes laterals sobre els paràmetres restants. Aprofitant el pas de paràmetres, es fan les assignacions pertinents.

2. La funció **(treure element llista)** retorna una llista l' que es comporta de la següent forma:

$$\left.\begin{array}{ll} e_i \in llista \wedge \; e_i \neq element & \Longrightarrow e_i \in l' \\ e_i = element & \Longrightarrow e_i \notin l' \end{array}\right\} \quad \forall i = 1, \ldots, n$$

   on n és el nombre d'elements de **llista**.

3. El comportament de **(treure_p pos llista)** = l' es defineix com:

$$l'_i = \begin{cases} llista_i, & \text{si } i < pos \\ llista_{i+1}, & \text{si } i \geq pos \end{cases}$$

   i correspon a eliminar de **llista** l'element que ocupa la posició **pos**, considerant que indexem tota llista des de 0.

[16] Veure expressió 3.1.

4. La funció **(n_primers pos llista)** retorna la subllista composada pels primers $n + 1$ elements de **llista**, ja que indexem des de les posicions 0 fins n.

5. **(inserta_ord element llista)**: Donada una **llista** d'elements ordenats de menor a major, el resultat d'aquesta funció és una altra llista d'elements ordenats de menor a major, que conté tots els elements de **llista**, i **element**.

6. Per últim, **(ordena element llista)** es fonamenta sobre l'anterior, i a més d'insertar **element** ordenat en **llista**, manté una longitud màxima de **llista** constant **(n_5_%)**, de forma que, si **llista** ja tenia aquesta longitud en la crida, el resultat serà ella mateixa si **element** es menor que tots els elements de **llista**, o la que conté **element** enlloc del mínim de la llista original, en la posició que li correspon per ordenació.

Capítol 4

Resultats

"Perquè la diferència és principi general,
fa del gènere moltes espècies".
R.LLULL

L'execució del programa s'ha dut a terme amb uns quants bancs de dades, intentant cobrir totes les possibilitats.

En primer lloc s'estudia un cas de variables definides sobre un espai enterament continu, amb poques dades a fi d'efectuar una comparació extensa amb el comportament de **LINNEO**, (secció 4.1).

El mateix s'ha fet per un cas de mètrica mixte, tractant també una matriu petita (secció 4.2).

En la darrera secció d'aquest capítol es presenta una prova més versemblant que les dues primeres, pertanyent també al cas mixte, però amb una matriu de dades de dimensions considerables (veure 4.3), i la particularitat que hi apareixen valors mancants.

4.1 Compositors

Es tracta de classificar vuit directors d'orquestra a partir dels minuts que tarden en dirigir cada un dels quatre moviments de la Novena Simfonia de Beethoven, representats, cada un d'ells en les variables ADAGIO, ALLEGRO, FINALE i SCHERZO. Els fitxers de dades, objectes i propietats són a l'apèndix B, i s'han tret de la tesi doctoral de Núria Piera i Carreté [PIER87].

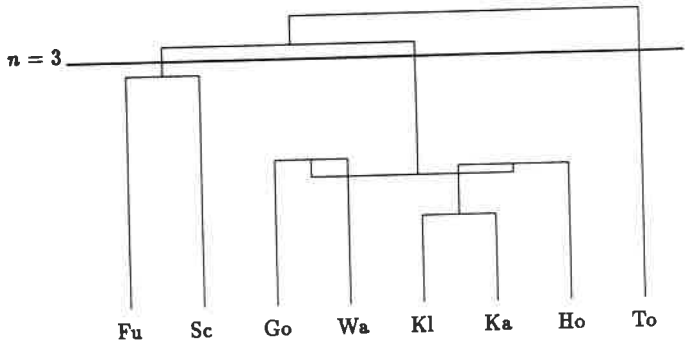


Figura 4.1: Compositors. Criteri del cetroide. Dades normalitzades. Mètrica Euclídia ordinària.

Per estar en un espai de variables contínues, correspon treballar en una mètrica euclídia, bé sia la normalitzada o no.

A la pràctica, les possibilitats que ofereix **KLASS** dins aquest marc són tres:

- Es pot treballar directament sobre les dades, utilitzant la mètrica euclídia ordinària. Això seria pel cas (**carregar compositors no**) que no normalitza les dades, i (**veins_reciprocs l_objectes euclidia**).
- Es pot usar un espai normalitzat, definint sobre els individus, com a mètrica de normalització, la inversa dels rangs. Correspon a les crides (**carregar compositors continuo**), que normalitza amb la inversa dels rangs, i (**veins_reciprocs l_objectes euclidia**).
- Es pot definir una segona mètrica de normalització que és la inversa de la desviació típica. En aquest cas, les crides serien (**carregar compositors no**) i (**veins_reciprocs l_objectes eucl_norm**).

En aquest exemple s'utilitzarà únicament el criteri del **centroide**, perquè el de **Ward** s'adequa a dades en que tingui sentit ponderar les classes de forma proporcional al nombre d'elements que cada una conté, i aquest no és el cas.

Executat el programa en el segon dels casos i sota el criteri del **centroide**, l'arbre jeràrquic que s'obté és el de la figura 4.1.

classe-1	classe-2	classe-3
Furtwangler	Kleiber	Toscanini
Scherchen	Karajan	
	Horenstein	
	Goehr	
	Wand	

Tallant horitzontalment l'arbre, s'obtenen les diferents particions. Les fulles dels subarbres que intersecten l'α-tall són a la mateixa classe. Cada un d'aquests subarbres constitueix una classe. Així, per n=3 classes es té la partició de la taula que apareix a cap de pàgina.

A la vista de l'arbre, Kleiber i Karajan són els compositors que més s'assemblen, ja que són els primers en agrupar-se, mentre que Toscanini és el més diferent respecte del conjunt, i es manté aïllat en una classe en totes les particions excepte en la darrera, en que hi ha una única classe que integra tots els elements.

En aquesta prova, apareix un cas d'inversió, on s'agreguen primer Wand i Goehr, per una banda, i Horenstein i el grup Kleiber-Karajan, per l'altra, quan els seus índexos de nivell són superiors als de la classe conjunta. El criteri del **centroide** és l'únic on es poden donar aquests fenòmens.

Si s'efectua amb LINNEO una classificació normalitzant les dades sota la mateixa mètrica (la inversa dels rangs), en fer variar el radi s'obtenen diverses particions de la mostra.

En concret, per $R = 0$, la partició és la pròpia mostra, i és la més fina possible. Augmentant el radi, les particions esdevenen cada cop més grolleres, és a dir, el nombre de classes és inversament proporcional al radi, ja que LINNEO implementa un mètode de classificació per boles on un element pertany a una certa classe si la seva distància al centre de la classe és menor que el radi. Quant més petit sigui el radi, més discriminarà.

Doncs bé, execucions de LINNEO amb radis diversos donen els resultats de la taula 4.2.

És a dir, com era d'esperar, les particions obtingudes amb un mètode i altre donat un cert nombre de classes n, *coincideixen* totalment. Sembla, per tant, que el criteri del **centroide** funciona de forma molt similar a una classificació per

Radi ∞	Fu	Sc	Go	Wa	Kl	Ka	Ho	To
2	Fu Sc Go Wa Kl Ka Ho							To
1.5	Fu Sc		Go Wa Kl Ka Ho					To
1	Fu	Sc	Go Wa Kl Ka Ho					To
0.9	Fu	Sc	Go Wa				Kl Ka Ho	To
0.7	Fu	Sc	Go Wa		Kl Ka		Ho	To
0.6	Fu	Sc	Go	Wa	Kl Ka		Ho	To
0.5	Fu	Sc	Go	Wa	Kl	Ka	Ho	To

Figura 4.2: Diferents particions dels compositors segons LINNEO.

boles, si es treballa en una mateixa mètrica.

Efectuar la mateixa anàlisi però treballant amb dades sense normalitzar porta a un arbre ascendent jeràrquic com el de la figura 4.3.

Els resultats coincideixen amb les classes del cas normalitzat per les particions més fines. A partir d'$n = 3$ hi ha ja alguna diferència. Ara Scherchen s'agrega al conjunt, a la classe central, enlloc d'assimilar-se a Furtwangler. I el mateix passa amb Toscanini en el següent pas, quedant ara Furtwangler com el més separat

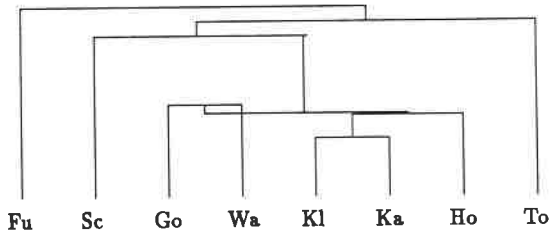


Figura 4.3: Compositors. Criteri del cetroide. Dades no normalitzades. Mètrica Euclídia ordinària.

Radi ∞					Fu	Sc	Go	Wa	Kl	Ka	Ho	To	
0.075			Fu										To
			Sc										:
			Go										:
			Wa										:
			Kl										:
			Ka										:
			Ho										:
0.060		Fu						Go					To
		Sc						Wa					:
		:						Kl					:
		:						Ka					:
		:						Ho					:
0.05	Fu		Sc					Go					To
	:		:					Wa					:
	:		:					Kl					:
	:		:					Ka					:
	:		:					Ho					:
0.035	Fu		Sc		Go						Kl		To
	:		:		Wa						Ka		:
	:		:		:						Ho		:
0.03	Fu		Sc		Go				Kl			Ho	To
	:		:		Wa				Ka				:
0.025	Fu		Sc	Go		Wa			Kl			Ho	To
	:		:	:		:			Ka			:	:
0.01	Fu		Sc	Go		Wa		Kl		Ka		Ho	To

Figura 4.4: Particions segons **LINNEO** sense normalitzar les dades.

del global.

Efectivament, observant la matriu de dades, la variable FINALE pren valors més elevats que les altres, quasi el doble, ja que el FINALE és el temps més llarg de la simfonia, i són precisament Scherchen i Furtwangler els que tenen per aquesta variable valors més diferents de la resta de compositors.

Doncs, és el temps dedicat al tercer moviment que, de fet, està guiant l'anàlisi. Pel que fa referència a **LINNEO**, els resultats que s'obtenen sota diferents radis, apareixen a la figura 4.4.

I es recupera l'arbre del cas normalitzat.

Sembla doncs que **LINNEO** ofereix més robustesa respecte de la normalització de les dades que **KLASS**, però, la realitat és que les particions no tenen un comportament monòton respecte del radi. És a dir, no és cert que hi hagi una partició constant en els intervals R_i i R_{i+1} ni que la partició i encaixi totalment en la següent [1.1].

Així, es poden trobar diferents particions per un mateix nombre de classes. La figura 4.5 mostra un exemple on Wand s'ha canviat de classe per a integrar-se

Radi	C-1	C-2	C-3	C-4	C-5	C-6
0.035	Fu	Wa	Ho	To	Sh	
		Go	Ka			
			Kl			
0.04	Fu	Go	Ho	To	Sh	
			Ka			
			Kl			
			Wa			

Figura 4.5: **LINNEO** Diferents particions de cinc elements en funció del radi.

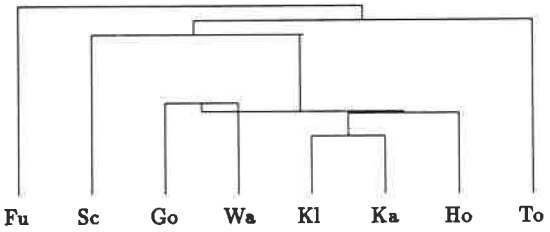


Figura 4.6: Compositors. Criteri del cetroide. Dades no normalitzades. Mètrica Euclídia Normalitzada.

abans al nucli principal de la classificació.

Si es normalitza la distància, a fi d'evitar efectes descompensats de les variables, **KLASS** organitza els vuit compositors de la mostra segons la figura 4.6, coincidint amb el que s'obté si no es fa aquesta normalització. Cal recordar aquí que la distància euclídia normalitzada equival a definir com a mètrica sobre els individus la inversa de la desviació tipus.

Sembla ser que el fet de normalitzar la distància euclídia no influeix en la classificació. Com ja s'ha comentat amb anterioritat, la mètrica euclídia normalitzada equival a treballar amb les dades dividides per la seva desviació tipus empírica. Efectivament, si hom estudia com són les desviacions de les variables de la matriu de dades, observarà que totes quatre són realment semblants, la qual cosa fa que treballar amb les dades així dividides o brutes sigui més o menys igual. La taula 4.7 exposa les desviacions tipus empíriques de les quatre variables.

Respecte de **LINNEO**, només s'hi contempla la primera de les normalitzacions, i no podem efectuar cap comparació per aquesta darrera mètrica.

Variable	ADAGIO	ALLEGRO	FINALE	SCHERZO
Desviació	.0168	.0148	.0102	.0099

Figura 4.7: Desviacions tipus de les variables.

Radi ∞								
2			Ca Us Br Ch Ar					Me : : : :
1.75		Ch Ar :			Ca Us Br			Me : :
1.6	Ar : :	Ch : :			Ca Us Br			Me : :
0.5	Ar :	Ch :	Br :			Ca Us		Me :
0	Ar	Ch	Br		Us		Ca	Me

Figura 4.8: Països classificats per LINNEO.

A la vista dels resultats, sembla que les classes més estables, les que es mantenen en totes les classificacions són les que agrupen Kleiber, Karajan i Horenstein per un cantó, Wand i Goehr per l'altra, deixant la resta de compositors en classes aïllades.

4.2 Paispet

Després de consultar un centena d'experts a efectes d'analitzar les possibilitats d'inversió de 43 països d'arreu del món, Hanner acaba descrivint-los via cinc factors de tipus econòmic, polític i financer, dos dels quals són qualitatius — la *situació actual*, que pot ser estable, bona o baixa, i la *tendència* en les inversions, que pot ser excellent, bona, estable o en baixa —, i la resta — *garantia política, d'execució de programes* i *financera* — quantitatius.

Aquest és un extracte de la mostra original, on únicament s'han considerat els països del continent americà, que ens permet d'estudiar un cas en mètrica mixta.

La taula 4.8 mostra les diferents particions que genera LINNEO en fer variar el radi de la classificació sobre les dades brutes.

n	Radi					
4	1.6	Ar	Ch	Ca Us [Br]		Me
	1.5	Ar	Ch		Ca Us	Me Br
2	2		Us Ca [Ar] [Br] Ch		[Me]	
	1.8		Us Ca Br Me		Ar Ch	

Figura 4.9: Diferents particions dels països en quatre classes.

De nou, en aquest cas s'observa el comportament no monòton de les dades respecte del radi, i es poden trobar particions diverses per un mateix nombre de classes. Així doncs, per un n fixat, amb radis molt propers hi ha particions diferents. La taula 4.9 mostra aquest fenòmen.

De fet, es poden considerar de forma conjunta totes aquestes particions, i el que sembla passar en créixer el radi és que, després de les dues primeres agregacions en que s'obtenen les classes USA-Canadà i Mèxic-Brasil hi ha una migració de Brasil a la classe d'USA-Canadà com a primer pas, segurament pel fet que en cada passa els centres de les classes es mouen i en créixer el radi Brasil queda atrapat en la primera. Es continua tenint una partició igual de fina, però amb diferent composició.

El següent pas és l'agregació d'Argentina i Xile, i s'obtenen tres classes. És justament abans del pas final que Mèxic es reuneix amb la classe majoritària i les dades queden bipartides en la darrera passa.

L'algorisme de normalització de la inversa dels rangs és comú a LINNEO i KLASS. El rang de totes les variables quantitatives d'aquest exemple és el mateix, ja que representen índexos totes elles. És per aquest motiu que els resultats obtinguts tant en LINNEO com en KLASS normalitzant o no les dades són idèntics.

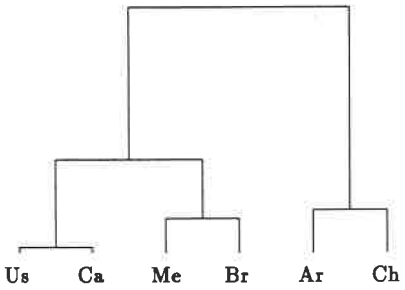


Figura 4.10: Països. Criteri del cetroide. Dades no normalitzades. Mètrica Euclídia Ordinària.

De totes maneres, KLASS normalitza amb la inversa de la desviació tipus la subdistància contínua quan es treballa en mètrica mixte, i no té gaire sentit definir dues mètriques de normalització sobre unes mateixes dades, així que únicament s'efectuarà aquí l'estudi sobre les dades sense normalitzar.

Les alternatives que ofereix KLASS en aquest cas són, doncs, utilitzar el criteri del **centroide** o el de **Ward** , que en aquest cas té més sentit que en el dels compositors.

Pel primer d'ells, KLASS genera l'arbre de classificació de la figura 4.10 .

L'enorme salt de la darrera agregació fa pensar que, en realitat, hi ha dos grans grups molt clars formats per USA, Canadà, Mèxic, Brasil, per una banda, i Argentina i Xile, per l'altra, partició que també es troba en LINNEO.

Pel criteri de **Ward**, l'arbre varia lleugerament, en el sentit que primer s'agreguen els quatre primers països de 2 en 2 i en una sola classe, i després s'agreguen Argentina i Xile, però de nou hi ha un molt considerable salt en el darrer pas i, per tant, la bipartició és patent aquí també.

De fet, KLASS efectua les fusions exactament en aquest mateix ordre sota el criteri del **centroide**, i no és fins que es representa gràficament l'arbre, considerant el índexos de nivell, que s'inverteix l'ordre de les dues agregacions intermitges.

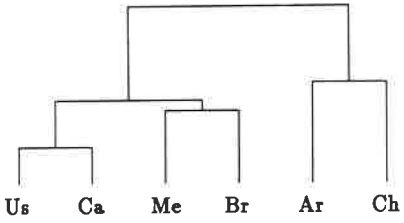


Figura 4.11: Països. Criteri de Ward. Dades no normalitzades. Mètrica Euclídia Ordinària.

La figura 4.11 té l'arbre generat per KLASS en aquest cas.

Convé comentar aquí que el fet que la relació de veïnatge sigui local, implica que l'arbre jeràrquic és totalment independent de l'individu amb el qual s'inicii la cadena de veïns. Així, en aquest exemple, començar per Argentina, que precisament forma part de la parella de veïns recíprocs menys propers fa que aquesta primera agregació tingui un índex de nivell més gran que les posteriors. Però l'arbre que es genera és el mateix que si es comença, per exemple, per Canadà que es troba en el cas contrari.

4.3 Esponges

Sembla ser que les esponges marines presenten greus problemes a l'hora d'organitzar-les en una taxonomia, i que actualment hi ha diverses classificacions d'aquests éssers vius acceptades per diferents escoles.

El departament de **Llenguatges i Sistemes Informàtics** de la **UPC** ha iniciat una estreta col·laboració amb el **Grup de Biologia Marina** del **Centre d'Estudis Avançats de Blanes**, un dels aspectes de la qual és aprofondir en l'estudi d'aquestes Porífers i tractar d'obtenir-ne una classificació per medis informàtics, en principi, més objectius que els purament humans.

El primer pas d'aquest procés ha estat treballar l'Ordre *Hadromerida* de la Classe *Demospongiæ* amb LINNEO i proposar una nova taxonomia a partir d'una mostra relativament gran i representativa d'aquest ordre [DOMI90].

En concret, es disposa d'una matriu de 76 espècies de 27 gèneres de l'Ordre *Hadromerida* descrites via 45 variables de naturalesa diversa, és a dir, quantitatives i qualitatives, essent algunes d'aquestes últimes binàries.

Les execucions que de KLASS s'han fet en aquest cas són amb dades normalitzades segons la inversa dels rangs, mètrica mixte, i els dos criteris d'agregació disponibles.

Aquest és un exemple clar de la necessitat d'incloure tractament de valors mancants en els mòduls estadístics: No és que el color sigui una característica massa important d'aquests animals. De fet, segons el lloc on es trobi i altres factors, una mateixa espècie pot canviar de color, però els espongiòlegs utilitzen aquesta característica, juntament amb d'altres, per a identificar l'espècie a la qual pertany una certa esponja. Algunes de les espècies incloses a la mostra o bé estan extingides o bé viuen prou amagades com per a no trobar-ne gaires, i les que es troben no s'aguanten gaire temps fora del seu hàbitat natural. Tot això fa que, d'algunes espècies, els pocs exemplars que en queden romanen conservats dins pots de formol en museus, laboratoris i altres llocs on els estudiosos recórren per saber d'aquests animals. El formol ha alterat el color original d'aquests exemplars, de manera que, en els casos en que no es disposa de prou bibliografia suplementària, no hi ha forma de definir aquesta característica de l'esponja. La casella corresponent de la matriu de dades contindrà un interrogant (?).

El nombre d'objectes a classificar és considerable, i s'han codificat per a facilitar la representació dels arbres jeràrquics. La taula 4.12 mostra aquesta codificació.

El que es pretén en aquest treball és veure quin és el resultat de KLASS per aquest mateix collectiu d'esponges.

La classificació obtinguda sota el criteri del **centroide** és a la taula 4.13, mentre que l'arbre jeràrquic generat segons el criteri de **Ward** és a la taula 4.14.

En ambdues taules es representa un eix vertical en el lateral esquerra que marca les unitats per tal que hom tregui, a simple vista, idea dels índexos de nivell corresponents a cada node. El codi corresponent a les diferents espècies de les fulles de l'arbre està posat en vertical. En els apèndixs es trobaran representacions més detallades d'aquests arbres.

Cap aquí fer una anàlisi a diferents nivells. En primer lloc es podria comparar directament el resultat que LINNEO genera amb el de KLASS per un determinat

Espècie	codi	Espècie	codi
Aaptos_aaptos	aa	Alectona_millari	am
Cliona_carteri	cca	Cliona_celata	cc
Cliona_labyrinthica	cl	Cliona_schmidti	cs
Cliona_viridis	cv	Diplastrella_bistellata	db
Diplastrella_ornata	do	Laxosuberites_ectyonimus	le
Laxosuberites_ferrerhernandezi	lf	Laxosuberites_rugosus	lr
Oxycordyla_pellita	op	Polymastia_agglutinaris	pa
Polymastia_conigera	pcn	Polymastia_corticata	pcr
Polymastia_ectofibrosa	pe	Polymastia_fusca	pf
polymastia_grimaldi	pg	Polymastia_hirsuta	ph
Polymastia_inflata	pi	Polymastia_infrapilosa	pip
Polymastia_invaginata	piv	Polymastia_littoralis	pl
Polymastia_mammillaris	pmm	Polymastia_martæ	pmt
Polymastia_polytylota	pp	Polymastia_radiosa	pra
Polymastia_robusta	pro	Polymastia_spinula	ps
Polymastia_tenax	pte	Polymastia_tissieri	pti
Polymastia_uberrima	pu	Prosuberites_epiphytum	pre
Prosuberites_longispina	prl	Prosuberites_rugosus	prr
Proteleia_sollasi	pso	Pseudosuberites_hyalinus	psh
Pseudosuberites_sulfureus	pss	Quasilina_brevis	qb
Quasilina_intermedia	qi	Quasilina_richardii	qr
Rhizaxinella_biseta	rb	Rhizaxinella_elongata	re
Rhizaxinella_pyriphera	rp	Rhizaxinella_uniseta	ru
Ridleya_oviformis	ro	Sphærotylus_antarcticus	sa
Sphærotylus_capitatus	sc	Spinularia_spinularia	ss
Spirastrella_cunctatrix	spc	Spirastrella_minax	spm
Stylocordyla_borealis	sb	Suberites_caminatus	suc
Suberites_carnosus_v.incrustans	sui	Suberites_carnosus_v.ramosus	sur
Suberites_carnosus_v.typicus	sut	Suberites_domuncula	sud
Suberites_ficus	suf	Suberites_gibbosiceps	sug
Tentorium_papillatus	tp	Tentorium_semisuberites	ts
Terpios_fugax	tf	Tethya_aurantium	ta
Tethya_citrina	tc	Timea_chondrilloides	tic
Timea_hallezi	tih	Timea_mixta	tim
Timea_stellata	tis	Timea_unistellata	tiu
Trachyteleia_stephensi	tra	trichostema_hemisphæricum	trh
Trichostema_sarsi	trs	Tylexocladus_joubini	tj
Weberella_bursa	wb	Weberella_verrucosa	wv

Figura 4.12: Índex d'espècies per ordre alfabètic.

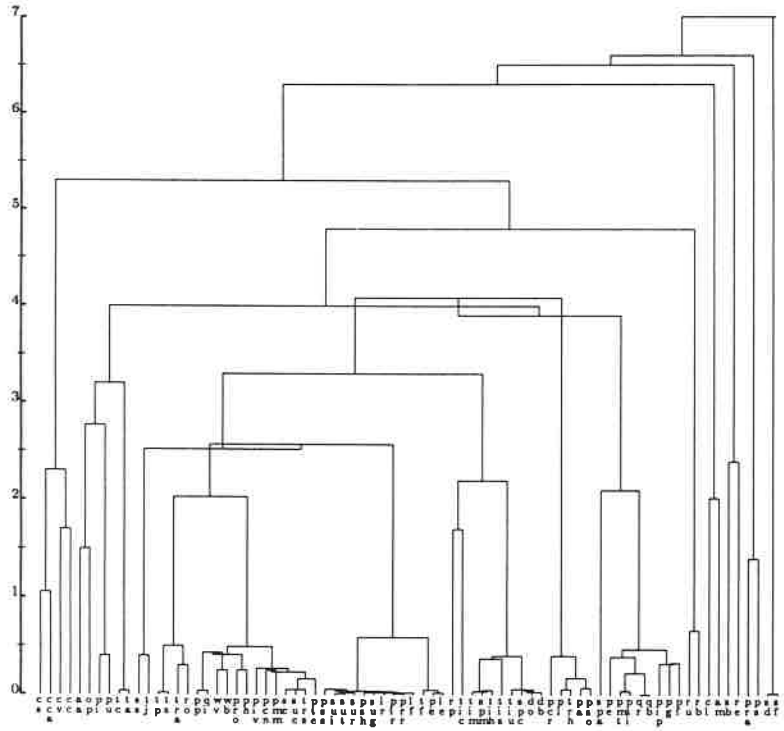


Figura 4.13: Classificació de les esponges segons el criteri del centroide

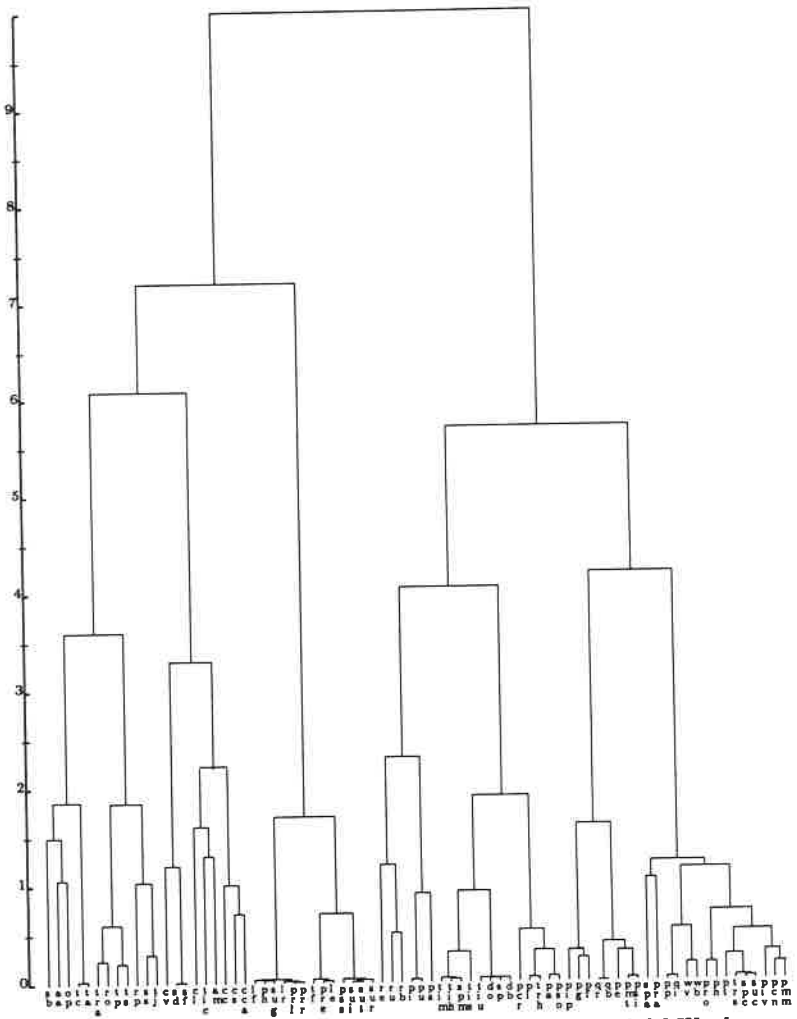


Figura 4.14: Classificació de les esponges segons el criteri del Ward

nombre de classes.

Fer un estudi acurat amb les dimensions d'informació que s'estan tractant és força complexe i podria ser la base de tot un altre treball. El que sí que es pot fer és analitzar globalment els dos arbres de classificació mirant d'emmarcar-los en la classificació proposada en [DOMI90], que no conté exactament les classes generades per LINNEO, sinó que s'han efectuat alguns canvis a partir d'un raonament *a posteriori* per part d'experts en el tema.

Observant l'arbre de classificació de la taula 4.13, el tret més destacat és que ja per nivells del tall molt propers a zero ($\alpha < 0.5$) apareix una classe molt compacta que és la que conté les espècies (veure 4.12):

pss sui sut sur psh sug lr prr lf tf pre le

grup que també es manté en l'arbre 4.14 si bé amb una estructura un xic diferent.

Efectivament, aquesta classe agrupa les espècies de la família *Suberitidæ* a excepció de *Suberites ficus* i *Suberites domuncula*, que a l'arbre 4.13 apareixen totalment aïllades i amb molta diferència respecte de la resta d'esponges, amb alta proximitat entre elles.

El que succeeix és que aquestes dues espècies allotgen bernat ermità, i són les úniques que presenten aquesta característica, la qual cosa les distingeix enormement de totes les altres.

LINNEO també genera una classe amb aquestes dues esponges i una altra amb totes les *Suberites* restants (gèneres *prosuberites, laxosuberites, pseudosuberites* i *suberites*). En la classificació final de [DOMI90] apareixen totes en una sola família.

L'única excepció és l'espècie *Suberites caminatus*, que en tots els casos surt en un altre grup. I de fet, aquesta és una espècie conflictiva que al llarg de la bibliografia ha anat canviant de gènere[1] perquè és força diferent de la resta d'aquesta família. Si hom observa els arbres de classificació veurà que és amb el grup de les *Tentorium* que s'uneix aquesta esponja.

Un segon bloc compacte a nivell de tall $\alpha = 0.5$ (sempre fent referència a l'arbre 4.13) es veu en detall en els gràfics dels apèndixs i agrupa alguns gèneres

[1] En algun moment apareix amb el nom de *Tentorium caminatus*.

de la família *Polymastiidæ* com són *Polymastia* i *Weberella*, que més amunt (i.e. $\alpha = 2$) s'agrupen amb *Tentorium*.

En aquesta anàlisi, però, apareixen espècies del gènere *Polymastia* repartits en diferents subarbres de forma més o menys dispersa. Aquest és un dels gèneres més ben estudiats i, per tant, un dels que es disposa de més informació. D'una banda, gran part dels descriptors utilitzats com a variables de l'anàlisi es deuen a espècies d'aquest gènere, que són les úniques en les que aquestes propietats tenen algun tipus de rellevància. De l'altra, aquest també és el gènere amb més representació en la mostra d'estudi, i pot ser el que s'està classificant més, figurant algunes de les espècies realment separades de les altres, quan en realitat sempre s'estudien totes juntes. Per tant, tractarem aquest gènere d'una forma més general, sense estudiar en detall tots els exemplars de la mostra, sinó el comportament del conjunt.

LINNEO ha generat dues classes composades majoritàriament per *Polymastia*; una d'elles en la seva totalitat, i l'altre incorporant els gèneres *Sphærotylus* i *Prolleteia*.

En l'arbre 4.13 també es troben dues classes de *Polymastia*. Una d'elles conté també *Weberella*; l'altra incorpora *Quasilina*, i *Sphærotylus* queda repartit en les dues classes.

Sota el criteri de **Ward**, però (taula 4.14), si bé també s'identifiquen dues classes d'aquest gènere per α's petites (aproximadament de 0.25), no són aquests dos grups separats, sinó que en créixer α passen a formar un únic grup amb tota la família, tal com es proposa a [DOMI90]. Les dues classes d'aquest cas contenen, per una banda *Weberella* i *Sphærotylus*, i per l'altra les *Quasilina*.

En la classificació tradicional, els gèneres *Polymastia, Sphærotylus, Weberella, Tentorium, Quasilina, Ridleya, Spinularia, Trichostema, Trachiteleia* i *Tyzelocladus* formen la família *Polymastiidæ*. Aquest és un grup molt ampli en el que hi ha gèneres que s'assemblen realment molt, i també alguns que, compartint gran part de les seves característiques amb les del grup (*Weberella, Tentorium, Quasilina* i *Trichostema*), són un xic més diferents del conjunt.

LINNEO genera classes a part pels gèneres *Weberella* i *Quasilina* d'una banda, i *Tentorium* i *Trachyteleia* de l'altra. Efectivament, els dos primers presenten un còrtex més senzill que la resta de gèneres d'aquest grup, i els segons tenen per papiles simples protuberàncies de l'estructura externa, a diferència de la resta,

les papiles dels quals tenen una estructura interna més elllaborada.

KLASS separa en dues classes que també contenen altres gèneres de la família *Polymastiidæ* el gèneres *Weberella* i *Quasilina*, però en tots els casos manté els gèneres *Tentorium* i *Ridleya* en una classe separada, i, sota el criteri de Ward, oposada a totes les altres *Polymastiidæ*.

En un altre grup, es troba la F.*Timeidæ*. De fet, la classificació tradicional presenta en aquesta família els gèneres *Timea, i Diplastrella*, mentre que *Spirastrella* forma una família apart. El bloc que presenta KLASS en tots els criteris agrupa aquests tres gèneres. Es creu que l'espícula principal de les *Diplastrella* correspon a la fusió de dues del tipus de les *Timea*, i per això s'agrupen aquests dos gèneres en una classe. A [DOMI90] s'argumenta com es pot considerar que l'espícula principal de les *Diplastrella* s'assembla a la de *Spirastrella*, i es proposa una família per aquests dos gèneres i una altra per *Timea*. Efectivament, LINNEO, executat amb radi 12, presenta una classe aïllada per les esponges d'aquest gènere, però si s'estudia la distància que presenten les espècies de *Timea* a altres classes, es veu que la classe que conté *Spirastrella* i *Diplastrella* inclou totalment la de les *Timea*. Cal recordar que LINNEO no genera una partició dels objectes, sinó que les classes que proposa poden intersecar entre elles.

A nivell d'$\alpha = 2.5$, de l'arbre 4.13, ja es troba formada la classe de les *Cliona*, família que també es manté en [DOMI90], incloent els gèneres *Cliona i Alectona*.

El criteri de **Ward** genera aquesta classe directament, mentre que el del centroide separa les espècies *Allectona millari i Cliona labirynthya* en una classe a part.

En últim lloc, es proposa la família *Tethyidæ*, amb les espècies *T.citrina i T.aurantium* que també apareixen en LINNEO i KLASS.

L'estructura que s'acaba de raonar suggereix un tall de l'arbre 4.13 a nivell 2.4 aproximadament, i del 4.14 a un nivell 2.3. De les espècies que queden en classes aïllades, els experts haurien de mirar com afectar-les a un o altre grup, o si es modifica el nivell del tall..., feina que queda ja fora de l'àmbit d'aquesta tesina, i de l'informàtic en general.

Com a darrer comentari dir que l'arbre enregistra dues espècies amb distància 0 entre elles, és a dir, idèntiques. Això, en principi no hauria de ser així, ja que no hi ha consciència d'haver introduït dues files d'iguals components a la matriu de

dades. S'han mirat les files de la matriu corresponents a aquestes dues esponges, que són *Quasilina brevis i Quasilina ricardii*. Tal i com s'havia suposat, aquestes dues esponges no són iguals des del punt de vista de les seves components, sinó que es distingeixen únicament pel color, desconegut en el primer cas, i de la categoria *altres* en el segon[2]. L'algorisme de tractament de valors mancants, casualment ha substituit per *altres*, també, el color de *Quasilina brevis*, convertint en la mateixa dues esponges que en principi no tenien la mateixa descripció en termes de les variables que intervenien en l'anàlisi.

De tota manera, el fet que per totes les altres propietats aquestes dues espècies tinguin els mateixos valors, les continua mantenint prou properes com per a que la classificació resultant no quedi alterada per aquest fet.

Una discussió en termes biològics més profunda del problema de la classificació taxonòmica de les esponges, en particular de l'Ordre *Hadromerida* es pot trobar en el treball ja referit [DOMI90].

[2]La propietat color es tracta com una variable categòrica amb quatre modalitats que són *groc_palid, blau_o_taronja_intens, altres, desconegut*.

Capítol 5

Conclusions i línies obertes

5.1 El llenguatge d'implementació

La primera observació a fer és que KLASS constata que LISP és un bon llenguatge de programació per a sistemes de tractament estadístic de dades, almenys a nivell experimental, sobretot pel que fa a la informació de tipus qualitatiu.

Tradicionalment, el llenguatge d'implementació per excel·lència del món estadístic ha vingut essent FORTRAN, com ja s'ha citat repetides vegades al llarg d'aquest document. L'avantatge que té LISP sobre ell és que, per ser funcional i eminentment recursiu, la seva legibilitat és netament superior, al igual que la facilitat per programar-hi.

A continuació es presenta el cos de l'algorisme principal de classificació ascendent jeràrquica per veïns recíprocs encadenats en les seves versions LISP (figura 5.1) i FORTRAN (figura 5.1) respectivament, i hom observarà que, mentre la darrera requereix un estudi profund de l'algorisme per a ser compresa, la primera resulta força més intuïtiva[1].

De fet, el que en realitat passa és que l'estructura iterativa DO, una de les més utilitzades en les implementacions FORTRAN després de l'IF, és intuïtivament poc clara i a la vista del codi no es capta fàcilment una idea del que fa l'algorisme. Pel contrari, la representació simbòlica que en llenguatges funcionals, entre els quals figura LISP, es pot fer dels processos, permet entendre a primer cop d'ull el que fa un algorisme més fàcilment.

[1]La versió FORTRAN correspon a la implementació que CYSIA integra en el paquet estadístic SPAD d'àmplia difusió en l'actualitat.


```
; CLASSIFICACIO ASCENDENT JERARQUICA PER VEINS RECIPROCS.
; recerca en cadena

;Utilititza formules.lsp general.lsp interfase.lsp lefec.lsp matriu.lsp
;veins.lsp
;Crida amb L_obj_creados

;L'entrada al classificador sera la llista d'objectes a classificar :
;(id_obj1 id_obj2 id_obj3...), un parametre que indicara quina metrica
;utilitzarem, i tres d'opcionals per al criteri d'agregacio i constants
;de ponderacio de subdistancies

(defun veins_reciprocs (llista td
                        &key (alfa 1) (beta 1) (criteri 'centroide))
       (vri llista td nil (car llista)
            (matriu_distancies td llista alfa beta criteri)
            alfa beta criteri
       )
)

;vri: Funcio interna per a fer la classificacio: indica l'objecte a clas-
;sificar i l'ordre com s'han classificat els anteriors, que permetra
;recular un pas en la classificacio quan es genera una nova subclasse.

(defun vri (llista td anteriors objecte matr_distancies alfa beta criteri)
       (if (null (cdr llista)) llista
           (classificar llista td anteriors objecte matr_distancies
                        alfa beta criteri
           )
       )
)


(defun classificar (llista td anteriors element matr_distancies
                    alfa beta criteri)
  (let* ((vei     (mes_proper element))
         (vei_rec (mes_proper vei    ))
        )
        (if (or (equal element vei_rec) (equal vei (car anteriors)))
            (agregar element vei
                     (treure vei anteriors)
                     llista td matr_distancies alfa beta criteri
            )

            (vri llista td (cons element anteriors) vei matr_distancies
                 alfa beta criteri
            )
        )
   )
)
```

Figura 5.1: Versió LISP de l'algorisme principal.

```
      SUBROUTINE CAHVR (COOR,NAXU, JAINE,JBENJ,NEFF,POIDS,VNIV,JELEM,
     -                  JINDI,NINDI, KHIST, PTOTI, VTOT, NIVED)
C************************************************************************
C*  C.LASSIFICATION A.SCENDANTE H.IERARCHIQUE - V.OISINS R.ECIPROQUES  *
C*                                                                      *
C* ENTREES  COOR(*,*)... COORDONNEES DES INDIVIDUS                      *
C* -------  NAXU........ NOMBRE D'AXES UTILISES                        *
C*          POIDS(*).... POIDS RELATIFS DES INDIVIDUS                   *
C*          NINDI....... NOMBRE D'INDIVIDUS A AGREGER                   *
C*          PTOTI....... POIDS TOTAL DES INDIVIDUS                      *
C*          NIVED....... NOMBRE DE NOEUDS A EDITER                      *
C*                                                                      *
C* SORTIES  JAINE(*).... ADRESSES DES AINES                             *
C* -------  JBENJ(*).... ADRESSES DES BENJAMINS                         *
C*          NEFF(*)..... EFFECTIFS DES NOEUDS                           *
C*          POIDS(*).... POIDS ABSOLUS DES NOEUDS                       *
C*          VNIV(*)..... INDICES DE NIVEAU EN ORDRE CROISSANT           *
C*          VTOT........ SOMME DES INDICES DE NIVEAU                    *
C*                                                                      *
C* TRAVAIL  COOR(*,*)... COORDONNEES DES ELEMENTS PENDANT L'AGREGATION  *
C* -------  NEFF(*)..... EFFECTIFS       ---       ---       ---       *
C*          POIDS(*).... POIDS RELATIFS  ---       ---       ---       *
C*          VNIV(*)..... DISTANCE ENTRE VOISINS DANS LA CHAINE          *
C*          JELEM(*).... 1. ADRESSES DES ELEMENTS PENDANT L'AGREGATION  *
C*                       2. ADR. : NOEUDS TRIES --> VECTEURS NON TRIES  *
C*          JINDI(*).... 1. CHAINE DES VOISINS - RESTE A AGREGER        *
C*                       2. ADR. : VECTEURS NON TRIES --> NOEUDS TRIES  *
C*          KHIST(*).... LIGNE D'ETOILES POUR L'HISTOGRAMME             *
C*                                                                      *
C* FICHIER  NBAND ..... (T) 1. EFFECTIF, POIDS ET INDICE DES NOEUDS     *
C* -------                  2. DESCRIPTION DE LA HIERARCHIE             *
C*                                                                      *
C* APPELS   SHERI, SHELI.                                               *
C* ------                                                               *
C************************************************************************
C-F77 IMPLICIT CHARACTER*4 (K)
      DIMENSION COOR(NINDI,NAXU),JAINE(NINDI),JBENJ(NINDI),NEFF(NINDI),
     -          POIDS(NINDI),VNIV(NINDI),JELEM(NINDI),JINDI(NINDI),
     -          KHIST(85)
      COMMON /SEUIL/ ZERO,RMAX
      COMMON /ENSOR/ LEC,IMP
      COMMON /AUX/   NXLPA,NLIMP

      DATA  KETOI /1H*/
C ======= INITIALISATIONS ===============================================
        NBAND = 25
ZERO = 0
RMAX = 0.99999E+30
NXLPA = MOD(NIVED+10,62)
        DO 10  I = 1,85
   10   KHIST(I) = KETOI
```



```
      REWIND NBAND
        DO 20  I = 1,NINDI
POIDS(I)=POIDS(I)/PTOTI
C Passem els pesos absoluts a relatius
        JELEM(I) = I
        JINDI(I) = I
   20   NEFF(I)  = 1
      VTOT = 0.
      INOEU = 0
      NCHAI = 1
      NREST = NINDI
      NIND1 = NINDI - 1
C ======= RECHERCHE DES NOEUDS DE LA HIERARCHIE ========================
   30 IF (NCHAI .GT. 1) GOTO 40
C     ------- LA CHAINE EST VIDE ---------------------------------
        NCHAI = 1
        DPRE = RMAX
        GOTO 50
C     ------- LA CHAINE N'EST PAS VIDE ---------------------------
   40   IPRE = JINDI(NCHAI-1)
        DPRE = VNIV(IPRE)
C   +++++++ CHARGEMENT DE LA FIN DE LA CHAINE ++++++++++++++++++++++
   50 ICUR = JINDI(NCHAI)
      DCUR = RMAX
C   +++++++ RECHERCHE DU PLUS PROCHE DANS LA QUEUE ++++++++++++++++++
      IDQUE = NCHAI + 1
      IF (IDQUE .GT. NREST) GOTO 80
      PCUR = POIDS(ICUR)
        DO 70  IND = IDQUE,NREST
        ITES = JINDI(IND)
        SPOI = PCUR + POIDS(ITES)
        IF (SPOI .GT. 0.0) GOTO 55
          COEF = 0.0
          GOTO 57
   55   COEF = PCUR * POIDS(ITES) / SPOI
   57   DTES = 0.
          DO 60  IAX = 1,NAXU
          DIFF = COOR(ICUR,IAX) - COOR(ITES,IAX)
          DTES = DTES + COEF * DIFF * DIFF
          IF (DTES .GE. DCUR) GOTO 70
   60     CONTINUE
        DCUR = DTES
        IPRO = IND
   70   CONTINUE
      VNIV(ICUR) = DCUR
      IF (DCUR .LT. DPRE) GOTO 100
C   +++++++ ICUR ET IPRE SONT DES VOISINS RECIPROQUES +++++++++++++++
   80 INOEU = INOEU + 1
      JAINE(INOEU) = JELEM(ICUR)
      JBENJ(INOEU) = JELEM(IPRE)
```

```
C     ------- CALCULS SUR LE NIVEAU D'AGREGATION -----------------
      IF (DPRE .LE. ZERO) DPRE = 0.
      VTOT = VTOT + DPRE
C     ------- EFFECTIF, POIDS ET INDICE SUR NBAND ----------------
      NEFF(IPRE) = NEFF(ICUR) + NEFF(IPRE)
      JELEM(IPRE) = NINDI + INOEU
      PNOEU = POIDS(ICUR) + POIDS(IPRE)
      WRITE (NBAND) NEFF(IPRE), PNOEU, DPRE

      IF (INOEU .EQ. NIND1) GOTO 110
C     ------- COORDONNEES ET POIDS DE (IPRE U ICUR) --> IPRE -----
      PPRE = POIDS(IPRE) / PNOEU
      PCUR = POIDS(ICUR) / PNOEU
        DO 90  IAX = 1,NAXU
   90   COOR(IPRE,IAX) = PCUR*COOR(ICUR,IAX) + PPRE*COOR(IPRE,IAX)
      POIDS(IPRE) = PNOEU
C     ------- MISE A JOUR DE LA QUEUE ET DE LA CHAINE ------------
      JINDI(NCHAI) = JINDI(NREST)
      NREST = NREST - 1
      NCHAI = NCHAI - 2
      GOTO 30
C   +++++++ ALLONGEMENT DE LA CHAINE +++++++++++++++++++++++++++++++
  100 NCHAI = NCHAI + 1
      IF (NCHAI .EQ. IPRO) GOTO 30
      IPIV         = JINDI(NCHAI)
      JINDI(NCHAI) = JINDI(IPRO)
      JINDI(IPRO)  = IPIV
      GOTO 30
C ======= MISE EN ORDRE DE LA HIERARCHIE ==============================
  110 REWIND NBAND
C   +++++++ CLASSEMENT DES INDICES DE NIVEAU ++++++++++++++++++++++++
        DO 120  IN = 1,NIND1
        JELEM(IN) = IN
  120   READ (NBAND) NEFF(IN), POIDS(IN), VNIV(IN)
      CALL SHERI (NIND1,VNIV, JELEM)
C   +++++++ REMISE EN ORDRE DES EX-AEQUO ++++++++++++++++++++++++++++
      IPRE = 1
      VPRE = VNIV(1)
        DO 130  ICUR = 2,NIND1
        IF (VPRE .EQ. VNIV(ICUR)) GOTO 130
        IDEB = IPRE
        NEGA = ICUR - IPRE
        IPRE = ICUR
        VPRE = VNIV(IPRE)
        IF (NEGA .GT. 1) CALL SHELI (NEGA,JELEM(IDEB))
  130   CONTINUE
C   +++++++ ON INVERSE JELEM(*) DANS JINDI(*) ++++++++++++++++++++++
        DO 140  IE = 1,NIND1
        IN = JELEM(IE)
  140   JINDI(IN) = IE + NINDI
```



```
C   +++++++ SUIVI DES ADRESSES DES SUCCESSEURS ++++++++++++++++++++++
      REWIND NBAND
        DO 150  IN = 1,NIND1
        IAV = JELEM(IN)
        IAINE = JAINE(IAV)
        IBENJ = JBENJ(IAV)
        IF (IAINE .GT. NINDI) IAINE = JINDI(IAINE-NINDI)
        IF (IBENJ .GT. NINDI) IBENJ = JINDI(IBENJ-NINDI)
        INOEU = IN + NINDI
  150   WRITE (NBAND) INOEU,IAINE,IBENJ,NEFF(IAV),POIDS(IAV),VNIV(IN)
C ======= EDITION ET RETOUR AUX POIDS ABSOLUS ==========================
      IF (NIVED .EQ. 0) GOTO 160
      IF (NIVED .GE. NIND1) WRITE (IMP,1100)
      IF (NIVED .LT. NIND1) WRITE (IMP,1200) NIVED
      WRITE (IMP,1300)
      NLIMP = 7
  160 IDEB = NIND1 - NIVED + 1
      REWIND NBAND
      VMAX = VNIV(NIND1)
        DO 180  IN = 1,NIND1
        READ (NBAND) INOEU,JAINE(IN),JBENJ(IN),NEFF(IN),PNOEU,VNIV(IN)
        PNOEU = PNOEU * PTOTI
        POIDS(IN) = PNOEU
        IF (IN .LT. IDEB) GOTO 180
        NLIMP = NLIMP + 1
        IF (NLIMP .LE. NXLPA) GOTO 170
          WRITE (IMP,1000)
          WRITE (IMP,1300)
          NLIMP = 4
  170   NETOI = 80. * VNIV(IN) / VMAX + 1.
        WRITE (IMP,1401) INOEU,JAINE(IN),JBENJ(IN),NEFF(IN),PNOEU,
     -                           VNIV(IN),(KHIST(I),I=1,NETOI)
  180   CONTINUE
      WRITE (IMP,1500) VTOT
      NLIMP = NLIMP + 3
      RETURN
C------------------------- FORMATS ---------------------------------
 1000 FORMAT (1H1)
 1100 FORMAT (1H1,130(1H-)/5X,26H CLASSIFICATION ASCENDANTE,
     -        38H HIERARCHIQUE : DESCRIPTION DES NOEUDS/1X,130(1H-)/1H0)
 1200 FORMAT (1H1,130(1H-)/5X,26H CLASSIFICATION ASCENDANTE,
     -        31H HIERARCHIQUE : DESCRIPTION DES,I4,7H NOEUDS,
     -        26H D'INDICES LES PLUS ELEVES/1X,130(1H-)/1H0)
 1300 FORMAT (43H  NUM. AINE  BENJ   EFF.    POIDS    INDICE,
     -        12X,34H HISTOGRAMME DES INDICES DE NIVEAU/)
C1400 FORMAT (1H ,I4,3I6,F10.2,F10.5,3X,85A1)
 1401 FORMAT (1H ,I4,3I6,F10.2,2X,G11.5,1X,85A1)
 1500 FORMAT (1H0/33H    SOMME DES INDICES DE NIVEAU =,G11.5,/)
C-----------------------------------------------------------------------
      END
```

Figura 5.2: Versió **FORTRAN** de l'algorisme principal.

Dues són les principals raons d'aquesta diferència. Per una banda, els llenguatges imperatius requereixen flotes senceres de variables, cada una d'elles amb una semàntica associada, sense el coneixement de la qual es fa impossible la comprensió de l'algorisme. Hom ha de gastar una quantitat important de temps per veure, a partir del propi codi, què representa cada variable, quan moltes vegades, la immensa majoria són auxiliars i únicament computen resultats intermedis. D'altra banda, l'accés a certes estructures de dades imperatives fa ús d'una sintaxi específica que cal conèixer també per entendre, i que en alguns casos no és, en absolut, trivial. A la pràctica, el codi funcional d'un procés s'entén de forma natural amb relativa rapidesa.

Quant a temps d'execució, el compilador de LISP s'encarrega de traduir a formes iteratives considerablement eficients les funcions recursives per la cua, de forma que el codi generat tingui un comportament més que acceptable. De fet, hi ha tota una teoria que justifica que, d'una banda, el disseny recursiu resulta molt més fàcil per la ment humana que l'iteratiu, ja que es basa en un principi de descomposar un problema en altres similars i més petits, i de l'altra, proporciona mecanismes de transformació automàtica recursiu-iteratiu assegurant un algorisme final més eficient que el que un programador faria directament en versió iterativa.

Això disposant a més d'un codi fàcilment comprensible a primera vista.

5.2 KLASS i LINNEO

Una de les aplicacions de KLASS és validar LINNEO des del punt de vista estadístic, la qual cosa és beneficiosa per als investigadors que no confien en les tècniques d'Intel·ligència Artifial.

En línies generals, s'observa que la classificació que fa LINNEO és força similar a la de KLASS, si aquest treballa amb el criteri del centroide i mètrica euclídia.

Quant a temps d'execució, no és convenient fer comparacions de moment, ja que, mentre KLASS té una implementació eminentment recursiva, LINNEO s'ha programat iterativament, i el codi que genera el compilador de LISP en un cas i altre no té res a veure, de forma que les mateixes estructures tenen temps d'execució molt diferents.

En dominis poc estructurats, dels que se'n té molta informació, però que no

s'organitza de forma natural, KLASS és una eina útil per a estudiar de quina forma es pot estructurar aquest coneixement. Aquest és, per exemple, el cas de les esponges presentat en l'apartat 4.3 o dels directors d'orquestra de 4.1. Si bé es disposa de molta informació sobre les esponges o els directors , no existeix una classificació natural ni clara de cap d'aquests dos col·lectius. KLASS pot ajudar a identificar trets característics de certs subgrups d'un domini d'aquest estil.

Quant a la qualitat de la classificació obtinguda, en els capítols introductoris d'aquest treball ja es parla de la dificultat d'avaluar el resultat generat per un algorisme de classificació, especialment quan no es disposa d'una classificació de referència externa, i de fet, aquesta és la situació. Cal que un expert validi, utilitzant el seu coneixement del domini, la classificació proposada per KLASS. El que sí que s'ha pogut observar és que característiques molt rellevants de certs objectes són decisives a l'hora de construir l'arbre jeràrquic. A títol d'il·lustració esmentar l'exemple del bernat ermità en les esponges (secció 4.3).

Tot seguit s'esmenten les possibles línies de treball que KLASS deixa obertes per un futur.

5.3 Línies futures

Algunes d'aquestes línies es desenvoluparan en l'entorn del departament d'**Estadística i Investigació Operativa** d'aquesta **Facultat d'Informàtica de Barcelona.**

5.3.1 Interfície d'usuari

Una forma d'assolir un sistema de fàcil ús és construir una capa externa entre el programa i l'usuari, i que sigui a través d'ella que l'usuari utilitzi KLASS. De fet, es tractaria de dissenyar un senzill sistema de menús que faci les preguntes pertinents per obtenir els paràmetres necessaris per a llençar l'execució de KLASS (família de fitxers de dades, mètrica de treball, càlcul automàtic dels valors de les constants de ponderació, ...).

5.3.2 Valors mancants

En aquest treball s'ha utilitzat un criteri de substitució per valors centrals dels valors mancants, sota la hipòtesi que les dades numèriques de la matriu de dades

s'ubicaven en l'interval $[0,1]$.

En un cas més general, es pot extendre aquest tractament al cas en que les variables tinguin un rang qualsevol, i substituir el valor perdut per la mitjana de la seva columna en el cas de les variables numèriques, i per una distribució uniforme, en el cas de les variables qualitatives.

Així, donada la matriu

$$\begin{pmatrix} o_{11} & \dots & o_{1k} & \dots & o_{1l} & \dots & o_{1n} \\ \dots & & & & & & \\ o_{i1} & \dots & ? & \dots & -1 & \dots & o_{1n} \\ \dots & & & & & & \\ o_{n1} & \dots & o_{nk} & \dots & o_{nl} & \dots & o_{nn} \end{pmatrix}$$

i suposant que k és una variable qualitativa de modalitats $(m_1 \dots m_c)$, el valor que s'hauria de donar a l'element o_{ik} de la matriu és $((m_1\ \frac{1}{c}) \dots (m_c\ \frac{1}{c}))$. D'altra banda, si l fos una variable numèrica, el valor pel qual quedaria substituït l'element o_{il} és la mitjana aritmètica dels valors no nuls de la variable l.

5.3.3 Classificació ponderada

Pot ésser interesant l'assignació de pesos als individus que hom vol classificar, per casos on hi hagi objectes més importants que d'altres, per a contrarestar l'efecte d'un cert objecte, o per a utilitzar mostres on part de la població queda sobrerepresentada sense que això determini el resultat de la classificació.

KLASS està dissenyat de forma que l'assignació de pesos no modifiqui més que localment algunes de les rutines. En concret, ja es disposa d'una propietat assignada a cada objecte, que és el seu pes, i només cal introduir-lo en aquelles fòrmules de càlcul de distàncies, o pseudodistàncies i centres de gravetat, que es modifiquin quan el pes dels objectes no és constant.

5.3.4 Paràmetres de l'algorisme

Aquest algorisme disposa de dos graus de llibertat que són la mètrica de treball i el criteri d'agregació. Pel propòsit d'aquest treball s'ha considerat suficient la implementació de dos criteris d'agregació i tres mètriques fonamentals, que es despleguen en cinc quan s'introdueixen conceptes de normalització. Ara

bé, l'algorisme ha estat dissenyat mantenint, en tot moment la possibilitat d'augmentar-ne la potència a base d'afegir-hi més mètriques i criteris amb relativa facilitat. De fet, tant unes com altres han estat tractades al llarg de tot el programa com a paràmetres, i en punts molt localitzats s'ha discriminat segons el valor que tenien. Així, la mètrica únicament és rellevant a l'hora de calcular una nova fila de la matriu de distàncies, mentre que el criteri afecta al càlcul de coordenades del centre de gravetat d'una nova classe. La resta de l'algorisme és independent del valor d'aquests paràmetres. En particular, podria ser interessant d'implementar el criteri de les distàncies mitges, que es comporta força bé.

5.3.5 Anàlisi de l'arbre de classificació

KLASS acaba un cop ha construït l'arbre de classificació a partir de la mostra. El que li queda a l'expert per decidir és el nivell de l'α-tall que definirà la partició definitiva dels objectes classificats.

Els criteris de decisió que normalment s'utilitzen tracten d'obtenir la partició que millor compromet la distància entre classes (convé que sigui gran, la qual cosa significaria que cada classe està realment separada de les altres), amb la distància entre objectes d'una classe (convé que sigui petita, per garantir que els objectes de la classe s'assemblen prou).

Es tractaria d'identificar de forma automàtica aquest nivell del tall, bé a base de calcular relacions de distàncies entre classes i dins de les classes, que no és gens trivial, o bé via algun criteri inspirat en altres fonts que no siguin les purament estadístiques.

Per exemple, LINNEO treballa amb un radi de classificació com a paràmetre, que opcionalment el propi sistema estima utilitzant la distància mitja d'una mostra aleatòria de grandària el 10% dels objectes a classificar. De la mateixa forma, es podria introduir un nou paràmetre en KLASS que fos el nivell del tall, la qual cosa potser seria un xic perillosa considerant que, *a priori* no es coneix el rang en que es mouran les distàncies entre classes ni, per tant, els índexos de nivell, i el valor d'α s'hauria de *preconitzar* de forma arbitrària.

Una tercera alternativa consisteix a imposar, no el nivell del tall, sinó el nombre de classes que es volen obtenir, i això ja és més raonable, perquè hom pot decidir el nombre de classes en funció de la grandària de la mostra, i l'obtenció del nivell del tall a partir del nombre de classes que es volen és relativament senzilla.

Recordant que a cada agregació es disminueix en una unitat el cardinal de la partició, una partició en n classes s'obtindria després de fer $2n_0 - n$ agregacions, essent n_0 la grandària de la mostra, i suposant que no hi ha cap cas d'inversió. I, per tant, el nivell del tall pot ser qualsevol entre els índexos de nivell de les agregacions $2n_0 - n$-èssima i $2n_0 - n + 1$-èssima.

En definitiva, KLASS deixa obert tot un ventall de possibilitats respecte de l'estudi i comparació de classificacions de grans bancs de dades.

Bibliografia

[ALUJ84] Aluja, T., "*Mètodes de classificació i anàlisi factorial sobre un graf: Aplicació a l'anàlisi de dades municipals de Catalunya: Contribució a l'estudi de la divisió territorial*",edició de la UPB, servei de publicacions cpda-etseib, Barcelona, 1984.

[BAIM88] Baim, P. W. "*A method for attribute selection in inductive learning systems*". IEEE Transaction on pattern analysis and machine intelligence 10 (6), (1988) pp 888–896.

[BENZ76] Benzecri, J.P., "*Histoire et prehistoire de l'Analyse des donnees*", Les cahiers de l'Analyse des donnees, vol.1 n. 1 à 4, Ed.Dunod, Paris, 1976.

[CLAN81] Clancey W. J., "*The epistemology of a rule-based expert system: A framework for explanation*". Dep. Comput. Sci., Stanford Univ., Rep. STAN-CS-81-896, (1981).

[DOMI90] Domingo i Gou, M. "*Aplicació de tècniques d'IA (LINNEO) a la classificació sistemàtica. O. HADROMERIDA (DEMOSPONGIÆ-PORÍFERA)*". UB, Departament d'Ecologia. Barcelona, 1990. Tesi de llicenciatura.

[EVER74] Everitt, B., "**Cluster analysis**",ed. Heinemann Educational Books Ltd, London, 1974.

[HAYE84] Hayes-Roth, F. "*The knowledge-based expert system: A tutorial*". Computer 17 (9), (1984).

[JARD71] Jardine N. & Sibson R. "**Mathematical Taxonomy**". Wiley and Sons, New York, London, 1971.

[LAMP86] Lamport, L., "**LaTeX. User's guide & reference manual**", ed. Addison–Wesley Publishing Company, E.E.U.U., 1986.

[LÓPE81] López de Màntaras, R., "*Algorismes d'aprenentatge amb reconeixement de formes i aplicacions a la robòtica*". UPB, 1981. Tesis doctoral.

[LÓPE90] López de Mántaras, R., Crespo, J. J., "*El problema de la selección de atributos en el aprendizaje inductivo: nueva propuesta y estudio experimental*" En Memorias IBERAMIA 90, 2º Congreso Iberoamericano de Inteligencia Artificial. Ed. Limusa-Noriega. México, (1990) pp 259–271

[LLULL] Llull, R., "**Proverbis de Ramon**", Editora Nacional, Madrid, 1978.

[MART90] Martín Muñoz, M., "**LINNEO** *eina per a l'ajut en la construcció de bases de coneixement en dominis poc estructurats*". **UPC**, Departament d'**LSI**.Barcelona, març 1991. Tesi de llicenciatura.

[MICH82] Michalski, R. S., Davis J. H., Bisht, V. S., Sincler, J. B. "*PLANT/ds: An expert consulting system for the diagnosis of soybean diseases*" In Proceedings of European Conference AI. Orsay, France, (1982).

[PIER87] Piera i Carreté, N., "*Connectius de lògiques no estàndar com a operadors d'agregació en classificació*". UPC, Facultat d'Informàtica de Barcelona, 1987. Tesi doctoral.

[QUIN84] Quinlan, J. R. "*Learning efficient classification procedures and their application to chess end games*" In Michalski, R.S., Carbonell, J.G. and Mitchel, T.M. (Eds): *Machine Learning: An Artificial Intelligence Approach*. Tioga, PA, California, (1984) pp 463–482.

[ROUX85] Roux, M., "**Algorithmes de classification**", ed. Masson, Paris, 1985.

[SCHL86] Schlimmer, J. C., Fisher, D. "*A case study of incremental concept induction*". In Proc. of the fifth national conference on artificial intelligence. Ed. Morgan Kaufmann, (1986) pp 496–501.

[SHORT76] Shortliffe, E.H., "*MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection*". Ph.D.Tesis. Standforf University, 1976.

[TATA87] Tatar, D.G., "**A programmer's guide to Common Lisp**", ed. Digital Press, E.E.U.U., 1987.

[THOR53] Thorndike, R., L. "*Who belongs in a family?*", Psycometrika, 18, (1853) pp 267–273.

[VELD89] Van de Velde, W. "*Incremental Learning of optimal decision trees*". In Proc. of the fourth european working session on learning. Ed. Morgan Kaufmann, (1989) pp 211–225.

[WINS84] Winston, P.H., Paul Horn, B.K., "*LISP*", Second Edition ed. Addison–Wesley Publishing Company, E.E.U.U., 1984.

[WOLF70] Wolfe, J., H. "*Pattern clustering by multivariate mixture analysis.*", Multiv. Behav. Res., (1971), 5, pp 329–350.

Apèndix A

Parts de codi d'interés

En aquest apèndix es mostren les parts més interessants del codi de KLASS, que apareixen referenciats al llarg del texte.

En primer lloc, **crear_l_objectes** és una de les funcions que utilitzen paràmetres opcionals per a controlar les iteracions. En aquest cas, el paràmetre **índex** és intern a la funció i serveis, com el seu nom vol indicar, per a indexar la llista que es passa com a primer paràmetre.

```
(defun crear_l_objectes (llista &optional index_de_nivell (index 0))
        (if (null llista) nil
           (append (cdar llista)
                   (crear_l_objectes (cdr llista)
                                     (setf (get (cadar llista) 'index) '0)
                                     (+ index 1)
                   )
           )
        )
)
```

A continuació apareix la implementació que es fa dels procediments recursius pel cap. En concret, es presenten aquí les tres funcions imbricades que formen part del mòdul *Matriu de distàncies* i serveixen per a construir les dues matrius de subdistàncies qualitativa i quantitativa.


```
(defun submatrius (lobjs Ml Mt)
       (if (null (cdr lobjs))
          (progn
                 (setq Dquali (append Ml
                                      (fila_quali (car lobjs) nil nil nil)
                              )
                 )
                 (setq Dquant (append Mt Ti))
          )
          (submatrius (cdr lobjs)
                 (append Ml
                         (list (fila_quali (car lobjs) (cdr lobjs) nil nil))
                 )
                 (append Mt (list Ti))
          )
       )
)
```

La funció **fila_quali** construeix la sobrediagonal d'una fila de les matrius de distàncies parcials

```
(defun fila\_quali (obj reste Fl Ft)
  (if (null reste)
      (progn (setq Ti Ft)
             (setq Li Fl)
      )
      (fila\_quali obj (cdr reste)
                   (append Fl (list (elem\_quali obj (car reste))))
                   (append Ft (list tij))
      )
  )
)
```

La funció **elem_quali** calcula la distància en espai numèric i simbòlic dels objectes ob1 i ob2. D'altra banda, manté estructures paral·leles amb les distàncies ordenades.

```
(defun elem\_quali (ob1 ob2)
  (progn
    (setq dmax\_quali
          (ordena (sqrt (d\_quali\_2 l\_efectius
                                     l\_propietats
                                     (fila\_matriu ob1)
                                     (fila\_matriu ob2)
                                     (* n\_quali n\_quali)
                                     0
                                     0
                        )
                  )
                  dmax\_quali
          )
    )
    (setq dmax\_quant (ordena tij dmax\_quant))
    lij
 )
)
```

Un altre aspecte àmpliament comentat al llarg de tot el treball ha estat l'efecte lateral. Per a imposar un cert ordre sobre un conjunt d'efectes laterals, es defineix una funció auxiliar que retorni idènticament algun dels seus paràmetres, i es crida utilitzant els paràmetres per a imposar aquest ordre. En el cas que es presenta, la funció **modif_mes_proper** retornarà el primer dels seus paràmetres.

```
(defun distancies_agregades (element vei llista_objectes
                             metrica m_distancies
                            )
  (let ((propers_intermedi (remove (assoc element l_mes_proper)
                           l_mes_proper
                           )
       ))
```



```
      (retorna_dist m_distancies
              (setq
                 l_mes_proper
                 (modif_mes_proper
                      llista_objectes element vei
                      metrica 1 m_distancies
                      llista_objectes
                      (remove (assoc vei
                                     propers_intermedi
                              )
                       propers_intermedi
                      )
                 )
              )
      )
   )
)
```

Apèndix B

Fitxers dels jocs de proves

B.1 Fitxers Compositors

Fitxer d'objectes

```
((1 FURTWANGLER) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((4 GOEHR) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((6 HORENSTEIN) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((7 KARAJAN) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((5 KLEIBER) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((2 SCHERCHEN) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((8 TOSCANINI) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
((3 WAND) ((ADAGIO ALLEGRO FINALE SCHERZO) NIL NIL))
```

Fitxer de propietats

```
((3 ADAGIO) ((0.1405 0.1935) 1 8 C NIL))
((1 ALLEGRO) ((0.1315 0.1735) 1 8 C NIL))
((4 FINALE) ((0.2305 0.26) 1 8 C NIL))
((2 SCHERZO) ((0.1 0.13) 1 8 C NIL))
```


Fitxer que conté la matriu de dades

```
((0.1935 0.1735 0.245 0.115))
((0.172 0.1415 0.2355 0.1155))
((0.145 0.151 0.232 0.112))
((0.1555 0.145 0.2355 0.1))
((0.165 0.1545 0.233 0.1015))
((0.161 0.172 0.26 0.122))
((0.1405 0.1315 0.2305 0.13))
((0.1715 0.1625 0.235 0.112))
```

B.2 Fitxers de Paispet

Fitxer d'objectes

```
((4 ARGENTINA)
 ((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
((5 BRASIL)
 ((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
((1 CANADA)
((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
((6 CHILE)
((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
((3 MEXICO)
((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
((2 USA)
((GAR_FINAN GAR_POLITICA GER_EXEC SITUACION TENDENCIA) NIL NIL)
)
```

Fitxer de propietats

```
((5 GAR_FINAN) ((0 1) 1 6 C NIL))
((3 GAR_POLITICA) ((0 1) 1 6 C NIL))
((4 GER_EXEC) ((0 1) 1 6 C NIL))
((1 SITUACION) (NIL 1 6 Q (BAJA ESTABLE)))
((2 TENDENCIA) (NIL 1 6 Q (BAJA BUENA EXCELENTE)))
```

Fitxer que conté la matriu de dades

```
((0.324 0.335 0.457 BAJA BAJA))
((0.53 0.758 0.596 ESTABLE BUENA))
((0.79 0.787 0.823 ESTABLE EXCELENTE))
((0.125 0.051 0.191 ESTABLE BAJA))
((0.639 0.609 0.639 BAJA BUENA))
((0.772 0.822 0.836 ESTABLE EXCELENTE))
```

Apèndix C

Esponges: Detall dels arbres jeràrquics

A continuació es presenten els subarbres resultants d'un tall horitzontal de la figura 4.13 per $\alpha = 3.5$ a fi que el lector pugui veure més detingudament els objectes de cada subarbre.

De fet, el tercer dels subarbres, (figura C.2) és prou gran com per a practicar-li un segon tall a nivell $\alpha = 1.5$, representant els subarbres resultants d'aquest tall en les figures C.3 à C.6.

A partir de la figura C.9 apareixen els subarbres corresponents al tall horitzontal de nivell 3.7 de la figura 4.14.

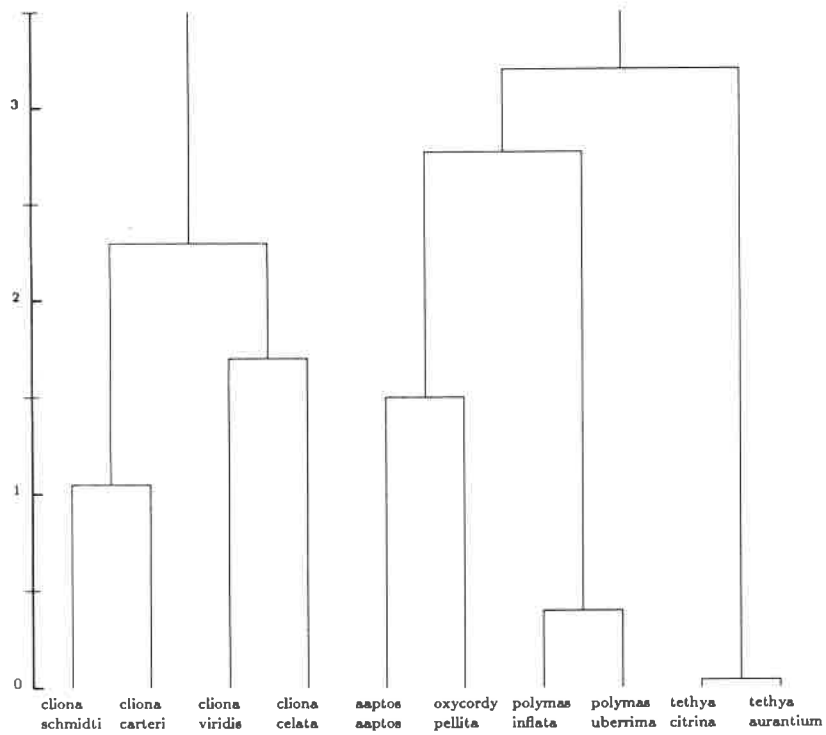


Figura C.1: Esponges. Criteri del centroide. Subarbres 1 i 2.

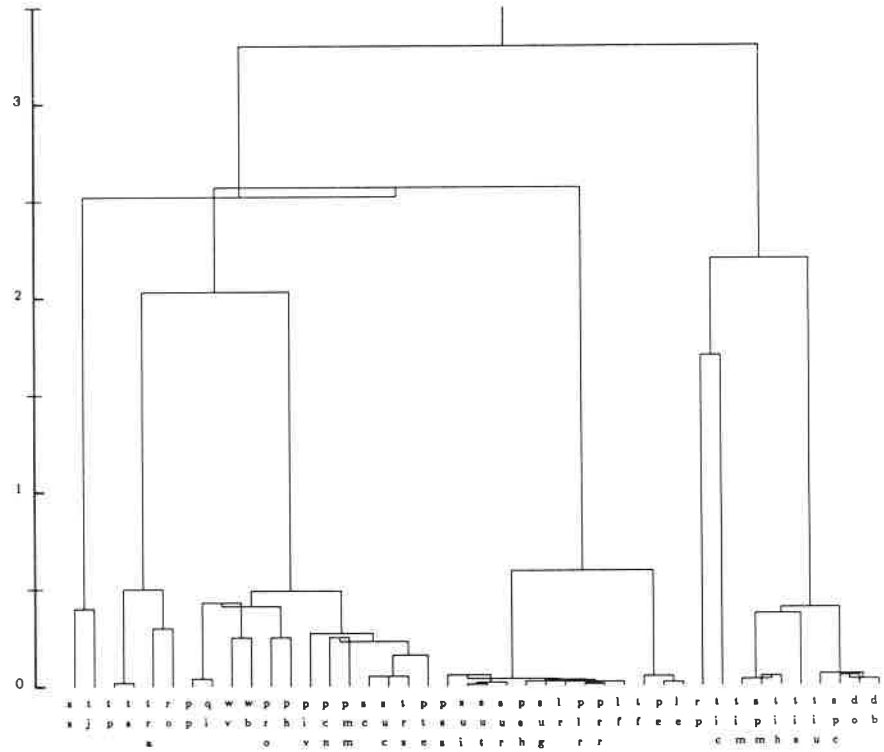


Figura C.2: Esponges. Criteri del centroide. Subarbre 3.

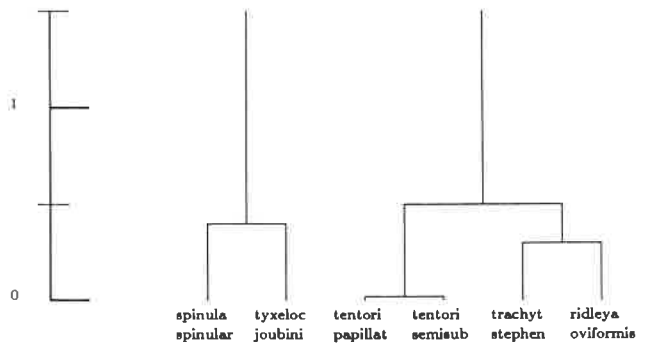


Figura C.3: Esponges. Criteri del centroide. Subarbres 3a i 3b.

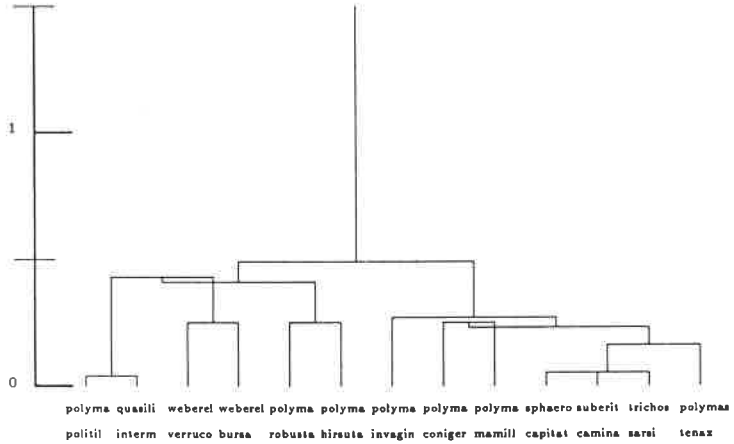


Figura C.4: Esponges. Criteri del centroide.Subarbre 3c.

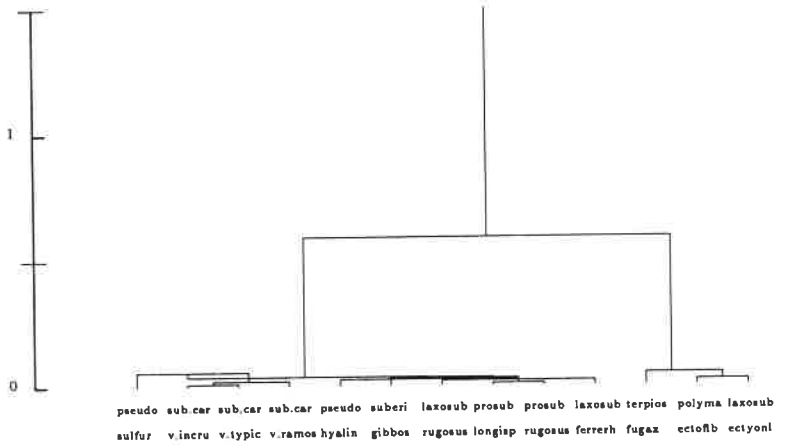


Figura C.5: Esponges. Criteri del centroide. Subarbre 3d.

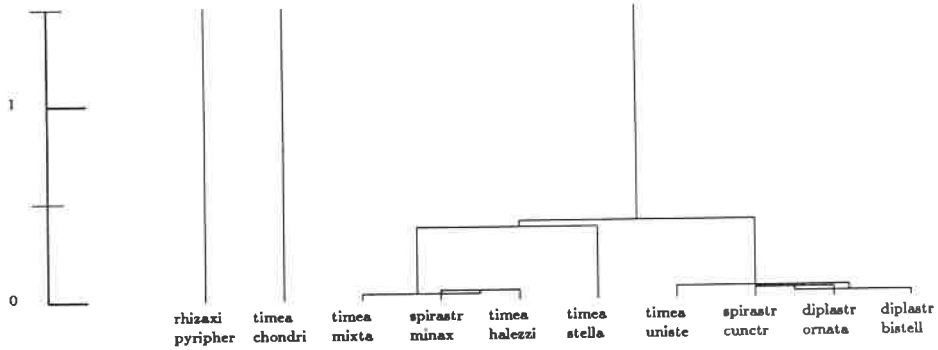


Figura C.6: Esponges. Criteri del centroide. Subarbre 3e.

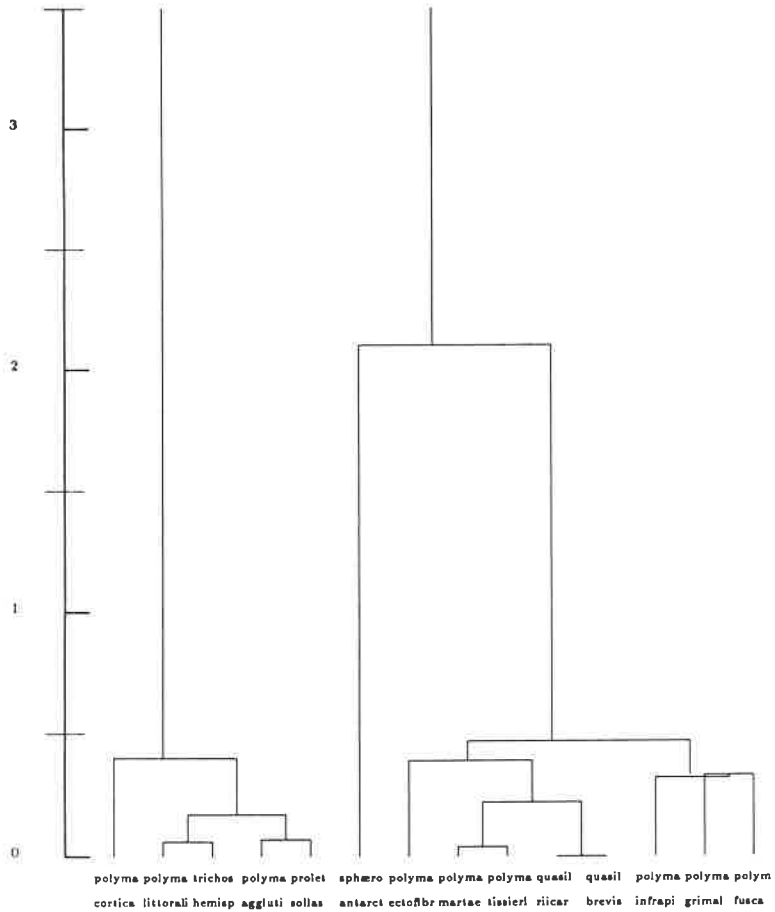


Figura C.7: Esponges. Criteri del centroide. Subarbre 4 i 5.

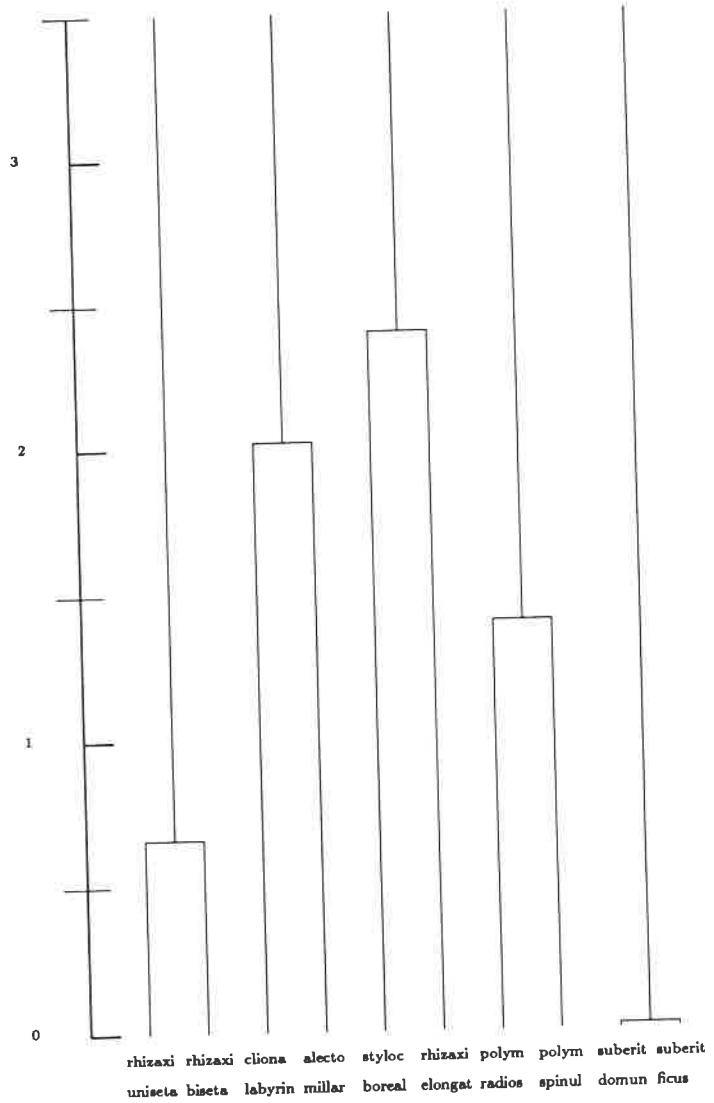


Figura C.8: Esponges. Criteri del centroide. Subarbre 6 à 10.

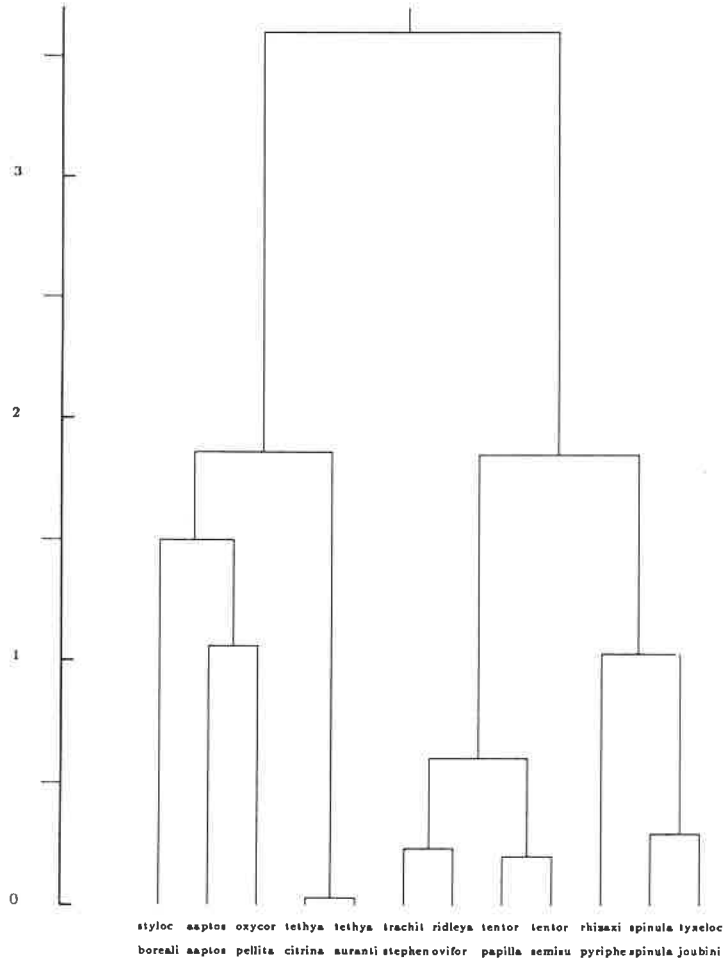


Figura C.9: Esponges. Criteri de Ward. Subarbre 1.

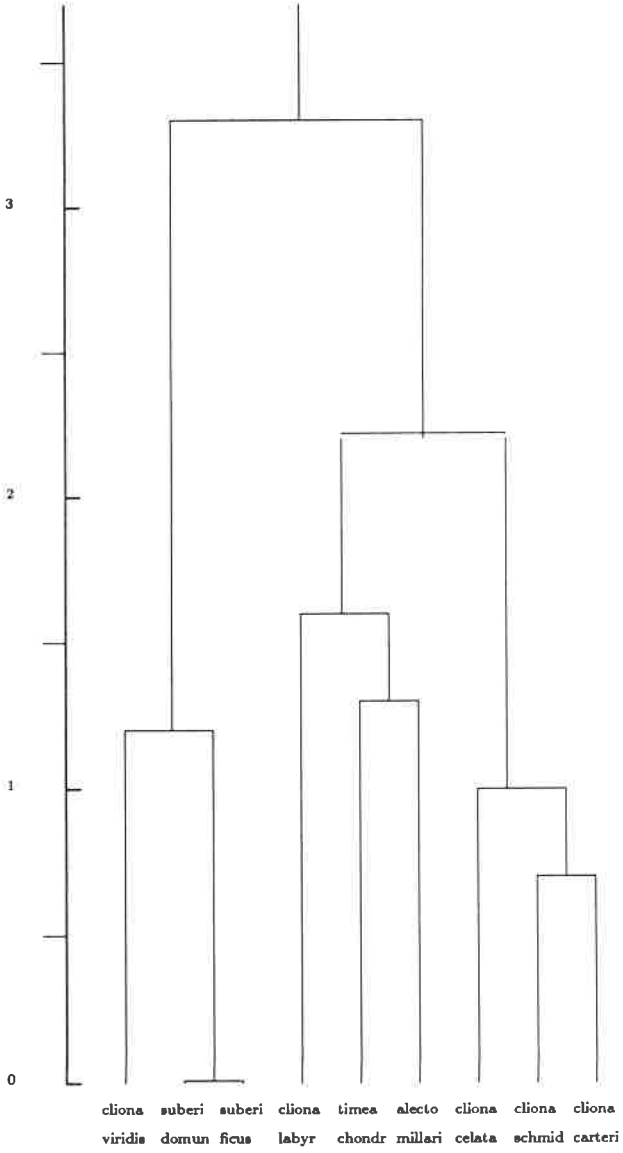


Figura C.10: Esponges. Criteri de Ward. Subarbre 2.

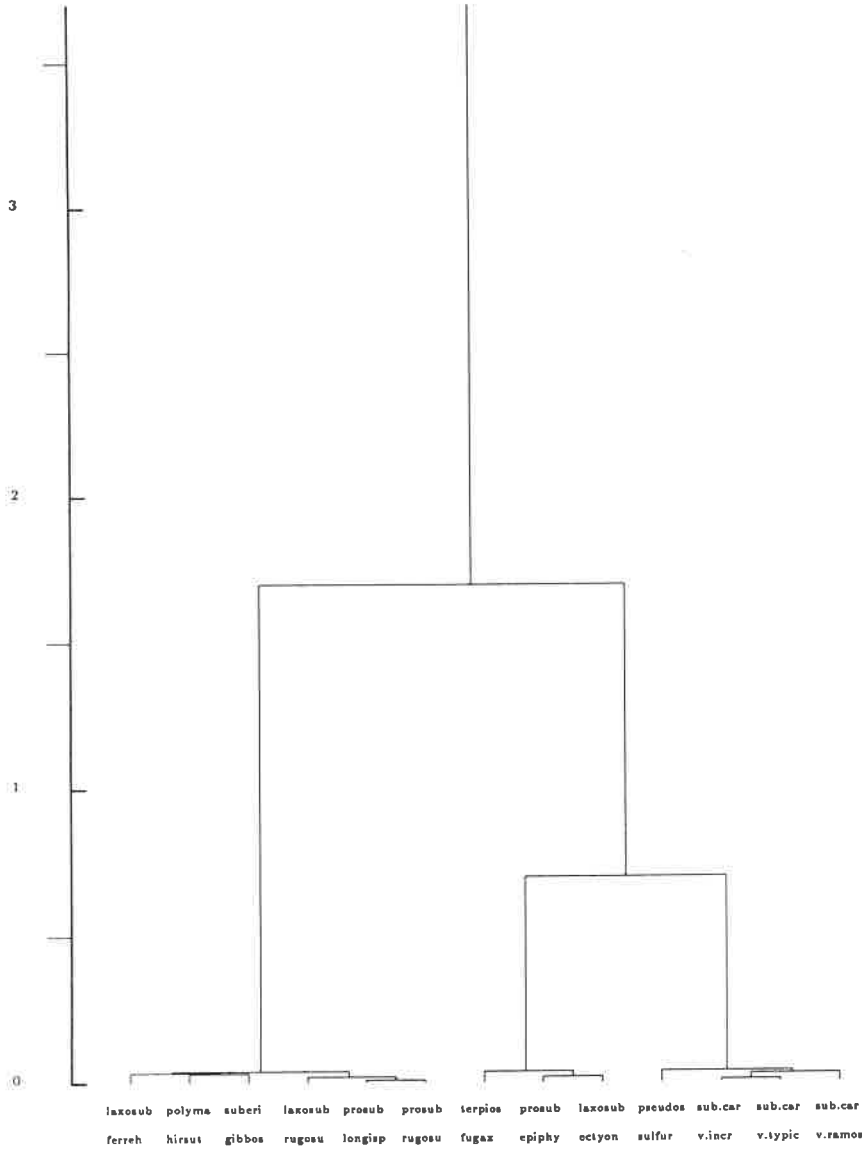


Figura C.11: Esponges. Criteri de Ward. Subarbre 3.

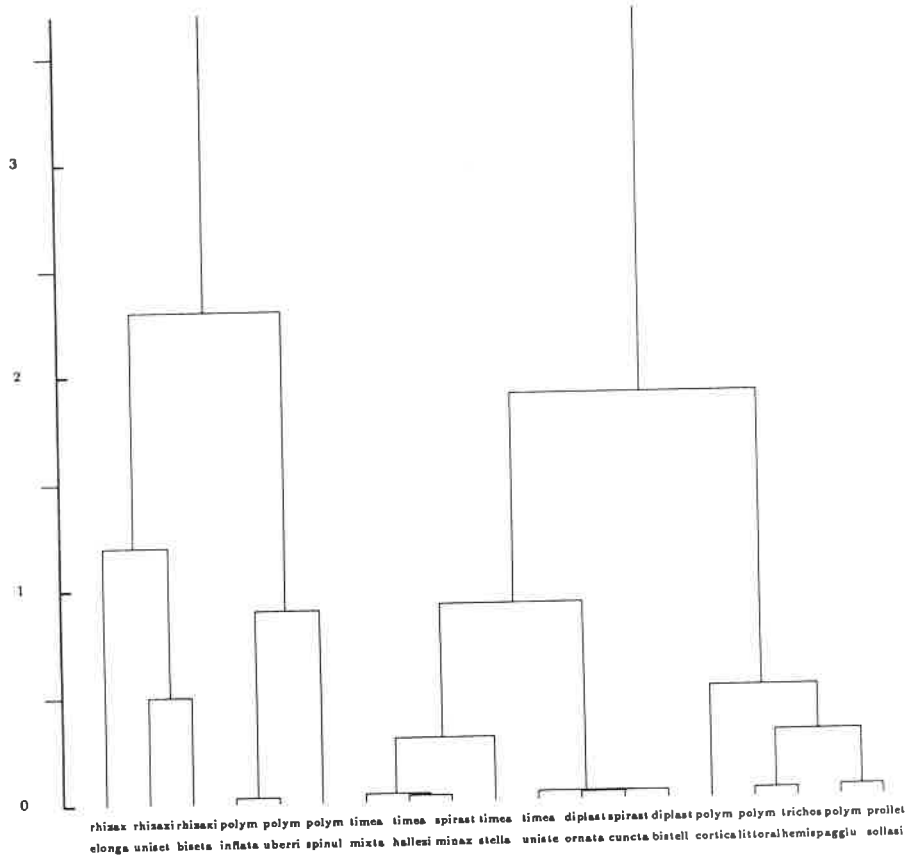


Figura C.12: Esponges. Criteri de Ward. Subarbres 4 i 5.

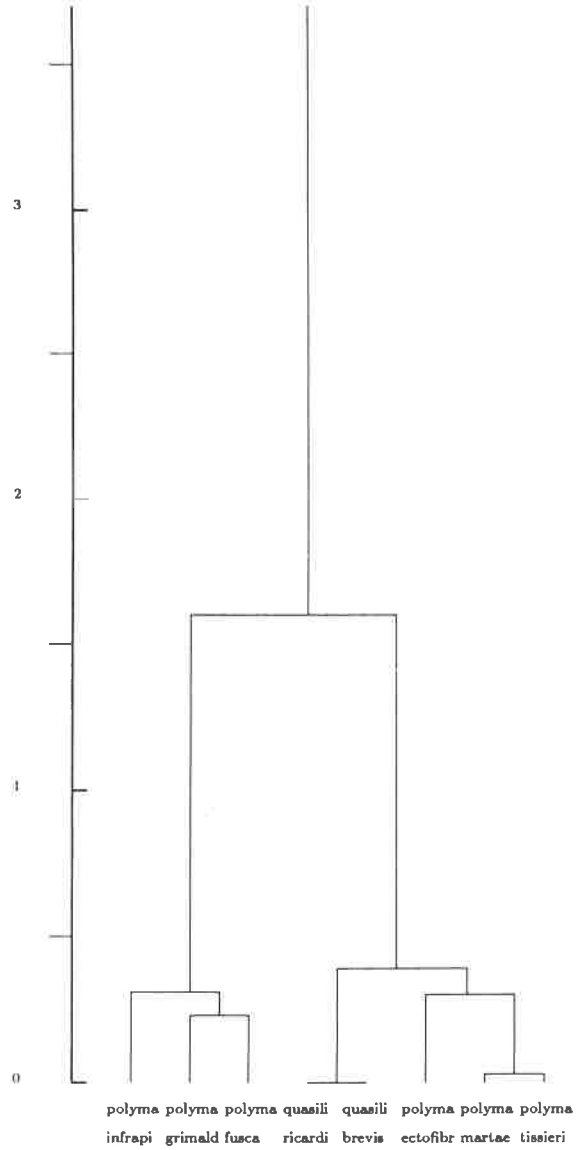


Figura C.13: Esponges. Criteri de Ward. Subarbres 6.

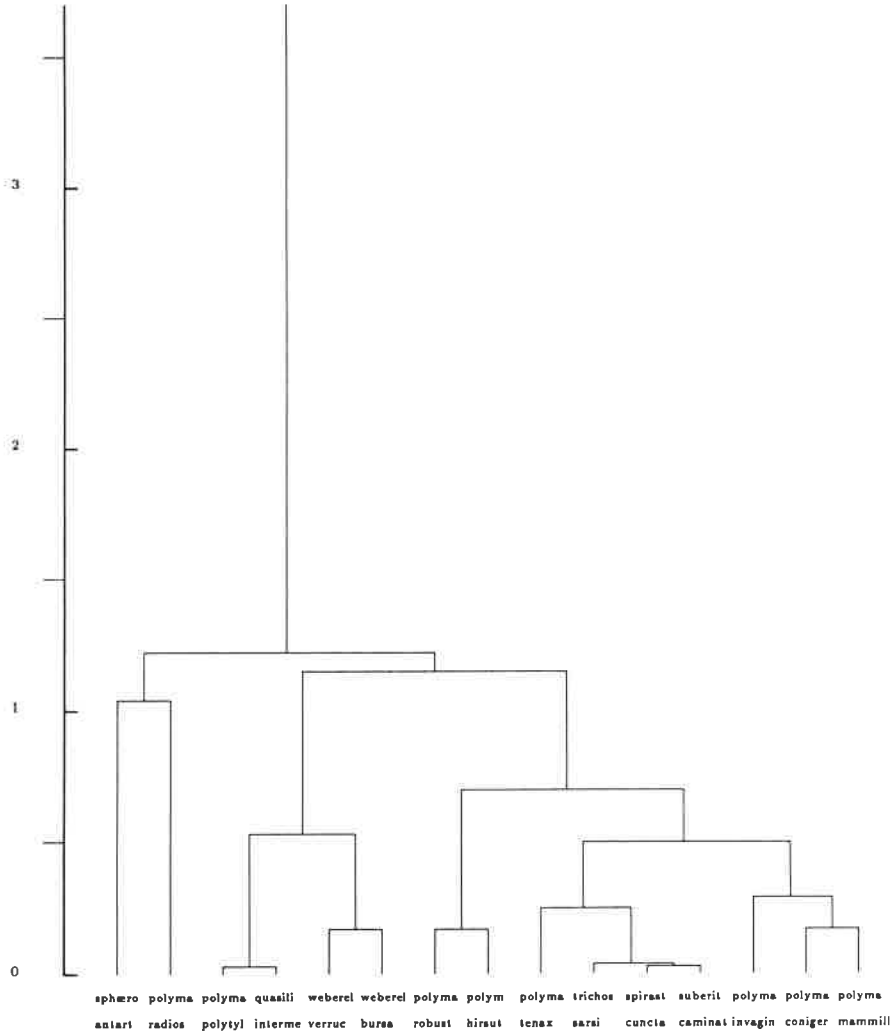


Figura C.14: Esponges. Criteri de Ward. Subarbre 7.