DEPARTAMENTO DE

ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

ANALISE DE AGRUPAMENTOS COM CONCEITOS IMPRECISOS COM VISTA À CLASSIFICAÇÃO DE DIAGRAMAS DE CARGA

FACULDADE DE ENGENHARIA

UNIVERSIDADE DO PORTO

Rua dos Bragas, 4099 Porto Codex – PORTUGAL

Faculdade de Engenharia da Universidade do Porto

ANALISE DE AGRUPAMENTOS COM CONCEITOS IMPRECISOS COM VISTA À CLASSIFICAÇÃO DE DIAGRAMAS DE CARGA

ANA BELA DE SOUSA OLIVEIRA COUTO MAGALHÃES

OUTUBRO de 1991

Dissertação submetida para satisfação parcial dos requesitos do programa de Mestrado em

Engenharia Electrotécnica e de Computadores

(Sistemas de Energia)

Dissertação realizada sob a supervisão do

Prof. Doutor Vladimiro Henrique Barrosa Pinto de Miranda

Professor Associado do

Departamento de Engenharia Electrotécnica e de Computadores

da

Faculdade de Engenharia da Universidade do Porto

Aos meus filhos
Joana, Filipa e Pedro

AGRADECIMENTOS

Ao Professor Vladimiro Miranda, meu orientador científico, pelo estímulo constante durante a elaboração deste trabalho.

A todos os meus colegas, em particular à Engª Isabel Abreu, ao Engº José Rui Ferreira e ao Engº João Machado, que sempre me incentivaram e de alguma forma contribuiram para a realização deste trabalho.

Ao Instituto Superior de Engenharia do Porto, que me facultou as condições necessárias à frequência do Mestrado.

Ao INESC pelas facilidades concedidas durante a elaboração da tese.

INDICE

1. Introdução

O conceito de imprecisão foi introduzido por Zadeh em 1965 [1] e encontra-se ligado à definição de pertença de um elemento a um dado conjunto.

Segundo a teoria matemática clássica, um elemento x e X ou é elemento de A $\subset$ X ou não. Pode, então, definir-se uma função de pertença u(x) que toma o valor 1 no primeiro caso e 0 no segundo.

A introdução daquele novo conceito permitiu alargar a noção de pertença de tal modo que a função u(x) deixa de estar limitada ao conjunto {0,1} e passa a tomar valores no intervalo [0,1].

Registe-se a diferença existente entre as definições de probabilidade e possibilidade. Ambas lidam com graus de incerteza e são definidas no intervalo [0,1]. No entanto, a qualidade da imprecisão é diferente nos dois casos. Na teoria de propabilidades a incerteza liga-se ao carácter aleatório dos acontecimentos enquanto que em teoria de possibilidades a incerteza é uma característica do modo vago como é definida a ocorrência.

O tema base desta dissertação é actual e tem merecido da parte dos investigadores crescente atenção. A análise de agrupamentos é uma das áreas óbvias de aplicação da teoria dos conjuntos imprecisos e abrange campos tão distintos como a medicina e a linguística.

Não se conhecendo em português nenhuma publicação que aborde o tema em questão, considerou-se de interesse apresentar uma compilação quer de conceitos básicos da teoria dos conjuntos imprecisos quer de aplicações efectuadas com base nestes mesmos conceitos.

Era intenção inicial vir a aplicar estas técnicas na análise e classificação de diagramas de carga de distribuição de energia eléctrica, apresentando exemplos trabalhados em que se pudessem apreciar devidamente as virtudes comparativas de partições imprecisas com outras obtidas por métodos distintos. Afirmar que um dado diagrama é do tipo industrial envolve, já por si, um certo grau de incerteza, o que leva a crer que bons resultados podem ser obtidos através de uma modelização imprecisa deste processo.

Como não foi possível obter uma amostra significativa de dados relativos a diagramas de consumo na distribuição em Portugal, o objectivo inicial teve que ser inflectido para a demonstração da técnica a utilizar. Assim, para ilustração do processo de separação imprecisa de diagramas lançou-se mão de um conjunto de curvas de afluências hidrológicas à barragem da Aguieira.

Não se pretende levar a extremos a comparação, mas julga-se que de qualquer modo será útil a observação dos resultados e da forma como foram obtidos, o que poderá encontrar-se no capítulo 6. Nos restantes capítulos a dissertação encontra-se organizada como segue:

No presente capítulo, capítulo 1, faz-se o enquadramento do assunto a desenvolver e descreve-se a estrutura desta dissertação.

No capítulo 2 é feita uma descrição dos processos clássicos de obtenção de agrupamentos rígidos e referem-se os resultados pouco satisfatórios com eles obtidos em algumas situações em que aspectos de natureza qualitativa prevalecem.

O capítulo 3 é dedicado a conceitos básicos da teoria de conjuntos e partições imprecisas que são a ferramenta matemática utilizada para desenvolvimentos de métodos imprecisos para análise de agrupamentos.

Alguns desses métodos são descritos no capítulo 4, onde se apontam as vantagens destes sobre os métodos clássicos de agrupamento rígido na interpretação de características de processos reais.

No capítulo 5 é abordado o tema de validação das partições incluindo a descrição de alguns dos indicadores conhecidos. Faz-se neste capítulo a apresentação original de um novo índice para avaliar a qualidade da partição obtida, reconhecidas que foram algumas insuficiências de carácter conceptual de outros índices descritos na literatura e recolhidos nesta dissertação.

O capítulo 6 contem um conjunto de exemplos que foram analisados com alguns dos métodos descritos no capítulo 4, fazendo-se uma análise dos resultados obtidos em termos de qualidade dos agrupamentos definidos.

No capítulo 7 é feita uma apreciação ao trabalho realizado e apresentam-se algumas conclusões.

Em anexo encontra-se uma compilação dos programas utilizados para obtenção dos resultados que se apresentam.

O título desta dissertação, podendo à partida julgar-se excessivamente ambicioso face à matéria reunida, justifica-se pela preocupação constante de trabalhar uma técnica que pudesse ter aplicação em Análise de Sistemas Eléctricos, embora, obviamente, a tal não esteja limitada. Espera-se que esta dissertação possa cumprir uma função de carácter didático a outros investigadores interessados no aprofundamento dos temas abordados.

2. A Análise Clássica de Agrupamentos e Suas Insuficiências

2.1. Introdução

A análise de agrupamentos é um método de classificação de elementos em grupos, através de uma medida de associação s, de forma que elementos de um mesmo grupo sejam semelhantes e dois grupos não sejam semelhantes entre si[2].

Os objectivos da análise de agrupamentos são vários, podendo ser resumidos em:

i) Elaboração de um bom processo de classificação.

ii) Identificação de estruturas no conjunto de dados.

iii) Simplificação do sistema em análise, conduzindo a uma melhor compreensão desse sistema.

Este método é aplicado nas mais diversas áreas tais como: medicina, biologia, engenharia, arqueologia e outras.

Quando se pretende desenvolver uma técnica de agrupamento torna-se necessário escolher:

a) O modelo apropriado ao processo em estudo

b) O critério de agrupamento

c) O número de grupos em que se presume dividir o conjunto

A maior parte dos conceitos matemáticos clássicos, no que respeita a teoria de conjuntos, são rígidos, determinísticos e precisos; isto é, dado um conjunto **X** e uma medida de semelhança s, o resultado de uma análise de agrupamentos será um número de grupos $\{A_1, A_2, \ldots\ldots, A_c\}$, que formam o que se chama uma **partição** de **X**. Esses grupos verificam as seguintes condições:

$$A_i \subset X \qquad i=1,\ldots\ldots,c$$

$$A_1 \cup A_2 \cup \ldots\ldots \cup A_c = X$$

$$A_i \cap A_j = \emptyset \qquad i \neq j \text{ com } i,j = 1,\ldots,c$$

A este tipo de partições, em que os agrupamentos formados são exaustivos e mutuamente exclusivos, dá-se o nome de **partições rígidas.**

2.2. Métodos de Agrupamento

Os métodos para análise de agrupamentos podem ser divididos em duas categorias, de acordo com o critério de agrupamento utilizado :

métodos hierárquicos

métodos não hierárquicos

2.2.1. Métodos hierárquicos

Estes métodos geram uma hierarquia de partições $\{G\}$, isto é, uma série de partições em que para qualquer par (G^i, G^j) se verifica o seguinte:

Sendo

$$G^i = \left\{ A^i_1, A^i_2, \ldots\ldots, A^i_M \right\} \quad G^j = \left\{ A^j_1, A^j_2, \ldots\ldots, A^j_N \right\}$$

para qualquer $A^i_k \in G^i$ existe um $A^j_r \in G^j$ tal que $A^i_k \subseteq A^j_r$.

Tal hierarquia pode ser representada por um diagrama em árvore, chamado **dendograma**, como o que se representa na figura 2.1.

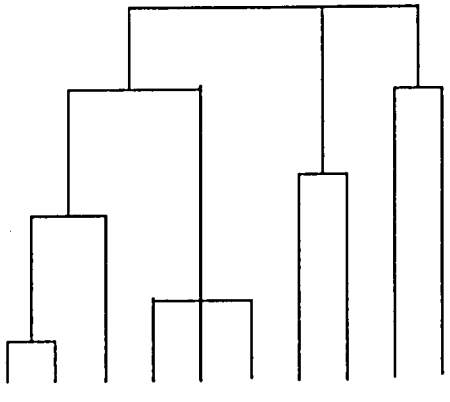


FIGURA 2.1.

Para obtenção desta série de grupos podem ser usados processos de **aglutinação** ou processos de **separação.**

Um algoritmo de separação começa com um único grupo contendo todos os elementos de **X** e divide-o em vários grupos e cada um destes em outros até se chegar a agrupamentos constituídos por um único elemento cada.

Num algoritmo de aglutinação o processo é inverso. Inicialmente cada elemento do conjunto **X** forma um grupo. Reune-se o par de grupos mais semelhantes, formando um novo grupo e continua-se o processo até se atingir a partição trivial $G^k = X$.

Um dos algoritmos mais importantes na análise hierárquica de agrupamentos é o "**single linkage**", que se exemplifica de seguida.

Exemplo 2.1

Considere-se o conjunto **X** definido pelos seis pontos representados na figura 2.2. Os pontos encontram-se ligados por ramos e os números nesses ramos representam o valor da distância entre os extremos, que pode ser convertida em medida de semelhança — quanto mais próximos estiverem os pontos mais semelhantes são considerados. Quando o ramo de ligação não existe, considera-se infinito o valor dessa distância.

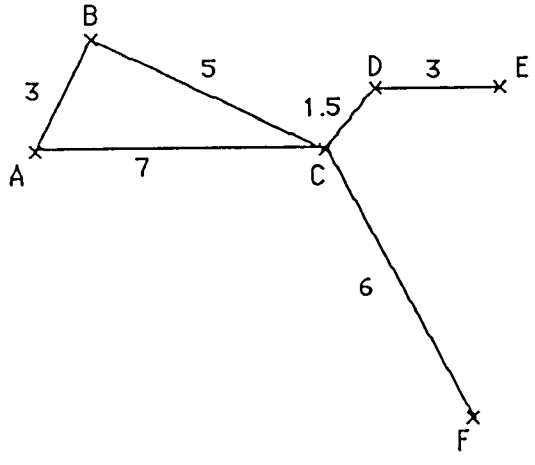


FIGURA 2.2

Inicialmente cada ponto constitui um grupo; logo, a primeira partição de **X** será :

$$G^1 = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}\}$$

Associam-se, em seguida, os dois pontos mais semelhantes, isto é, mais próximos. Neste caso são os pontos C e D. Logo,

$$G^2 = \{\{A\}, \{B\}, \{C, D\}, \{E\}, \{F\}\}$$

Para progredir no processo de aglutinação, torna-se necessário definir distância entre grupos, que pode ser tomada como a menor entre os elementos desses grupos. Por exemplo,

$$d(\{A\}, \{C, D\}) = \min\{d(C, A), d(D, A)\} = 5$$

Calculando as distâncias entre todos os elementos da partição G^2, os candidatos à reunião são dois: os grupos {A } e {B } e os grupos {C , D } e {E }.

Vem, então

$$G^3 = \{\{A, B\}, \{C, D, E\}, \{F\}\}$$

Prosseguindo, chegar-se-ia a uma partição final de um só agrupamento contendo os seis pontos.

Representando o processo através de um dendograma virá:

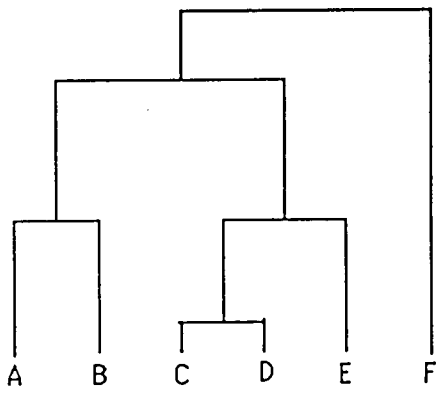


FIGURA 2.3

Um dos problemas de mais difícil resolução, na análise dos resultados obtidos por este algoritmo, é a escolha, de entre as muitas partições obtidas, da mais significativa.

2.2.2. Métodos Não Hierárquicos

Grande parte dos métodos não hierárquicos baseia-se na optimização de uma função exprimindo um dado critério. Várias funções podem ser definidas para a obtenção de agrupamentos; uma das formas que essas funções apresentam é

$$J(G) = \sum_{i=1}^{c} \sum_{x_k \in A_i} ||x_k - v_i||^2$$

onde

v_i é o centróide do agrupamento i

c é o número de agrupamentos em que foi dividido o conjunto inicial

$X = \{x_1, x_2, \ldots\ldots, x_n\}$

$||\ .\ ||$ é a norma Euclideana

Este funcional J(G) deve ser minimizado no domínio de todas as partições admissíveis e o resultado pode ser representado por uma matriz (c×n), a que se chama **matriz de**

partição. Cada linha desta matriz é constituída por 0 e 1. Os 1 da linha i significam que os elementos de X, correspondentes a essas colunas, pertencem ao agrupamento i; todos os outros não pertencem. Por outro lado, como cada elemento de X pertence a um e um só grupo, já que se trata de partição rígida, cada coluna da matriz tem apenas um elemento igual a 1.

A minimização de J(G) é difícil, uma vez que o número de partições possíveis aumenta rapidamente à medida que o número de elementos do conjunto em análise aumenta. Daí que muitos dos algoritmos não hierárquicos sejam processos iterativos, podendo assim trabalhar com um grande número de objectos.

Um dos métodos mais conhecidos neste tipo de análise de agrupamentos é o clássico **Hard c-Means** de Duda e Hart, que se descreve com mais detalhe na secção 4.3.1 do capítulo 4.

Refiram-se, ainda dentro dos métodos não hierárquicos, os métodos baseados em teoria de grafos que definem os agrupamentos a partir de árvores mínimas e eliminando, nessas mesmas árvores, alguns ramos originando sub-grafos, aos quais correspondem os vários grupos identificados no conjunto X.

2.3. Insuficiências dos Métodos Clássicos

Como já foi referido, os conceitos matemáticos utilizados na elaboração de qualquer um destes métodos de análise de agrupamentos são rígidos, determinísticos e precisos. No entanto, muitas das situações da vida real envolvem conceitos de alguma forma vagos, não podendo ser traduzidas de maneira conveniente por partições rígidas. Mesmo a adopção de modelos probabilísticos ou estocásticos não resolve de todo o problema, uma vez que a incerteza nestes casos se deve a uma definição imprecisa do processo e não a uma natureza aleatória do mesmo.

Suponha-se o conjunto X da figura 2.4.

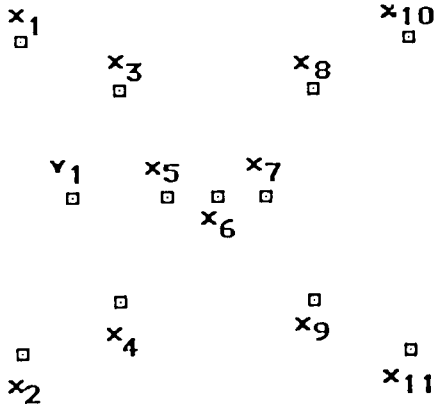


FIGURA 2.4

Qualquer que seja o método clássico de agrupamento que se utilize, obtêm-se sempre partições tais que, para cada uma destas, cada elemento de X pertence a um só grupo, entre todos os que a constituem.

Ora, o elemento x_6 é simétrico relativamente aos sub-conjuntos

$$A_1 = \{x_1, x_2, x_3, x_4, x_5\} \quad e \quad A_2 = \{x_7, x_8, x_9, x_{10}, x_{11}\}$$

A qual dos grupos fazer pertencer este ponto? Como traduzir essa mesma simetria?

Por outro lado, quer o ponto x_1 quer o ponto x_2 não são, em termos da medida de associação s, tão semelhantes, relativamente ao ponto mais "central" v_1, como x_3 e x_4. Como representar esta diferença?

Numa partição rígida, cada elemento do conjunto **X** pertence a um e um só grupo e todos os elementos de um mesmo grupo apresentam o mesmo valor de pertença. Portanto, toda esta informação disponível a respeito das características do conjunto dado se perde quando se estabelecem agrupamentos de uma forma clássica.

3. Conceitos Básicos de Teoria dos Conjuntos Imprecisos e das Partições Imprecisas.

3.1. Introdução

É indiscutível a vantagem do uso de modelos matemáticos para caracterização e estudo de fenómenos da vida real. No entanto, a maior parte destas situações são não determinísticas, não podendo ser descritas com precisão.

Zadeh referia numa publicação sua em 1973: "À medida que a complexidade de um sistema aumenta a nossa capacidade para produzir afirmações precisas e significativas a respeito do seu comportamento diminui até que um patamar é atingido, para além do qual precisão e significado tornam-se características mutuamente exclusivas." [3].

A imprecisão referida pode ser de diferentes naturezas.

No caso de um processo aleatório os resultados são deduzidos com base em probabilidades de ocorrência impostas à partida, havendo incerteza quanto àquilo que na realidade se passará.

Ou então, a própria descrição do processo em causa é vaga. Por exemplo, como definir o conjunto das pessoas com altura aproximada de 1,60 metros? O que se entende por aproximada? Como representar este conceito matematicamente? A imprecisão, desta vez, está contida na própria caracterização do conjunto, **conjunto impreciso** (fuzzy set).

3.2. Alguns Conceitos Básicos [4]

Sejam dados um conjunto **X** e um sub-conjunto $A \subset X$.

Segundo a teoria matemática clássica, qualquer elemento $x \in X$ ou é elemento de **A** ou não. Este sub-conjunto **A**, dito **rígido** (hard set), pode ser definido de várias formas: em extensão, por enumeração de todos os seus elementos; em compreensão, enunciando a propriedade comum a todos os seus elementos; ou, ainda, usando uma função chamada **função característica** ou **função de pertença**

$$u_A : X \rightarrow \{0,1\}$$

$$u_A(x) = \begin{cases} 0 \text{ se } x \in A \\ 1 \text{ se } x \notin A \end{cases} \qquad \forall x \in X$$

Considerando, agora, um sub-conjunto impreciso **B** de **X**, poder-se-á imaginar uma nova função de pertença que em vez de caracterizar os elementos como pertencentes ou não pertencentes a **B**, os defina como mais ou menos pertencentes a este sub-conjunto. Isto é, a função passa a tomar valores no intervalo [0,1], ao contrário do que acontecia para o sub-conjunto **A**, em que a função tinha como valores 0 ou 1. Então, $u_B : X \rightarrow [0,1]$ em que $u_B(x)$ se identifica com o grau de pertença de x a **B**.

Exemplo 3.1

Seja um conjunto **X** de n pessoas e tomem-se sobre este conjunto dois sub-conjuntos, um rígido **A** e outro impreciso **B**, definidos como se segue.

A é o sub-conjunto das pessoas com uma altura de 1,60 metros, admitindo uma tolerância de 0.5%

$$A = \{x \in X \mid h(x) = 1.60 \pm 0.005\}.$$

B é o sub-conjunto impreciso definido como sendo o das pessoas com uma altura aproximada de 1.60 metros

$$B = \{x \in X \mid h(x) \text{ é aproximadamente } 1.60\}.$$

Definindo a função de pertença para estes dois conjuntos vem:

$$u_A(x) = \begin{cases} 1 \text{ se } x \in A \\ 0 \text{ se } x \notin A \end{cases} \qquad \forall x \in X$$

$$u_B(x) = \begin{cases} 1 \quad \text{se } 1.5995 \leq h(x) \leq 1.6005 \\ 0.95 \text{ se } 1.5990 \leq h(x) < 1.5990 \text{ ou } 1.6005 < h(x) \leq 1.6010 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots\ldots \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots\ldots \\ 0.05 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots\ldots \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots\ldots \end{cases}$$

Os valores tomados pela função $u_B(x)$ pretendem representar o grau com que h(x) se aproxima de 1.60 metros.

A definição do conjunto B permitiu representar, em termos matemáticos, o conceito de aproximadamente, sendo o modelo impreciso aquele que mais informação torna disponível acerca do conjunto em estudo.

Parece oportuno estabelecer a diferença conceptual entre probabilidade e imprecisão. Repare-se que ambos os conceitos lidam com graus de incerteza e que ambos usam o intervalo [0,1] para a definição das suas medidas. No entanto, quando, por exemplo, se define que

$$Pr\,(1.5990 \leq h(x) < 1.5995) = 0.95$$

após a observação de $h(x_1) = 1.5989$ esta mesma probabilidade vem

$$Pr\left(1.5990 \leq h(x) < 1.5995 \mid h(x_1)1.5989\right) = 0$$

Pelo contrário, o valor de $u_B(x_1)$, grau com que $h(x_1)$ se aproxima de1.60, mantem-se igual a 0.9 após a ocorrência de $h(x_1)$=1.5989.

3.3. Álgebra dos Conjuntos Imprecisos

Seja P(X) o conjunto dos sub-conjuntos rígidos de **X** e sejam **A** e **B** elementos de P(X). Entre **A** e **B** pode ser formulado um conjunto de operações:

$$A \subset B \Leftrightarrow x \in A \Rightarrow x \in B$$

$$A = B \Leftrightarrow A \subset B \text{ e } B \subset A$$

$$\tilde{A} = \{ x \in X : x \notin A \} = X\text{-}A$$

$$A \cap B = \{x \in X : x \in A \text{ e } x \in B\}$$

$$A \cup B = \{x \in X : x \in A \text{ ou } x \in B \text{ ou } x \in \text{ambos}\}$$

Como as funções de pertença são as componentes fundamentais dos conjuntos imprecisos, parece conveniente fazer-se uma caracterização de P(X) em termos destas mesmas funções.

Seja, então, P(F) o conjunto das funções de pertença dos sub-conjuntos rígidos definidos em **X**. Assim,

$$u_A \leq u_B \Leftrightarrow u_A(x) \leq u_B(x) \qquad \forall x \in X$$

$$u_A = u_B \Leftrightarrow u_A(x) = u_B(x) \qquad \forall x \in X$$

$$\tilde{u}_A(x) = u_{\tilde{A}}(x) = 1\text{-}u_A(x)$$

$$u_{A \cap B}(x) = \min \left\{u_A(x), u_B(x)\right\}$$

$$u_{A \cup B}(x) = \text{máx} \left\{u_A(x), u_B(x)\right\}$$

O conjunto vazio é definido por uma função de pertença constantemente nula e ao conjunto inteiro corresponde a função constantemente unitária; isto é,

$$0(x) = 0 \qquad \forall x \in X$$

$$1(x) = 1 \qquad \forall x \in X$$

Pode afirmar-se, então, que P(X) e P(F) são estruturas algébricas idênticas, P(X)≈P(F). Por outras palavras, o comportamento dos sub-conjuntos rígidos de **X** é equivalente ao comportamento das suas funções de pertença.

Propriedades semelhantes a estas podem ser definidas para os conjuntos imprecisos tomando-se, a partir de agora, $P_f(F)$ como o conjunto de todos os sub-conjuntos imprecisos de **X**.

Assim sendo, pode afirmar-se que a estrutura algébrica (**X**, **P(F)**) está contida nesta outra estrutura (**X**, $P_f(F)$), uma vez que se pode considerar os conjuntos rígidos como um caso particular dos conjuntos imprecisos.

3.4. Partições Imprecisas

Qualquer sub-conjunto rígido $A \subset X$ goza das seguintes propriedades:

$$A \cup \tilde{A} = X$$
$$A \cap \tilde{A} = \emptyset$$
$$\emptyset \subset A \subset X$$
$$\emptyset \subset \tilde{A} \subset X$$

o que, em termos de funções de pertença pode ser escrito da seguinte forma:

$$u_A \vee u_{\tilde{A}} = 1 \qquad (3.1.a)$$
$$u_A \wedge u_{\tilde{A}} = 0 \qquad (3.1.b)$$
$$0 < u_A < 1 \qquad (3.1.c)$$

(os símbolos $\vee$ e $\wedge$ indicam máximo e mínimo, respectivamente).

A e $\tilde{A}$ são, então, sub-conjuntos não vazios, mutuamente exclusivos e cuja união reproduz o conjunto **X**. O par (**A**, $\tilde{A}$), nestas condições, forma aquilo que se costuma chamar uma **partição-2 rígida** de **X**.

As propriedades agora enunciadas não se verificam, no entanto, para o caso dos sub-conjuntos ou partições imprecisas de **X**. Por exemplo, seja **A** um sub-conjunto impreciso de **X** tal que

$$u_A(x) = 0.5$$

Então

$$u_{\tilde{A}}(x) = 1 - u_A(x) = 1 - 0.5 = 0.5 \qquad \forall\, x \in X.$$

Logo

$$u_A \wedge u_{\tilde{A}} = 0.5 \text{ e não } 0$$

$$u_A \vee u_{\tilde{A}} = 0.5 \text{ e não } 1$$

Qual a imposição a fazer a uma partição para que possa ser considerada uma partição-2 imprecisa de X? Seja **X** um dado conjunto e $P_f(F)$ o conjunto de todos os sub-conjuntos imprecisos de **X**. O par (u_A, $u_{\tilde{A}}$) é uma **partição-2 imprecisa** de **X** se

$$u_A + u_{\tilde{A}} = 1$$
$$0 < u_A < 1$$

Note-se que estas equações reduzem-se às equações (3.1) se se tratar de uma partição rígida.

Generalizando, diz-se que $\{A_i\}$ para $1 \leq i \leq c$ é uma **partição-c rígida** de um conjunto **X** se se verificarem as seguintes condições :

$$\bigcup_{i=1}^{c} A_i = X \qquad\qquad \bigvee_{i=1}^{c} u_i = \mathbf{1}$$

$$A_i \cap A_j = \emptyset \quad 1 \leq i \neq j \leq c \quad \Leftrightarrow \quad u_i \wedge u_j = \mathbf{0} \quad 1 \leq i \neq j \leq c$$

$$\emptyset \subset A_i \subset X \quad 1 \leq i \leq c \qquad\qquad \mathbf{0} < u_i < \mathbf{1} \quad 1 \leq i \leq c$$

Designando por u_{ik} o valor da função de pertença do elemento x_k ao agrupamento i, vem:

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall\, x_k \in X$$

$$\exists_k : 1 = u_{ik} > 0 \qquad 1 \leq i \leq c$$

$$\exists_i : 0 = u_{ik} < 1$$

Estas igualdades sugerem uma nova forma de caracterização das partições-c rígidas, feita a partir do uso de uma matriz. Tomando V_{cn} como o espaço vectorial das matrizes (cxn), $2 \leq c \leq n$, o conjunto das partições-c rígidas de **X** pode ser definido como

$$M_c = \left\{ U \in V_{cn} \mid u_{ik} \in \{0,1\}\ \forall\, i,k;\ \sum_{i=1}^{c} u_{ik} = 1\ \forall\, k;\ 0 < \sum_{k=1}^{n} u_{ik} < n\ \forall\, i \right\}$$

isto é,

— u_{ik} é 0 ou 1, conforme o elemento k pertence ou não ao agrupamento i

— $\sum_{i=1}^{c} u_{ik} = 1$, já que o elemento k pertence a um único grupo

— $0 < \sum_{i=1}^{c} u_{ik} < n$, evita as partições triviais (agrupamentos vazios e agrupamentos coincidentes com o próprio conjunto)

Exemplo 3.2

Seja $X = \{x_1$—amarelo, x_2—laranja, x_3—vermelho$\}$. Duas das possíveis partições-2 rígidas de **X** são representadas pelas seguintes matrizes

$$U_1=\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad U_2=\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

U_1 corresponde a uma partição em que o amarelo e o laranja formam um agrupamento e o vermelho constitui o segundo agrupamento. A U_2 corresponderá um agrupamento formado pelo amarelo e pelo vermelho e um outro constituído pelo laranja.

Outras matrizes poderiam ser construídas, correspondentes a outras tantas partições, mas nas condições referidas para as partições rígidas não se conseguiria uma em que o laranja fosse partilhado pelas duas cores que lhe dão origem, o amarelo e o vermelho, como seria de esperar.

No sentido de ultrapassar tal obstáculo, Ruspini definiu, de modo análogo ao anterior, **partição-c imprecisa** de **X**. Seja $X = \{x_1, x_2, \ldots\ldots, x_n\}$ e seja V_{cn} o conjunto das matrizes (c×n), $2 \leq c \leq n$. Então, o espaço das partições-c imprecisas de **X** pode ser representado por

$$M_{fc} = \left\{ U \in V_{cn} \mid u_{ik} \in [0,1] \ \forall i,k; \ \sum_{i=1}^{c} u_{ik} = 1 \ \forall k; \ 0 < \sum_{k=1}^{n} u_{ik} < n \ \forall i \right\}$$

A principal diferença entre as definições de M_{fc} e M_c é que para as partições imprecisas os u_{ik} tomam valores no intervalo [0,1] ao contrário do que acontece para as partições rígidas, para as quais os elementos só têm como valores possíveis 0 ou 1.

Voltando ao exemplo 3.2, pode-se definir uma nova matriz U_3

$$U_3=\begin{bmatrix} 0.91 & 0.55 & 0.13 \\ 0.09 & 0.45 & 0.87 \end{bmatrix}$$

em que, na partição que lhe corresponde, o laranja é partilhado por ambos os agrupamentos, facto que melhor traduz a realidade.

Para melhor compreensão da principal diferença entre partições imprecisas e partições rígidas enuncie-se o seguinte teorema:

Suponha-se $U \in M_{fc}$ uma partição-c imprecisa de **X**. Então

$$\left\{ \begin{array}{c} u_i \wedge u_j = u_{i \cap j} = 0 \\ \text{para todo } i=j \end{array} \right\} \Leftrightarrow U \in M_c$$

isto é, afinal **U** é rígida.

Entende-se como **partições degeneradas**, rígidas M_{co} ou imprecisas M_{fco}, aquelas em que

$$0 \leq \sum_{k=1}^{c} u_{ik} \leq n \quad \forall \ i$$

isto é, aquelas em que existem agrupamentos vazios e agrupamentos constituídos por todos os elementos do conjunto dado.

3.5. Relações Imprecisas

Muitos dos processos reais geram conjuntos de dados definidos por relações entre elementos, não se tendo acesso a estes mesmos elementos. Sendo assim, parece conveniente abordar-se o caso da caracterização das partições por uma **matriz de relação R.**

No caso das partições rígidas, a relação existente entre pares de pontos é inequívoca:

x_i e x_j estão relacionados $\Leftrightarrow$ x_i e x_j pertencem ao mesmo grupo.

Pode, neste caso, definir-se a matriz R como sendo uma matriz (n×n), cujos elementos verificam o seguinte:

$$r_{ii} = 1 \qquad 1 \le i \le n \tag{3.2.a}$$

$$r_{ij} = r_{ji} \quad 1 \le i \ne j \le n \tag{3.2.b}$$

$$\begin{cases} r_{ij} = 1 \\ r_{jk} = 1 \end{cases} \Leftrightarrow r_{ik} = 1 \quad \forall\, i,j,k \tag{3.2.c}$$

O espaço destas matrizes é, então

$$ER_n = \left\{ R \in V_{nn} \mid r_{ij} \text{ satisfaz as condições 3.2} \right\}$$

Cada matriz de ER_n corresponde a uma única partição-c rígida, fazendo-se $r_{ij}=1$ se x_i e x_j são elementos do mesmo agrupamento e $r_{ij}=0$ caso contrário. Então, pode afirmar-se que M_c e ER_n são estruturas semelhantes $M_c \approx ER_n$.

Para as partições imprecisas a equivalência entre M_{fc} e ER_n não é tão transparente uma vez que (3.2.c) toma várias formas, havendo para uma mesma partição-c imprecisa mais do que uma matriz de relação.

Exemplo 3.3

Considere-se a seguinte partição-2 rígida de $X = \{x_1, x_2, x_3\}$

$$U_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Então, a matriz de relação R_1 de ER_3 associada a U_1, de acordo com (2) será

$$R_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Considerando, agora, a partição-2 imprecisa U_2 do mesmo conjunto **X**

$$U_2 = \begin{bmatrix} 0.91 & 0.58 & 0.13 \\ 0.09 & 0.42 & 0.87 \end{bmatrix}$$

uma das matrizes de relação é, usando a expressão $R = (U^T (\Sigma \wedge) U)$ definida em [5]

$$R_2 = \begin{bmatrix} 1.00 & 0.67 & 0.22 \\ 0.67 & 1.00 & 0.55 \\ 0.22 & 0.55 & 1.00 \end{bmatrix}$$

De R_2 conclui-se que:
— x_1 e x_2 estão mais relacionados entre si (r_{12}=0.67) do que x_2 e x_3 (r_{23}=0.55);
— x_1 e x_3 estão pouco relacionados um com o outro (r_{13}=0.22).

4. Métodos de Agrupamento Impreciso

4.1. Introdução

Suponham-se já seleccionadas as características que, por hipótese, identificam a existência de grupos no conjunto de dados. Assim, o problema a resolver consiste em, dado $\mathbf{X} = \{ x_1, x_2, \ldots\ldots, x_n \}$, encontrar um inteiro c, $2 \leq c \leq n$, e uma partição-c de **X**, correspondente a c sub-conjuntos homogéneos em **X**, designados por **agrupamentos**. Entende-se por agrupamentos homogéneos os grupos de objectos de algum modo semelhantes e o mais semelhantes possível.

Então, qual o critério de agrupamento a utilizar? Quais as propriedades matemáticas, inerentes aos próprios dados, que devem ser consideradas para a identificação dos agrupamentos?

Torna-se claro que :
a) Nenhum critério é universalmente aplicável;
b) A escolha de um, entre os vários critérios disponíveis, é sempre subjectiva e deverá tomar em atenção a estrutura do conjunto de dados.

A figura 4.1, retirada de [6], chama a atenção para estes factos. O critério de agrupamento que melhor identificou estruturas no conjunto correspondente aos casos **a** e **b**, não conduz a bons resultados no caso de **c** e **d**.

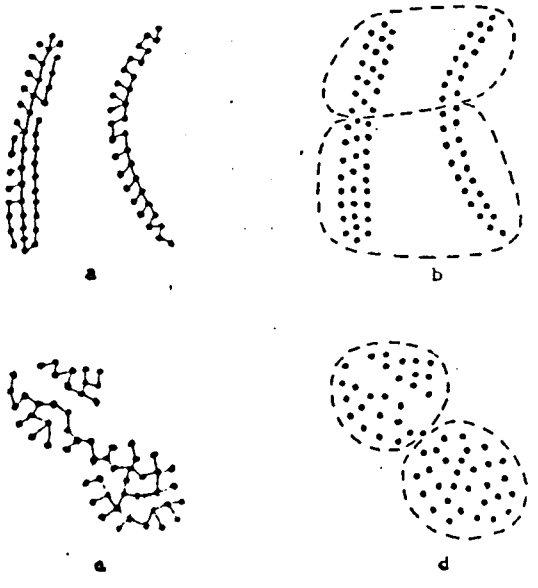


FIGURA 4.1

Os métodos usados para agrupamento podem ser classificados, de acordo com os princípios axiomáticos em que se baseiam, em :

métodos determinísticos

métodos estocásticos

métodos imprecisos

Para cada uma destas classes e de acordo com o critério de agrupamento utilizado, pode falar-se em :

métodos hierárquicos

métodos não hierárquicos

Tal como já foi referido no capítulo 2, o primeiro tipo de métodos gera uma hierarquia de agrupamentos encaixados, por meio de aglutinação ou separação de outros obtidos anteriormente.

Também já referido no capítulo 2, no caso dos métodos não hierárquicos, podem considerar-se os baseados em teoria de grafos e os que utilizam optimização de funções objectivo, estabelecendo os agrupamentos a partir de um extremo local dessa mesma função. Estes últimos permitem, normalmente, a formulação mais precisa do critério de agrupamento.

4.2. Método de Ruspini

A utilização de conjuntos imprecisos na resolução de problemas relacionados com reconhecimento de formas foi sugerida por Bellman, Kalaba e Zadeh em 1966 [7].

Várias tentativas de abordagem nesta área se seguiram, sendo a primeira exposição sistemática sobre o assunto um trabalho de Ruspini em 1970[8]. É o método de agrupamento impreciso desenvolvido por este investigador que se descreve sucintamente de seguida.

Seja $\mathbf{X} = \{x_1, x_2, \ldots\ldots, x_n\} \subset \mathcal{R}^p$ e seja d uma função distância definida da seguinte forma :

$$d : X^2 \rightarrow \mathcal{R}^+$$

$$d_{jk} = d(x_j, x_k)$$

$$d_{jk} = 0 \Leftrightarrow x_j = x_k$$

$$d_{jk} = d_{kj}$$

isto é, d é definida positiva e simétrica.

Ruspini estabeleceu como critério de agrupamento o funcional J_R, com base nas distâncias d_{jk} e nas partições-c imprecisas de **X**.

$$J_R : M_{fco} \rightarrow \mathcal{R}^+$$

$$J_R(U) = \sum_{j=1}^{n} \sum_{k=1}^{n} \left\{ \left[\sum_{i=1}^{c} \sigma \, (u_{ij} - u_{ik})^2 \right] - d_{jk}^{\,2} \right\}^2$$

onde σ é uma constante real e $2 \leq c \leq n$ é previamente fixado.

Repare-se que J_R será tanto menor quanto menores forem os termos desta expressão, o que por sua vez acontece quando pares de pontos próximos têm funções de pertença aproximadamente iguais.

Partições óptimas de **X** correspondem a mínimos locais de J_R.

O algoritmo básico de Ruspini consiste na resolução, por proceso iterativo e através do método do gradiente, do seguinte problema de minimização

$$\min_{U \in M_{fco}} \left\{ J_R(U) \right\}$$

O principal contributo dos métodos de Ruspini, dada a dificuldade da sua implementação, foi, sem dúvida, abrir caminho a outras famílias de critérios de agrupamento impreciso.

Parece importante chamar a atenção para os seguintes factos, comuns a este e muitos outros algoritmos do mesmo tipo:

a) Os pontos de convergência da sequência de matrizes de partição gerada pelo processo iterativo são pontos estacionários da função objectivo, mas não são necessariamente mínimos locais dessa mesma função.

b) Nada garante que um óptimo global de uma qualquer função objectivo seja uma boa partição de **X**. Muitas vezes os óptimos correspondem a soluções irrealistas para **X**.

c) A escolha de valores diferentes para os parâmetros algorítmicos conduz a partições óptimas de **X** distintas.

d) Pode existir mais do que um valor de c para os quais estruturas em **X** sejam identificáveis.

São estas questões que constituem a base do problema de validação das partições, que será abordado no capítulo 5.

4.3. Agrupamentos em volta de centróides

4.3.1. Hard c-means

Um dos algoritmos baseado em medidas de densidade e que gera agrupamentos rígidos em **X** é o clássico Hard c-Means de Duda e Hart[9].

Refere-se aqui este método para se dispôr de uma base de comparação para os resultados obtidos por algoritmos de partição imprecisa. A sua generalização a este tipo de partições é feita na secção 4.3.2.

Seja então,

$$J_W : M_{fco} \times \mathcal{R}^{cp} \rightarrow \mathcal{R}^{+}$$

$$J_w(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}(d_{ik})^2$$

em que

$d_{ik} = ||x_k - v_i||$ representa a distância do ponto k ao centróide do agrupamento i

$v = (v_1, v_2, \ldots\ldots, v_c) \in \mathcal{R}^{cp}$ onde $v_i \in \mathcal{R}^p$ é o centróide do agrupamento i, que não pertence necessariamente a **X**

$U = [u_{ik}] \in M_{co}$

Como $u_{ik} = 1$ se x_k pertence ao agrupamento i e $u_{ik} = 0$ se não pertence, outra forma tomada pelo funcional J_w será

$$J_w(U,v) = \sum_{i=1}^{c} \sum_{x_k \in u_i} ||x_k - v_i||^2$$

J_w será tanto menor quanto menores forem os valores dos vários d_{ik}, isto é, quanto mais aderentes estiverem os pontos do agrupamento i ao seu centróide.

Assim, a partição óptima será a correspondente a

$$\min_{M_c \times \mathcal{R}^{cp}} \left\{ J_w(U,v) \right\}$$

De acordo com a definição de **matriz rígida de dispersão do agrupamento i**

$$S_i = \sum_{k=1}^{n} u_{ik} (x_k - v_i) (x_k - v_i)^T \qquad (4.1)$$

e de **matriz rígida de dispersão entre agrupamentos**

$$S_w = \sum_{i=1}^{c} S_i$$

conclui-se que quanto menores forem os elementos de S_w mais concentrados serão os agrupamentos e mais distintos uns dos outros, tal e qual se pretende.

Descreve-se, em seguida, o algoritmo.

A.1. [Hard c-means (HCM) Duda e Hart]

A.1.1. Fixa-se c, $2 \leq c \leq n$.
Inicializa-se a matriz $U^o \in M_c$.

Para cada iteração l l= 0,1,...

A.1.2. Calculam-se os centróides $v_i^{(l)}$ através da expressão

$$v_i^{(l)} = \frac{\sum_{k=1}^{n} u_{ik}^{(l)} x_k}{\sum_{k=1}^{n} u_{ik}^{(l)}} \qquad i = 1, \ldots\ldots, c$$

A.1.3. Actualiza-se a matriz $U^{(l)}$ da seguinte forma :

$$\forall\, i,k \quad u_{ik}^{(l+1)} = \begin{cases} 1 \text{ se e só se } d_{ik}^{(l)} = \min \{d_{jk}^{(l)}\} & \\ & 1 \le j \le c \\ 0 \text{ caso contrário} & \end{cases}$$

A.1.4. Comparam-se $U^{(l)}$ e $U^{(l+1)}$:

Se $\| U^{(l+1)} - U^{(l)} \| \le \varepsilon$ termina;

senão, l passa a l+1 e volta a A.1.2.

Pode acontecer que $\min \{d_{jk}^{(l)}\}$ não seja único; neste caso, o ponto x_k é atribuído ao primeiro agrupamento para o qual se verifique esse mínimo. É o que se designa por **singularidade** deste algoritmo. Note-se que pode existir um $d_{ik}=0$, quando o ponto k coincide com o centróide do agrupamento i.

Exemplo 4.1

Considere-se o conjunto de dados **X** representado na figura 4.2.

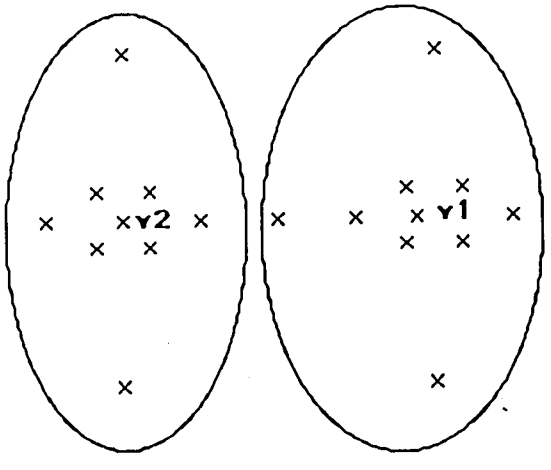


FIGURA 4.2

Este conjunto foi processado com o algoritmo anterior (HCM) e os resultados obtidos são apresentados na tabela 4.1.Fixou-se c=2 e considerou-se como matriz de partição inicial, $U^o \in M_c$, a seguinte

$$U^o = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Como patamar para o critério de paragem fez-se $\varepsilon = 0.0001$.

Repare-se que o ponto x_9 é simétrico relativamente aos outros pontos de **X**. No entanto, u_1 e u_2 não são simétricos relativamente a este ponto uma vez que, por condição de partição rígida, x_9 deve ser atribuído a um e um só agrupamento. Isto mesmo se reflecte nos centróides obtidos para os dois grupos identificados, $v_1 = (7.556, 0)$ e $v_2 = (0, 0)$, que se encontram igualmente representados na figura 4.2.

DADOS		u_{1k}	u_{2k}
x_1	(0,4)	0.00	1.00
x_2	(0,-4)	0.00	1.00
x_3	(2,0)	0.00	1.00
x_4	(-2,0)	0.00	1.00
x_5	(0.7,0.7)	0.00	1.00
x_6	(0.7,-0.7)	0.00	1.00
x_7	(-0.7,0.7)	0.00	1.00
x_8	(-0.7,-0.7)	0.00	1.00
x_9	(4,0)	1.00	0.00
x_{10}	(8,4)	1.00	0.00
x_{11}	(8,-4)	1.00	0.00
x_{12}	(10,0)	1.00	0.00
x_{13}	(6,0)	1.00	0.00
x_{14}	(7.3,0.7)	1.00	0.00
x_{15}	(7.3,-0.7)	1.00	0.00
x_{16}	(8.7,0.7)	1.00	0.00
x_{17}	(8.7,-0.7)	1.00	0.00

TABELA 4.1

4.3.2. Fuzzy c-Means

Encontrar pares óptimos (U, v) para J_w não é tarefa fácil, uma vez que o conjunto $\mathbf{M_c}$ é finito mas contem um grande número de elementos. O algoritmo que se analisa em seguida é o **Fuzzy c-Means** elaborado por Bezdeck[10] e consiste numa generalização às partições imprecisas do Hard c-Means anteriormente descrito.

O funcional que serve esta família de algoritmos é

$$J_m : M_{fc} \times \mathbb{R}^{cp} \rightarrow \mathbb{R}^+$$

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \, (d_{ik})^2 \qquad m > 1$$

Como cada termo de J_m é proporcional a d_{ik}, a solução óptima corresponderá a

$$\min_{M_{fc} \times \mathbb{R}^{cp}} \left\{ J_m \, (U, v) \right\}$$

Repare-se que se está a trabalhar no espaço $\mathbf{M_{fc}}$ que, ao contrário de $\mathbf{M_c}$, é um conjunto conexo contínuo, permitindo o uso do cálculo diferencial na determinação das condições necessárias a pontos estacionários de J_m.
Diferenciando a função objectivo relativamente a v_i (fixando **U**) e relativamente a u_{ik} (fixando **v**) e sabendo que $\sum_{i=1}^{c} u_{ik}=1$, obtem-se, como par óptimo de J_m, o par $(\mathbf{U^*}, \mathbf{v^*})$ que satisfaz às igualdades

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m \, x_k}{\sum_{k=1}^{n} (u_{ik})^m} \qquad (4.2.a)$$

e, para pontos não coincidentes com centróides

$$u_{ik} = \frac{1}{\sum_{k=1}^{n} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/m-1}} \qquad (4.2.b)$$

ou para pontos coincidentes com centróides, isto é, se $x_k = v_i$

$$u_{jk} = \begin{cases} 1 \text{ para } j=i \\ 0 \text{ para } j \neq i \end{cases} \qquad (4.2.c)$$

Descreve-se, em seguida, o algoritmo que, por processo iterativo, procura o óptimo local de J_m.

A.2. [Fuzzy c-means (FCM), Bezdeck]

A.2.1. Fixa-se c, $2 \leq c \leq n$;
Escolhe-se uma norma em $\Re^p$ e fixa-se $m \in]1, \infty[$.

Para cada iteração l= 0,1, ...

A.2.2. Calculam-se os centróides $\{v_i^{(l)}\}$ a partir de (4.2.a) e $U^{(l)}$.

A.2.3. Actualiza-se a matriz $U^{(l)}$, utilizando (4.2.b), (4.2.c) e $\{v_i^{(l)}\}$

A.2.4. Comparam-se $U^{(l+1)}$ e $U^{(l)}$.

Se $\| U^{(l+1)}-U^{(l)} \| < \varepsilon$ termina; senão, volta a A.2.2.

Este algoritmo tem vários parâmetros : c, m, U^o, $\| \|_A$, ε. Analise-se o que se passa com o parâmetro m.

Para cada **A** existe uma família de soluções parametrizadas por m. À medida que m tende para 1 por valores superiores, a solução converge para uma solução rígida; J_m aproxima-se de J_W que passa a ter uma variável adicional, **A**. Contrariamente, à medida que $m \rightarrow \infty$ a matriz $U \rightarrow [1/c] \quad \forall$ i,k, como se deduz facilmente a partir de (4.2.a) e (4.2.b).

Resumindo, quanto maior for m mais imprecisa é a solução encontrada e quanto mais próximo de 1 estiver m "mais rígida" é a solução. Então, o expoente m funciona como uma espécie de controle à partilha de pontos por agrupamentos diferentes.

Outro dos parâmetros algorítmicos é a matriz definidora da norma em $\Re^p$. Se **A** for a matriz identidade, trabalha-se, então, em geometria euclideana e o critério conduz a bons resultados para conjuntos **X** onde se possam identificar agrupamentos "circulares", uma vez que aquela geometria é hiperesférica em $\Re^p$.

No entanto, isto nem sempre se verifica. Assim sendo, pode falar-se noutro tipo de normas: a norma diagonal, N_D, induzida a partir da matriz das variâncias das coordenadas dos pontos de **X**

$$A_D = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \dots & 0 \\ 0 & (1/\sigma_2)^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & (1/\sigma_p)^2 \end{bmatrix}$$

em que

$$\sigma_j^2 = \frac{1}{n} \sum_{k=1}^{n} (x_{kj} - \overline{x}_j)^2 \quad \text{e} \quad \overline{x}_j = \frac{1}{n} \sum_{k=1}^{n} x_{kj}$$

e a norma de Mahalanobis, N_M, induzida pela inversa da matriz das covariâncias de **X**

$$[\mathrm{Cov}\,(\mathbf{X})]_{ij} = \frac{1}{n} \sum_{k=1}^{n} (x_{ki} - \overline{x}_i)\,(x_{kj} - \overline{x}_j)$$

Uma e outra norma tentam compensar distorções à forma circular básica; no caso da norma N_D, devidas a grandes variâncias nas componentes dos pontos de **X**, ou, no caso da norma N_M, devidas a grandes variâncias das componentes e à dependência estatística verificada entre as componentes medidas. Então, **A** pode ser tomada como uma variável de optimização, como no algoritmo de Gustafson e Kessel, que se aborda na secção 4.4 .

Exemplo 4.2

Considere-se o conjunto **X** definido no exemplo 4.1. Processou-se **X** com o FCM, fazendo c=2, ε=0.0001, norma euclideana e a mesma partição inicial. Tomou-se m=2 e os resultados obtidos para a função de pertença ao agrupamento 1, encontram-se na tabela 4.2, assim como os que se obtiveram tomando m=1.75, m=1.50 e m=1.25. Note-se que estes últimos devem ser considerados como arredondamentos e não como valores exactos obtidos pelo algoritmo. Os centróides, para os diferentes valores de m, mantêm-se aproximadamente em (7.807,0) e (0.190,0).

DADOS	m=2.00	m=1.75	m=1.50	m=1.25
x_1	0.172	0.110	0.042	0.002
x_2	0.172	0.110	0.042	0.002
x_3	0.088	0.043	0.009	0.000
x_4	0.048	0.018	0.002	0.000
x_5	0.014	0.004	0.001	0.000
x_6	0.014	0.004	0.001	0.000
x_7	0.017	0.005	0.001	0.000
x_8	0.017	0.005	0.001	0.000
x_9	0.500	0.500	0.500	0.500
x_{10}	0.828	0.890	0.958	0.998
x_{11}	0.828	0.890	0.958	0.998
x_{12}	0.952	0.982	0.998	1.000
x_{13}	0.912	0.957	0.991	1.000
x_{14}	0.986	0.996	0.999	1.000
x_{15}	0.986	0.996	0.999	1.000
x_{16}	0.983	0.995	0.999	1.000
x_{17}	0.983	0.995	0.999	1.000

TABELA 4.2

Repare-se que:

a) Ao contrário do que acontecia por aplicação de algoritmos de partição rígida, em todos os resultados o ponto x_9 é partilhado pelos dois agrupamentos com igual valor de pertença, como é indicado no próprio conjunto de dados.

b) À medida que os pontos se aproximam dos centróides, caso de x_5 e x_6, apresentam funções de pertença aos vários agrupamentos mais distintas, permitindo identificar dois grupos compactos.

c) Conforme $m \xrightarrow{+} 1$, as partições obtidas tendem a aproximar-se de soluções rígidas. Atenda-se aos resultados obtidos para m=1.25. Todos os valores de pertença, à excepção dos correspondentes ao ponto x_9, são idênticos aos obtidos por aplicação do HCM. À medida que m toma valores crescentes (m=1.50, m=1.75, m=2.00) as soluções tornam-se mais imprecisas.

d) Os pontos radialmente simétricos relativamente aos centróides apresentam valores de pertença idênticos. É o caso dos pontos x_1 e x_2 bem como dos pontos x_5 e x_6. Isto acontece porque a norma utilizada é a norma euclideana, que tende a forçar agrupamentos circulares em **X**. Quando não for este o tipo de geometria contida nos dados, melhores resultados podem ser obtidos utilizando outra norma.

Exemplo 4.3

Considere-se o conjunto de dados que se representa na figura 4.3. Processou-se este conjunto com o FCM, utilizando norma euclideana e norma diagonal. Os parâmetros do algoritmo foram fixados com valores iguais aos utilizados para o exemplo anterior e fazendo m=2.00.

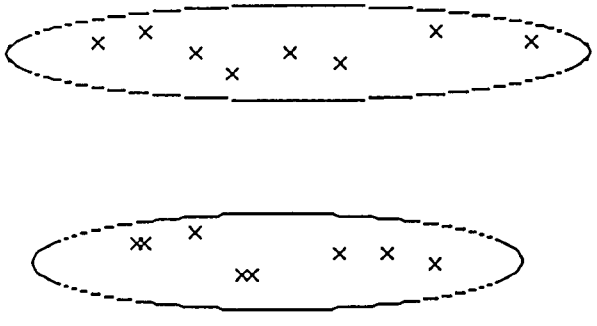


FIGURA 4.3

Os resultados obtidos para a função de pertença ao primeiro agrupamento encontram-se na tabela 4.3 e sugerem a escolha da norma diagonal para identificação dos agrupamentos como sendo a que conduz a uma partição que mais se aproxima da estrutura contida nos próprios dados.

DADOS		N_E	N_D
x_1	(0.5,1)	0.827	0.102
x_2	(3,0.8)	0.180	0.021
x_3	(1.9,0.7)	0.756	0.027
x_4	(5,1.0)	0.168	0.164
x_5	(2.5,0.9)	0.425	0.001
x_6	(1,1.1)	0.801	0.060
x_7	(1.5,0.9)	0.803	0.028
x_8	(4,1.1)	0.120	0.078
x_9	(1,-0.9)	0.914	0.939
x_{10}	(2,-1.2)	0.760	0.990
x_{11}	(3.5,-1)	0.213	0.945
x_{12}	(4,-1.1)	0.192	0.909
x_{13}	(0.9,-0.9)	0.909	0.931
x_{14}	(1.5,-0.8)	0.930	0.962
x_{15}	(2.1,-1.2)	0.729	0.991
x_{16}	(3,-1)	0.345	0.979

TABELA 4.3

Atenda-se, agora, à convergência do algoritmo. Vários resultados já obtidos asseguram a convergência do FCM, pelo menos teoricamente, à medida que o número de iterações, l, tende para infinito. Nada garante que essa convergência se verifique para um valor finito de l, mas, quando acontece, e felizmente na maior parte dos casos assim o é, os valores $\left(U^{(l)},v^{(l)}\right)$ encontrados correspondem a um mínimo local de J_m. Note-se, contudo e mais uma vez, que mínimos locais de J_m nem sempre são uma "boa" partição do conjunto de dados.

4.3.3 Algoritmo de Gustafson e Kessel

No Fuzzy c-Means, um dos parâmetros a fixar no início do algoritmo é a matriz definidora da norma, matriz A, como já foi referido. Apesar de se poderem induzir várias métricas em $\Re^p$, através da escolha da norma que se utilizará, essa métrica mantem-se constante para todos os agrupamentos a identificar em X. Ora, podem surgir situações em que estes mesmos agrupamentos tenham formas geométricas diferentes e, nestas condições, o funcional J_m não conduz a resultados satisfatórios.

Na tentativa de ultrapassar este problema, Gustafson e Kessel [11] propuseram que, em vez de se manter a norma constante, houvesse uma variação local dessa mesma norma, uma vez que é esta que controla a forma dos sub-conjuntos.

Sendo assim, foi estabelecida uma nova função objectivo, $\hat{J}_m$, que torna possível a identificação de grupos correspondentes a estruturas topológicas distintas, contidas no conjunto de dados.

Seja, então, A_i a matriz definidora da norma para o agrupamento i e seja **A** o vector das matrizes de norma dos c sub-conjuntos de **X**. Assim,

$$\hat{J}_m : M_{fc} \times \mathcal{R}^{cp} \times PD^c \rightarrow \mathcal{R}^+$$

$$\hat{J}_m (U,v,A) = \sum_{k=1}^{n} \sum_{k=1}^{n} (u_{ik})^m \; \|x_k - v_i\|^2_{A_i}$$

onde PD^c representa o produto cartesiano do conjunto das matrizes (p×p) definidas positivas simétricas

$U \in M_{fc}$

$v = \{v_1, v_2, \ldots\ldots, v_c\} \in \mathcal{R}^{cp}$

$m \in]1,+\infty[$

O critério de agrupamento estabelecido a partir deste novo funcional coincide com o que utiliza J_m. A diferença fundamental é que, enquanto neste último as distâncias são medidas por uma norma pré-especificada, em $\hat{J}_m$ a norma varia para os diferentes grupos. Repare-se que, se A_i for constante $\forall i$, $\hat{J}_m (U,v,A) = J_m (U,v)$.

Então, o problema a resolver consiste em encontrar as soluções de

$$\min_{M_{fc} \times R^{cp} \times PD^c} \left\{ \hat{J}_m (U,v,A) \right\}$$

Estabeleçam-se as condições necessárias a pontos estacionários de $\hat{J}_m$. No caso de se fixar **A**, como já foi referido $\hat{J}_m(U,v,A) = J_m(U,)$; logo, as soluções encontradas para (U^*,v^*) no caso da minimização de J_m são igualmente valores óptimos para a minimização de $\hat{J}_m$.

Resta deduzir a forma óptima para o vector **A**, A^*. Para isso, fixem-se **U** e **v** e aplique-se o método dos multiplicadores de Lagrange à nova função objectivo, tomando m >1 e fixando $\det(A_i) = \rho_i$. Virá

$$A_i^* = \left[\rho_i \det(S_{fi})\right]^{1/p} (S_{fi})^{-1} \qquad 1 \le i \le c \qquad (4.3)$$

onde

$$S_{fi} = \sum_{k=1}^{n} (u_{ik})^m \ (x_k - v_i) \ (x_k - v_i)^T$$

é a **matriz imprecisa de dispersão do agrupamento i**, por analogia com (4.1).

Descreve-se agora o algoritmo, que pode ser entendido, de alguma forma, como uma generalização do FCM descrito na secção anterior.

A.3. [Gustafson e Kessel]

A.3.1. Fixa-se c, $2 \leq c \leq n$.
Escolhe-se $m > 1$.
Fixam-se os valores de $\rho_i > 0$, $1 \leq i \leq c$.
Inicializa-se $U^o \in M_{fc}$.

Em cada iteração l l=0,1,...

A.3.2. Calculam-se os centróides $\left\{v_i^{(l)}\right\}$ a partir de (4.2.a) e $U^{(l)}$.

A.3.3. Determinam-se as matrizes $\left\{S_{fi}^{(l)}\right\}$ utilizando (4.1) e $\left(U^{(l)}, v^{(l)}\right)$. Calculam-se os respectivos determinantes e inversas.

A.3.4. Utilizando (4.3), calculam-se as matrizes $\left\{A_i^{(l)}\right\}$.

A.3.5. Actualiza-se a matriz $U^{(l)}$ utilizando $v_i^{(l)}$, (4.2.b) e (4.2.c), não esquecendo que $d_{ik}^{(l)} = \|x_k - v_i^{(l)}\|_{A_i}$.

A.3.6. Comparam-se $U^{(l)}$ e $U^{(l+1)}$: Se $\|U^{(l+1)} - U^{(l)}\| < \varepsilon$ pára; senão volta a A.3.2.

A relação encontrada entre a matriz A_i^* e a matriz S_{fi} conduziu a que se definisse uma nova matriz, designada por **matriz de covariância imprecisa**, C_{fi}

$$C_{fi} = \frac{\sum_{k=1}^{n} (u_{ik})^m \ (x_k - v_i) \ (x_k - v_i)^T}{\sum_{k=1}^{n} (u_{ik})^m} = \frac{S_{fi}}{\sum_{k=1}^{n} (u_{ik})^m}$$

No caso de U ser uma matriz rígida, C_{fi} reduz-se a

$$C_i = \frac{\sum_{x_k \in u_i} (x_k - v_i) \ (x_k - v_i)^T}{n_i}$$

que é a matriz de covariâncias dos n_i pontos do grupo i com centróide v_i. Daqui, o facto desta família de algoritmos ser igualmente designada por algoritmos de covariância imprecisa.

Fazendo

$$\lambda_{ik} = \frac{(u_{ik})^m \, [\rho_i \det(S_{fi})]^{1/p}}{\sum_{t=1}^{n} (u_{it})^m} \qquad 1 \le i \le c \text{ e } 1 \le t \le n$$

o funcional $\hat{J}_m$, no ponto óptimo, poderá ser escrito da seguinte forma

$$\hat{J}_m \, (U^*, v^*, A^*) = \sum_{k=1}^{n} \left[\lambda_{ik} \, (x_k - v_i)^T \, C_{fi}^{-1} \, (x_k - v_i) \right]$$

O valor de $(x_k - v_i)^T \, C_{fi}^{-1} \, (x_k - v_i)$ não é mais do que o quadrado da distância de Mahalanobis entre x_k e v_i, na forma imprecisa. Então, os resultados obtidos, U^*, v^*, A^*, podem ser interpretados como aqueles que conduzem a uma minimização da dispersão dos pontos em torno dos centróides, conseguida por uma combinação de diferentes normas na mesma função objectivo.

Propriedades de convergência deste algoritmo supõem-se semelhantes às referidas para o caso do Fuzzy c-Means.

Um exemplo de aplicação desta técnica de agrupamento encontra-se na secção 6.3 do capítulo 6, onde se faz a análise dos resultados obtidos.

Refiram-se outros algoritmos, desenvolvidos com objectivos semelhantes aos que conduziram àquele que se acaba de descrever: os designados, genericamente, por **Fuzzy c-Varieties** (FCV) que utilizam a mesma norma para todos os grupos, mas em que os centróides deixam de ser pontos em $\mathbb{R}^p$ e passam a ser variedades lineares de dimensão r, $0 \le r \le p-1$; e os **Fuzzy c-Elliptotypes** (FCE) em que os centróides são, igualmente, variedades lineares mas de dimensões diferentes e a função objectivo é uma combinação convexa de funções definidas a partir das distâncias dos pontos a cada uma dessas mesmas variedades lineares.

5. A Validação das Partições

5.1. Introdução

O objectivo da validação das partições consiste em medir o grau de ajuste entre a partição gerada por qualquer um dos algoritmos disponíveis e as estruturas contidas no conjunto de dados, uma vez que os dados e as possíveis partições se encontram separados por esse mesmo algoritmo.

Processando um conjunto **X** por qualquer um dos algoritmos conhecidos obter-se-á, para cada valor de c, um grande número de partições (note-se que existem vários parâmetros impostos inicialmente para os diversos processos de análise). Mais, é possível encontrar "bons" agrupamentos para diferentes valores de c. Então, como saber qual das partições obtidas é a óptima, no sentido de melhor identificar as estruturas existentes nos dados?

Para métodos que utilizam optimização de funções objectivo, uma estratégia, que à partida se julgaria óbvia, consiste em registar os óptimos destas como uma função de c e a partição a escolher será a correspondente a um extremo desta nova função. No entanto, um problema se levanta: a maior parte das funções objectivo são monótonas em c e, além disso, apresentam, para cada c, vários pontos estacionários, podendo o óptimo global não corresponder a uma "boa" partição de **X**.

Como consequência, várias medidas de validade das partições foram estabelecidas, os **indicadores de validade**, para avaliarem da qualidade das soluções encontradas. Note-se, no entanto, que a maior parte destes escalares não têm ligação directa ao conjunto de dados, cabendo sempre ao analista a avaliação final dos resultados obtidos.

5.2. Coeficiente de Partição

Seja $U \in M_{fc}$ uma partição-c imprecisa de um conjunto **X** de n pontos. O **coeficiente de partição** [12] é definido como sendo o escalar

$$F(U,c) = \frac{\sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^2}{n}$$

Demonstra-se que, para $1 \leq c \leq n$ vem:

$$\frac{1}{c} \leq F(U,c) \leq 1$$

$$F(U,c) = 1 \Leftrightarrow U \in M_{co} \text{ é rígida}$$

$$F(U,c) = \frac{1}{c} \Leftrightarrow U = [1/c]$$

isto é, F é 1 para todas as partições rígidas e toma o valor mínimo apenas para $U = [1/c]$,

isto é, o centróide de M_{fc}. Note-se que F é contínua e estritamente convexa em M_{fc} que, por sua vez, é um conjunto convexo; logo, o mínimo global de F em M_{fc} é único.

Se se definir uma matriz S em que

$$s_{ij}=\frac{\sum_{k=1}^{n} u_{ik}u_{jk}}{n}$$

isto é, cada elemento de S mede o nível de partilha dos pontos de X por cada par de agrupamentos, F tomará uma nova forma

$$F(U,c) = \frac{\sum_{i=1}^{c} s_{ii}}{\sum_{i=1}^{c} s_{ii} + 2 \sum_{i=j+1}^{c} \sum_{j=1}^{c-1} s_{ij}}$$

Assim, F será máximo quando $s_{ij}=0$, isto é, quando não houver partilha de pontos pelos vários agrupamentos, o que equivale a dizer quando U representar uma partição rígida.

Em consequência do que acaba de ser dito, poder-se-ia ser levado a pensar que qualquer partição rígida é mais válida que uma imprecisa. Com efeito assim não é. O conjunto definido no exemplo 4.1 do capítulo anterior demonstra-o. O que na realidade se passa é que, se X contiver estruturas distintas, os algoritmos deverão construir matrizes U relativamente rígidas, tal e qual são medidas por F. É esta a lógica subjacente ao uso de F como indicador de validade.

A estratégia a seguir é, então

$$\max_{c} \left\{ \max_{\Omega_c} \{ F(U,c) \} \right\}$$

onde Ω_c representa o conjunto de partições óptimas para cada c.

Note-se, mais uma vez, que se $F(U^*, c^*)$ não estiver próximo de 1, não se deverá supôr que X não tem estruturas identificáveis, mas sim que o algoritmo não foi capaz de as distinguir.

Exemplo 5.1

Considere-se o conjunto X representado na figura 5.1, em que se identificam facilmente, por observação directa, 3 grupos distintos. Nesta figura ilustram-se igualmente os agrupamentos derivados, da partição obtida, pela regra da máxima pertença. Segundo esta regra, uma matriz $U^* \in M_{fco}$ é convertida numa matriz $U_m \in M_{co}$ fazendo

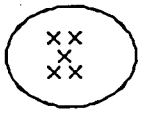
$$u_{m_{ik}} = \begin{cases} 1 \text{ se } u_{ik} = \max\limits_{1 \le j \le c} \{u_{jk}\} \\ 0 \end{cases}$$

Este conjunto foi processado com FCM, tomando como valores para os parâmetros os definidos para o exemplo 4.2 do capítulo anterior. Os resultados encontrados apresentam-se na tabela 5.1.

c	F
2	0.856
3	0.968
4	0.878
5	0.829

TABELA 5.1

A função F(c) apresenta um extremo local para c=3, sugerindo o indicador F este valor de c como sendo óptimo. Note-se que para as outras escolhas de c, o indicador afasta-se claramente daquele valor.



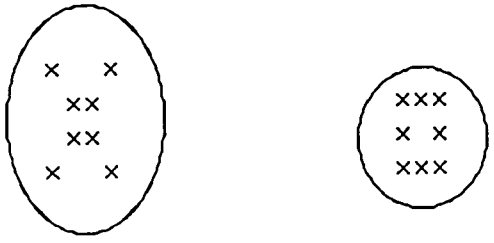


FIGURA 5.1

Outros conjuntos foram formados a partir deste, diminuindo gradualmente a distância entre os dois grupos inferiores, no sentido de se observar o que se passa com F.

À medida que os grupos se vão aproximando o valor do indicador para c=3 diminui e para c=2 aumenta até se chegar a uma distribuição de pontos no espaço para a qual F(2)>F(3). Na tabela 5.2 resumem-se estes resultados.

	distância entre os dois grupos			
c	15	9	3	2
2	0.856	0.858	0.883	0.915
3	0.968	0.949	0.891	0.865
4	0.878	0.834	0.775	0.751
5	0.829	0.808	0.706	0.693

TABELA 5.2

O valor de F pode, igualmente, sugerir indicações a respeito da norma a utilizar no tratamento do conjunto de dados. Tal como já foi referido, a norma, a partir da qual se definem as distâncias em X, é um dos parâmetros algorítmicos e controla a forma dos agrupamentos que se procuram (norma euclideana, norma diagonal ou norma de Mahalonobis). Fazendo variar a norma, podem-se inferir propriedades a respeito dos

dados a partir da "melhor" partição indicada por F. É preciso, todavia, interpretar os resultados com cautela, pois um melhor "arranjo" dos valores de pertença, obtidos em função da norma adoptada, não implicam necessariamente que a partição seja melhor, quando apreciada do ponto de vista humano.

Exemplo 5.2

No sentido de se verificar o que acaba de ser referido, atenda-se aos valores do indicador F, para o exemplo 4.3 do capítulo anterior no caso de se utilizarem as duas normas, a norma euclideana e a norma diagonal.

c	N_E	N_D
2	0.694	0.772
3	0.690	0.729
4	0.633	0.691

TABELA 5.3

Como se verifica, os valores de F quando se usa a norma diagonal são superiores aos que se obtêm com a norma euclideana, coincidindo com o facto de que neste caso, o uso da primeira conduz a melhores partições de **X**.

As maiores desvantagens deste indicador são:
— A sua variação monótona com c
— A falta de ligação directa entre os seus valores e as estruturas contidas nos dados.

5.3. Entropia da Classificação

Como os conjuntos imprecisos representam um certo tipo de incerteza, incerteza essa devida à falta de precisão na definição do conjunto, parece natural estabelecer para estes um conceito semelhante ao de entropia ligada à incerteza estatística. Recorde-se:

Seja $\{A_i \mid 1 \leq i \leq c\}$ um conjunto de acontecimentos ligados a uma experiência e seja $\mathbf{p} = (p_1, p_2, \ldots\ldots, p_c)$ o vector das probabilidades de ocorrência desses mesmos acontecimentos. A **entropia estatística** ligada ao par $(\{A_i\}, \mathbf{p})$ é definida como sendo o escalar

$$h(\mathbf{p}) = -\sum_{i=1}^{c} p_i \log_a(p_i)$$

onde $a \in]1,+\infty[$ e $p_i \log_a(p_i) = 0$ para $p_i = 0$.

Esta medida goza das seguintes propriedades:

$$0 \leq h(\mathbf{p}) \leq \log_a c$$

$$h(\mathbf{p}) = 0 \Leftrightarrow \mathbf{p} \text{ é certo } (\exists_i: p_i = 1)$$

$$h(p) = \log_a c \Leftrightarrow p = \frac{1}{c}$$

para todo o $\mathbf{p} = (p_1, p_2, \ldots\ldots, p_c)$ tal que $\sum_{i=1}^{c} p_i = 1 \quad p_i \geq 0 \ \forall i$

Por analogia com este conceito, estabeleceu-se o de **entropia da partição** [13], uma vez que para todo o $\mathbf{U} \in \mathbf{M_{fc}}$ se verifica $\sum_{i=1}^{c} u_{ik} = 1$, como acontece com o vector **p** de $h(\mathbf{p})$.

Sendo assim, define-se para cada partição imprecisa, $\mathbf{U} \in \mathbf{M_{fc}}$, de **X**, $|\mathbf{X}|=n$, o escalar

$$H(\mathbf{U},c) = -\frac{\sum_{j=1}^{n} \sum_{i=1}^{c} u_{ik} \log_a u_{ik}}{n}$$

com $a \in]1,+\infty[$ e $u_{ik}\log_a u_{ik} = 0$ se $u_{ik} = 0$.

Atendendo a que $\sum_{i=1}^{c} u_{ik}=1 \ \forall k$, uma forma alternativa de representar H é

$$H(\mathbf{U},c) = \frac{\sum_{k=1}^{n} h(u_k)}{n} \qquad (5.1)$$

em que u_k representa a coluna k da matriz $\mathbf{U} \in \mathbf{M_{fc}}$.

Com base na expressão (5.1) e atendendo às propriedades enunciadas para h(**p**), é fácil chegar às seguintes conclusões:

$$0 \leq H(\mathbf{U},c) \leq \log_a c$$

$$H(\mathbf{U},c) = 0 \Leftrightarrow \mathbf{U} \text{ é rígida}$$

$$H(\mathbf{U},c) = \log_a c \Leftrightarrow \mathbf{U} = [1/c]$$

Procurando interpretar o conceito de entropia da partição pode dizer-se que $h(u_k)$ representa o grau de incerteza a respeito da pertença do ponto x_k aos vários agrupamentos; a incerteza será tanto maior quanto mais "imprecisa" for a coluna k da matriz **U**. Então, a minimização de H(**U**,c) corresponde a maximizar a informação sobre estruturas existentes num conjunto **X**, sugerida por um algoritmo de partição.

Assim, a estratégia a seguir para validação das partições será

$$\min_{c} \left\{ \min_{\Omega_c} \{ H(\mathbf{U},c) \} \right\} \qquad (5.2)$$

com Ω_c a representar o conjunto finito dos $\mathbf{U} \in \mathbf{M_{fc}}$ óptimos.

O par (U^*, c^*), correspondente à solução de (5.2), supõe-se ser a partição "mais válida" entre as consideradas em Ω_c. Convem lembrar, no entanto, que a utilização deste indicador de validade deve ser feita no mesmo contexto em que foi feita a de F(U,c), isto é, se existem estruturas em **X** a identificar, os algoritmos devem produzir **U**'s relativamente rígidas.

Repare-se que F e H apresentam extremos para as mesmas partições, embora em sentidos contrários. Mais, é possível estabelecer uma ligação directa entre os seus valores. Pode-se demonstrar que para $U \in M_{fc}$, $a \in]1,+\infty[$ e sendo **e** a base natural

$$0 \leq 1\text{-}F(U,c) \leq \frac{H(U,c)}{\log_a e}$$

verificando-se a igualdade apenas para $U \in M_{co}$. Se a=e, então vem

$$0 \leq 1\text{-}F(U,c) \leq H(U,c)$$

Como inconvenientes a apontar ao uso deste indicador referem-se:
— A tendência monótona com c
— A falta de um meio de avaliar a aceitação ou não da partição sugerida por H como uma boa interpretação do conjunto de dados.

Exemplo 5.3

O conjunto representado na figura 5.2 foi processado atrav és do mesmo algoritmo utilizado nos exemplos anteriores e tomando os mesmos valores para os parâmetros. Os resultados obtidos para os indicadores F e H são apresentados na tabela 5.4. Na figura estão identificados os agrupamentos correspondentes a U_m para c=2 e para c=3 .

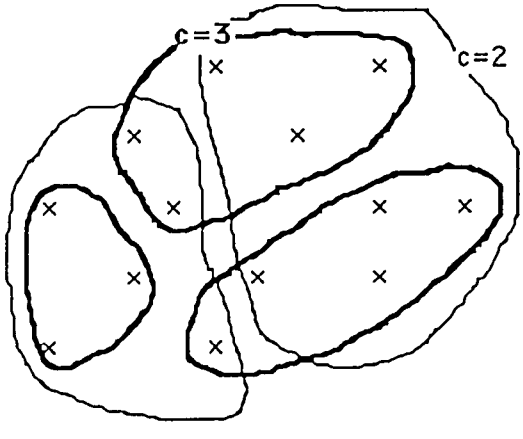


FIGURA 5.2

Ambos os indicadores sugerem $c^*=2$ como o valor óptimo para c no que respeita às estratégias desenvolvidas com base nos valores de F e H. Note-se ainda que estes valores estão algo longe dos respectivos limites que são 1 e 0, já que no conjunto, como se pode observar, as estruturas a identificar não são realmente distintas. Nas tabelas 5.5 e 5.6 encontram-se os valores de pertença dos diferentes pontos aos agrupamentos definidos, para c=2 e c=3. Neste último caso, há mais incerteza na atribuição dos pontos aos grupos.

c	F	H
2	0.724	0.431
3	0.653	0.541
4	0.610	0.765
5	0.580	0.833

TABELA 5.4

DADOS	u_{1k}	u_{2k}
x1	0.143	0.857
x2	0.091	0.989
x3	0.020	0.980
x4	0.251	0.749
x5	0.269	0.731
x6	0.322	0.678
x7	0.584	0.416
x8	0.884	0.116
x9	0.651	0.349
x10	0.832	0.168
x11	0.980	0.020
x12	0.843	0.157
x13	0.892	0.109

TABELA 5.5

DADOS	u_{1k}	u_{2k}	u_{3k}
x1	0.062	0.071	0.867
x2	0.089	0.212	0.699
x3	0.010	0.013	0.977
x4.	0.100	0.664	0.236
x5	0.201	0.134	0.665
x6	0.202	0.508	0.290
x7	0.063	0.894	0.043
x8	0.172	0.781	0.047
x9	0.627	0.171	0.202
x10	0.903	0.048	0.049
x11	0.971	0.024	0.005
x12	0.361	0.540	0.009
x13	0.852	0.100	0.048

TABELA 5.6

5.4. Expoente de Proporção

Outro dos indicadores de validade é o **expoente de proporção** [14], definido pela expressão

$$P(U,c) = - \ln \left\{ \prod_{k=1}^{n} \left[\sum_{j=1}^{\mu_k^{-1}} (-1)^{j+1} \binom{c}{j} \left(1 - j \bigvee_{i=1}^{c} u_{ik} \right)^{c-1} \right] \right\}$$

onde

$U \in (M_{fc}\text{-}M_{co})$

$|X| = n$

$2 \leq c \leq n$

$\mu_k = \max_{1 \leq i \leq c} \{u_{ik}\}$

μ_k^{-1} = maior inteiro $\leq \frac{1}{\mu_k}$

Note-se que P utiliza apenas os valores máximos de cada uma das colunas de **U**, ao contrário de F e H que utilizam todos os elementos da matriz, o que se explica pelo facto de grandes valores de μ_k sugerirem a existência de estruturas "mais distintas" em **X**.

O uso deste escalar como indicador de validade está relacionado com o seguinte conceito: se se definir o conjunto S_μ como o conjunto das colunas das matrizes do espaço M_{fc}-M_{co} com um elemento máximo não inferior a μ, onde $1/c \leq \mu \leq 1$, $2 \leq c \leq n$, e representando por μ^{-1} o maior inteiro $\leq 1/\mu$, diz-se que a proporção de S_μ é dada por

$$p(S_\mu) = \sum_{j=1}^{\mu^{-1}-1} (-1)^{j+1} \binom{c}{j} (1 - j\mu)^{c-1}$$

Por exemplo, se $p(S_\mu) = 0.12$ isto significa que entre todas as colunas imprecisas possíveis, 12% têm um elemento máximo que, pelo menos, é igual a μ.

É fácil verificar que $p(S_\mu)=0$ para $\mu=1$ e $p(S_\mu)=1$ para $\mu= 1/c$. Assim sendo, faça-se na expressão de P

$$\eta\,(U,c) = \prod_{k=1}^{n} \left[\sum_{j=1}^{\mu_k^{-1}-1} (-1)^{j+1} \binom{c}{j} \left(1 - j \bigvee_{i=1}^{c} u_{ik}\right)^{c-1} \right]$$

Este valor pode ser interpretado como a proporção de matrizes em M_{fc}-M_{co} que apresentam, em todas as colunas (daí o sinal de produto), um valor máximo superior aos da matriz U em questão; isto é, η representa a proporção de partições-c que melhor identificam agrupamentos em X do que aquela em questão. Repare-se que $\eta(U,c)=0$ se alguma das colunas de U for rígida e $\eta(U,c)=1$ se e só se $U=\left[1/c\right]$.

Para calcular o valor de P toma-se o simétrico do ln η. Porquê? η pode tomar valores muito próximos de zero para matrizes mais ou menos imprecisas (basta que uma coluna se aproxime de valores rígidos). Quando se toma o ln η, o valor torna-se mais sensível a variações de U. Então,

$$0 \leq P(U,c) < \infty \qquad U \in M_{fc}\text{-}M_{co}.$$

Logo, a estratégia a seguir será

$$\max_{2\leq c<n} \left\{ \max_{U} \{P(U,c)\} \right\}$$

Atenda-se ao seguinte:

— O domínio de aplicação dos indicadores anteriores é ligeiramente mais amplo que o de P, o que os torna mais gerais;

— O facto de uma coluna se tornar rígida conduz P a infinito, o que leva a que este indicador não seja muito fiável para partições que se aproximem da solução rígida. Pelo contrário, quanto mais U se afasta de M_{co}, melhor é o comportamento de P.

— Tal como para F e H , também relativamente a P se põe a questão de saber até que valor poderá descer $P(U^*,c^*)$ para se considerar U^* uma "boa" partição de X.

5.5. Normalização de H

Dunn [15] observou que, relativamente a H, o valor deste se aproxima de zero para c=1 e para c=n, o que contraria toda a estratégia baseada na sua minimização. Como solução, foi proposta a normalização deste indicador.

Assim, tome-se um conjunto $\mathbf{X}_o$ em que os pontos se encontram uniformemente distribuídos nas diferentes direcções, apresentando o conjunto estruturas de agrupamento apenas para c=1 e c=n (correspondentes às partições triviais). Então, a validação da partição poderá ser feita com base no desvio entre o valor de H (referente ao conjunto **X**) e o valor de H_o (referente ao conjunto $\mathbf{X}_o$), maximizando esse desvio.

Baseado em resultados obtidos processando $\mathbf{X}_o$ com o FCM, Dunn supôs que se poderia representar H_o através da equação

$$H_o(c) = 1 - \frac{c}{n}$$

Definindo **entropia normalizada** como $\hat{H}(\mathbf{U},c)$ em que

$$\hat{H}(\mathbf{U},c) = \frac{H(\mathbf{U},c)}{H_o(c)}$$

vem

$$\hat{H}(\mathbf{U},c) = \frac{H(\mathbf{U},c)}{1 - c/n}$$

Como H é sempre superior a H_o (o conjunto $\mathbf{X}_o$ é um conjunto muito desordenado) a nova estratégia de validação será

$$\min_{2\le c<n} \left\{ \min_{\Omega_c} \left\{ \hat{H}(\mathbf{U},c) \right\} \right\}$$

Note-se que Ω_c corresponde a soluções do funcional J_m, uma vez que a equação estabelecida para H_o foi baseada em resultados obtidos pelo FCM.

$\hat{H}(\mathbf{U},c)$ e H têm comportamentos aproximadamente iguais para c<<1, o que corresponde à maior parte dos casos, sugerindo os dois indicadores as mesmas soluções. Mas à medida que $\frac{c}{n} \to 1$, $\hat{H}$ comporta-se de um modo diferente de H e é, provavelmente, um melhor indicador, uma vez que referencia a solução óptima $(\mathbf{U}^*,c^*)$ a uma hipótese nula àcerca das estruturas em **X**.

Exemplo 5.4

Considere-se o conjunto constituido por 9 pontos, distribuidos no espaço da forma representada na figura 5.3 [15]. Aplicou-se a este conjunto o FCM, tomando ε=0.001,norma Euclideana e m=2. Na tabela 5.7 encontram-se os resultados obtidos para os indicadores F, H e $\hat{H}$.

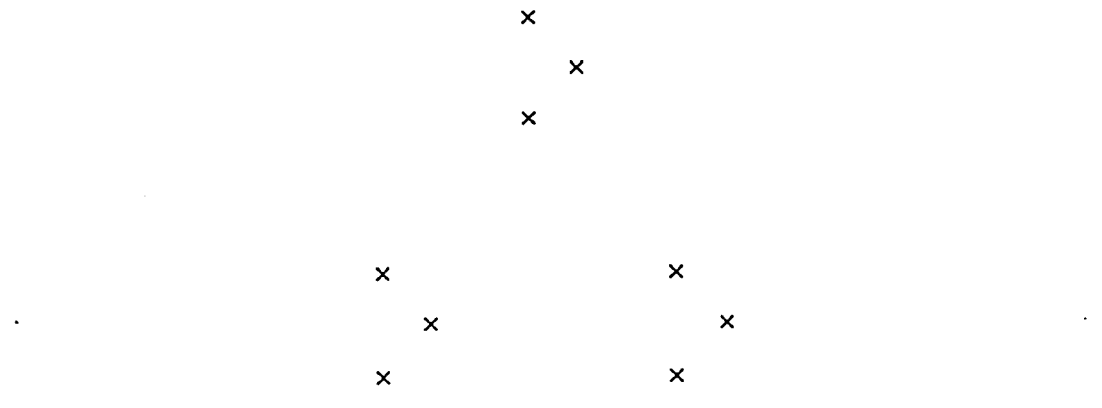


FIGURA 5.3

Repare-se que F e H apresentam extremos locais em c=3, mas os extremos globais verificam-se para c=9, como era de esperar já que para este valor de c a partição obtida é rigida. Esta é, no entanto, uma partição trivial que, em termos de análise de agrupamentos não é significativa.

O comportamento de $\hat{H}$ é diferente, contrariando a tendência referida para os outros dois. O mínimo global de $\hat{H}$ acontece para c=3, o que condiz com o resultado de uma observação directa dos dados.

c	F	H	$\hat{H}$
2	0.749	0.397	0.511
3	0.905	0.227	0.340
4	0.835	0.351	0.632
5	0.793	0.419	0.944
6	0.851	0.328	0.983
7	0.845	0.312	1.405
8	0.921	0.162	1.461
9	1	0	indefinido

TABELA 5.7

5.6. Coeficiente de Separação

Os indicadores de validade, até agora mencionados, apresentam a desvantagem de uma falta de ligação directa aos próprios dados, uma vez que todos eles são calculados com base nos elementos de uma matriz $U \in M_{fc}$, encontrada por um algoritmo de partição e não com base nos elementos de X.

Gunderson [16] estabeleceu uma estratégia de validação das partições, baseada na configuração geométrica dos agrupamentos, que optimiza o chamado **coeficiente de separação.**

Então, seja $U \in M_c$ uma partição rígida de **X**, $|X|=n$. Seja $v=\{v_i\}$ o conjunto dos centros, não necessariamente centróides, dos vários agrupamentos. Para $1 \leq i \neq j \leq c$ defina-se

$$r_i = \max_{1 \leq k \leq n} \left\{ u_{ik}\, d(x_k, v_i) \right\}$$

$$c_{ij} = d(v_i, v_j)$$

O coeficiente de separação G pode ser definido como

$$G(U,v,c,X,d) = 1 - \max_{i+1 \leq j \leq c} \left\{ \max_{1 \leq i \leq c-1} \left\{ \frac{r_i + r_j}{c_{ij}} \right\} \right\}$$

Repare-se que, na lista de parâmetros ligados a G, aparece o próprio conjunto **X**, ao contrário do que acontece para F e H.

G varia no intervalo $]-\infty,1[$ isto é, à medida que $\frac{r_i+r_j}{c_{ij}}$ se aproxima de zero, ou seja, $r_i+r_j<<c_{ij}$, os agrupamentos identificados são cada vez mais compactos e mais afastados. O par óptimo de partição rígida será (U^*,v^*) tal que

$$\max_{2 \leq c < n} \left\{ \max_{\Omega_{cd}} \{ G(U,c,v,X,d) \} \right\}$$

onde Ω_{cd} representa o conjunto dos pares (U,v) candidatos para valores de c e d fixos.

A utilização de G como um indicador de validade difere das estratégias utilizadas para F e H, na própria concepção, uma vez que a procura do óptimo se encontra ligada a uma propriedade geométrica dos agrupamentos.

Note-se, também, que a estratégia de validação corresponde a maximizar a separação dos agrupamentos, o que se encontra em aparente discordância com o conceito de partições imprecisas. Daí o facto de G ser utilizado, como indicador de validade, apenas para partições rígidas. No entanto, como já foi referido, qualquer $U \in M_{fc}$ pode ser transformada numa matriz $U_m \in M_c$ por meio de uma regra qualquer de conversão, nomeadamente a regra da máxima pertença.

5.7 Índices de Separação e o FCM

Um par de índices, semelhantes ao proposto por Gunderson,foi sugerido por Dunn [17] para estudos de validação das partições rígidas. Estes índices dependem, tal como G, de **X** e da métrica d escolhida. Contudo, não são utilizados os centros dos agrupamentos.

Seja $\mathbf{U} \in \mathbf{M_c}$ uma partição rígida de **X**. Então

$$D_1(\mathbf{U},c,\mathbf{X}.d) = \min_{i+1\le j\le c-1} \left\{ \min_{1\le i\le c} \left\{ \frac{dis(u_i,u_j)}{\max\limits_{1\le k\le c}\{diam(u_k)\}} \right\} \right\}$$

$$D_2(\mathbf{U},c,\mathbf{X},d) = \min_{i+1\le j\le c-1} \left\{ \min_{1\le i\le c} \left\{ \frac{dis(u_i,conv(u_j))}{\max\limits_{1\le k\le c}\{diam(u_k)\}} \right\} \right\}$$

em que

$$diam\ u_k = \sup_{x,y\in u_k} (d(x,y))$$

$$dis(u_i,u_j) = \inf_{\substack{x\in u_i \\ y\in u_j}} (d(x,y))$$

Estes dois índices gozam de uma relação entre eles:

$$D_1 \ge D_2 \ge D_1\text{-}1$$

Em ligação com os valores de D_1 e D_2, podem definir-se os conceitos de **agrupamentos separados** (CS) e de **agrupamentos bem separados** (CWS)

$$\mathbf{U} \text{ é uma partição CS} \Leftrightarrow D_1(\mathbf{U},c,\mathbf{X},d) > 1$$

$$\mathbf{U} \text{ é uma partição CWS} \Leftrightarrow D_2(\mathbf{U},c,\mathbf{X},d) > 1$$

Sendo assim e atendendo à relação existente entre D_1 e D_2, qualquer partição CWS é uma partição CS.

A diferença entre as indicações sugeridas por D_1 e D_2 são mais significativas quando D_1 tem um valor próximo de 1. À medida que D_2 aumenta a distinção diminui, já que entre D_1 e D_2 se verifica $D_1 \ge D_2 \ge D_1\text{-}1$.

Verifica-se que para um dado c, se existir uma partição $\mathbf{U} \in \mathbf{M_c}$ para a qual vem $D_1(\mathbf{U},c,\mathbf{X},d) > 1$, então essa partição é única. Logo, fazendo $\bar{D}_1(c) = \max\limits_{\mathbf{U}\in\mathbf{M_c}} \{D_1(\mathbf{U},c,\mathbf{X},d)\}$

pode afirmar-se que **X** contem c agrupamentos CS $\Leftrightarrow \bar{D}_1(c) > 1$.

Já que $D_2 > 1 \Rightarrow D_1 > 1$, então, quando muito, existe uma partição-c de **X** do tipo CWS.

Definindo $\bar{D}_2(c) = \max_{U \in M_c} \{D_2(U,c,X,d)\}$ vem que

X contem c agrupamentos CWS $\Leftrightarrow$ $\bar{D}_2(c) > 1$.

Ao contrário do que acontecia com os indicadores F, H e P, D_1 e D_2 permitem fazer uma afirmação directa sobre a existência e unicidade de estruturas nos dados.

O grande inconveniente na utilização de D_1 como indicador de validade verifica-se em termos computacionais: à medida que c e n aumentam torna-se impensável o cálculo de D_1 e respectivo máximo, mesmo para apenas alguns $U \in M_c$. É , então, necessário gerar, através de um algoritmo, matrizes de partição que maximizem D_1.

Dunn [ref] supôs que, se $\bar{D}_1(c)$ se tornar suficientemente grande, as funções de pertença de uma partição-c imprecisa óptima, gerada pelo FCM, devem aproximar-se das funções características definidas por $U \in M_c$ correspondente a $\bar{D}_1(c)$, baseando-se nos seguintes resultados:

— Considere-se $U \in M_c$ uma partição rígida de X, $|X|=n$ e seja $D_1 = D_1(U,c,X,d)$, em que $2 \leq c < n$, $m \in]1, +\infty[$ fixo e $U^* \in M_{fc}$ uma partição óptima para J_m. Então,

$$\sum_{i=1}^{c} \left\{ \min_{1 \leq j \leq c} \left\{ \sum_{x_k \notin u_j} \left(u_{ik}^* \right)^m \right\} \right\} \leq \frac{2n}{D_1^2}$$

— Seja $\hat{U} \in M_c$ a partição correspondente ao máximo de $D_1(U,c,X,d)$ tal que $D_1(\hat{U},c,X,d) = \bar{D}_1(c)$ e seja $U^* \in M_{fc}$ uma solução óptima de J_m. Então, para $1 \leq i \leq c$

$$0 \leq \min_{1 \leq j \leq c} \left(\sum_{x_k \notin \hat{u}_j} \left(u_{ik}^* \right)^m \right) \leq \frac{2n}{\left[\bar{D}_1(c)\right]^2}$$

Sendo σ o agrupamento em que

$$\sum_{x_k \notin \hat{u}_\sigma} \left(u_{ik}^* \right)^m = \min_{1 \leq j \leq c} \left\{ \sum_{x_k \notin \hat{u}_\sigma} \left(u_{ik}^* \right)^m \right\}$$

então

$$x_k \notin \hat{u}_\sigma \Rightarrow 0 \leq u_{ik}^* \leq \left(\frac{2n}{\left[\bar{D}_1(c)\right]^2} \right)^{1/m}$$

À medida que $\bar{D}_1$ aumenta, os valores de pertença em $U^* \in M_{fc}$ dos pontos de $X-\hat{u}_\sigma$ são muito pequenos e, contrariamente, para pontos pertencentes a $\hat{u}_\sigma$ estes valores aproximam-se de 1, já que $\sum_{i=1}^{c} u_{ik}^* = 1$. Como resultado final, tem-se:

Para $0 < \delta < 1$ e para $1 \leq i \leq c$, sendo $\hat{U} \in M_c$ correspondente ao máximo de $D_1(U,c,X,d)$, $U^* \in M_{fc}$ uma solução óptima de J_m, σ o agrupamento rígido correspondente ao mínimo do somatório dos valores de pertença dos pontos exteriores a ele, vem que

$$x_k \notin \hat{u}_\sigma \Rightarrow 0 \leq u_{ik}^* \leq \delta/c$$

$$x_k \in \hat{u}_\sigma \Rightarrow 1 - \frac{(c-1)\delta}{c} \leq u_{ik}^* \leq 1$$

A consequência imediata deste resultado é a seguinte: se $\bar{D}_1 > 1$, X contem agrupamentos CS únicos e se $\bar{D}_1 > 2$, X contem agrupamentos CWS únicos, como já foi referido. À medida que $\bar{D}_1$ aumenta, as soluções de J_m tornam-se boas aproximações destas estruturas, sendo o FCM um bom processo para as identificar.

Então, como proceder? Procura-se $U^* \in M_{fc}$ utilizando o FCM, converte-se $U^* \in M_{fc}$ em $U_m \in M_c$ e calcula-se o valor de $D_1(U_m)$. Se o conjunto Ω_{cd} representar os candidatos $U \in M_c$, para uma métrica fixa d, o problema a resolver consiste em

$$\max_{2 \leq c < n} \left\{ \max_{\Omega_{cd}} \left\{ D_1(U,c,X,d) \right\} \right\}$$

Se $U_m^* \in M_c$ for a solução deste problema e se o valor do indicador D_1 for superior a 1 os agrupamentos CS foram identificados; se D_1 for maior que 2 encontraram-se os agrupamentos CWS.

Convem realçar, mais uma vez, que com D_1 foi possível estabelecer uma relação directa entre os valores do indicador e as estruturas contidas no conjunto de dados X.

Se o FCM gerou ou não uma boa aproximação às estruturas CS ou CWS é comprovado pelo valor de D_1. No caso de haver valores de c para os quais este indicador é inferior a 1, não quer dizer que estas estruturas não existam em X, mas sim que, para esses valores de c, as partições rígidas deduzidas a partir do FCM não são destes tipos. Note-se que podem existir agrupamentos CS e CWS para mais do que um valor de c.

Exemplo 5.5

Analise-se o conjunto referido no exemplo 5.4, com base nos valores de pertença obtidos por aplicação do já referido FCM. A regra de conversão utilizada para a transformação de $U^* \in M_{fc}$ em $U_m \in M_c$ foi a regra da máxima pertença.

Na tabela 5.8 encontram-se representados os agrupamentos resultantes desta conversão.

	c = 2	c = 3	c = 4
x1	2	2	2
x2	2	2	2
x3	2	2	3
x4	1	3	4
x5	1	3	4
x6	1	3	4
x7	1	1	1
x8	1	1	1
x9	1	1	1

TABELA 5.8

Para os indicadores os resultados foram os seguintes

c	F	H	$\hat{H}$	$\bar{D}_1(U_m)$
2	0.749	0.397	0.511	0.728
3	0.905	0.227	0.340	2.121
4	0.835	0.351	0.632	0.707

TABELA 5.9

Note-se que o indicador $\bar{D}_1$ optimiza para c=3, tal como os outros, e apresenta um valor que identifica os agrupamentos obtidos pelo FCM, para c=3, como sendo do tipo CWS (bem separados).

Outros conjuntos, obtidos a partir deste por aproximação dos pontos, foram analisados. O valor de $\bar{D}_1$ foi diminuindo até que passou a identificar como óptima a partição obtida para c=2, mas esta do tipo CS, como se pode verificar na tabela 5.10.

c	F	H	$\hat{H}$	$\bar{D}_1(U_m)$
2	0.923	0.156	0.201	1.217
3	0.846	0.304	0.456	0.500
4	0.810	0.401	0.722	0.707

TABELA 5.10

5.8. Novo Indicador

É exposta, de seguida,uma nova forma de abordar o problema de validação das partições,

baseada no conceito de pontos rejeitados a um nível α de um conjunto **X**.

Seja uma partição-c imprecisa de **X**. Para cada agrupamento i obtido, define-se núcleo de nível α da seguinte forma

$$C(u_i,\alpha) = \left\{ x \in X \mid u_i(x) \geq \alpha \right\}$$

Então, entenda-se como pontos rejeitados a um nível α aqueles cuja função de pertença apresenta um valor inferior a α em todos os agrupamentos formados, ou seja

$$R(U,\alpha) = \left\{ x \in X \mid u_i(x) < \alpha \ \forall i \right\}$$

Para um dado valor de c, quanto "mais fortes" forem os núcleos-α de X, isto é, para cada coluna da matriz **U** quanto maiores forem os elementos máximos e menores todos os outros, "melhor" é a partição obtida. E, também, menor será o número de pontos rejeitados.

Atenda-se à figura 5.4, onde se representa, como se fosse contínua, a evolução do número de pontos rejeitados para uma partição-c imprecisa de um conjunto **X**, $|X|=n$, quando a este foram aplicados dois algoritmos distintos. As coordenadas em x correspondem aos vários níveis da função de pertença e as coordenadas em y correspondem ao número de pontos rejeitados.

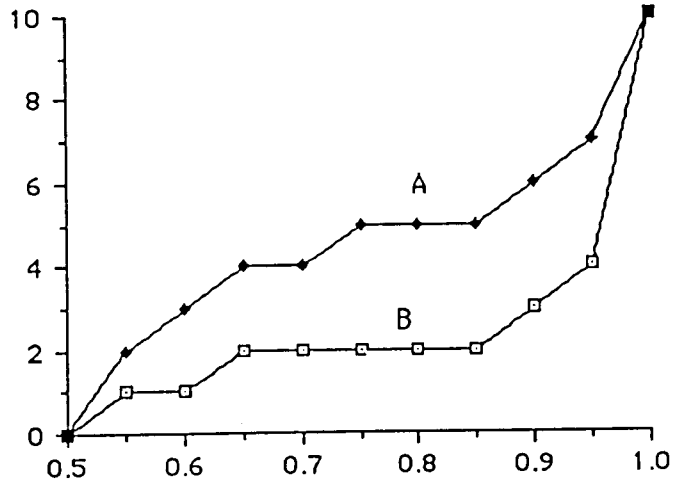


FIGURA 5.4

De acordo com o que acaba de ser exposto a partição B é sugerida como sendo "melhor" do que a partição A.

Podem surgir, no entanto, situações como a representada na figura 5.5. Qual a melhor partição? Para α entre 0 e 0.75 a partição B corresponde a menor número de rejeitados que A; entre 0.75 e 0.90 A conduz a melhores resultados, mas para α entre 0.90 e 1 torna a ser B a "melhor" partição.

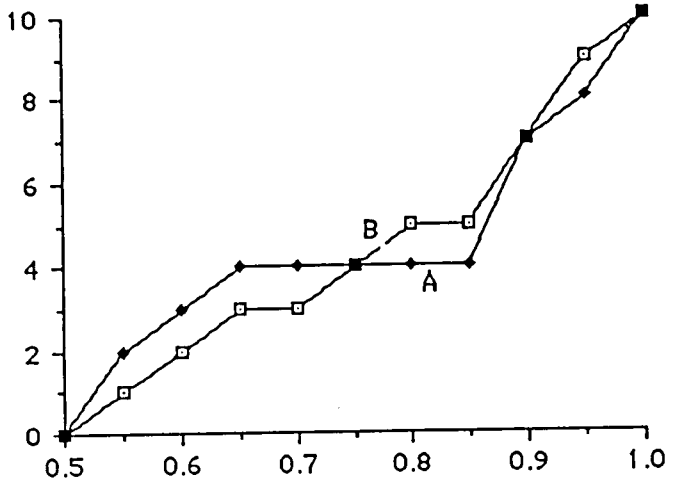


FIGURA 5.5

Então, defina-se um novo indicador I da seguinte forma

$$I(c) = \frac{\sum_{\alpha=1/c}^{1} \alpha\, f(\alpha)}{\sum_{\alpha=1/c}^{1} \alpha}$$

onde α representa os vários níveis da função de pertença e $f(\alpha)$ o número de pontos rejeitados para cada um desses níveis. O indicador é uma soma pesada do número de pontos rejeitados no sentido de representar a importância dos pontos com funções de pertença elevadas.

Como à medida que c aumenta o limite inferior do somatório diminui, I definido desta forma não permitiria uma avaliação conveniente da qualidade das partições obtidas para diferentes valores de c. Neste sentido fez-se uma normalização, quer em termos de α quer em termos de $f(\alpha)$, que conduziu à expressão

$$\hat{I}(y) = \frac{\sum_{y=0}^{1} y\ f(y)}{\sum_{y=0}^{1} y}$$

onde $y = \frac{\alpha c - 1}{c - 1}$ e $f(y) = \frac{f(\alpha)}{n}$

Tal como os indicadores F e H os valores extremos de $\hat{I}$ verificam-se para partições rígidas, para as quais este índice tem valor zero, e para $U = \left[1/c\right]$ onde toma valor 1 qualquer que seja c.

Então, a estratégia a seguir, no sentido de a "melhor" partição ser a que mais se aproxima da partição rígida, será

$$\min_{c} \left\{ \min_{\Omega_c} \{ I(U) \} \right\}$$

com Ω_c a representar o conjunto finito dos $U \in M_{fc}$ óptimos.

Repare-se que, ao contrário do que se passa com outros indicadores, o valor de I não depende de c. Assim, este indicador permite a comparação da qualidade das partições obtidas para conjuntos diferentes.

Exemplo 5.6

Considere-se o conjunto X representado na figura 5.3. Foi processado com o FCM, tomando m=2, ε=0.001 e norma Euclideana. Os resultados obtidos para o indicador $\hat{I}$, quando c varia de 2 a 5, encontram-se na tabela 5.11, assim como os valores de F, H e $\hat{H}$.

c	F	H	$\hat{H}$	$\hat{I}$
2	0.749	0.397	0.511	0.530
3	0.905	0.227	0.340	0.175
4	0.835	0.351	0.632	0.259
5	0.793	0.419	0.944	0.315

TABELA 5.11

Tal como os outos indicadores, $\hat{I}$ identifica c=3 como sendo o que corresponde à melhor partição de X. Na figura 5.6 estão representados os gráficos deste índice para os diferentes valores de c.

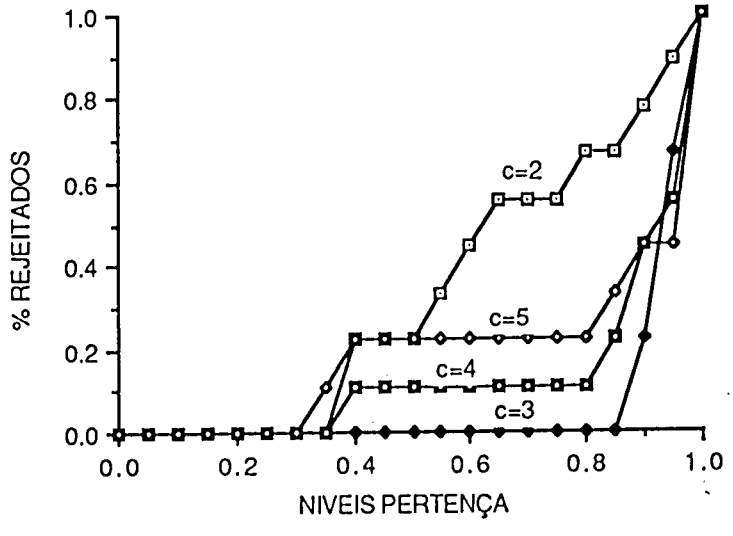


FIGURA 5.6

A partir deste conjunto formaram-se outros, aproximando os três grupos, que se encontram representados nas figuras 5.7, 5.8 e 5.9. Na tabela 5.12 apresentam-se os valores encontrados para vários indicadores tomando diferentes c's.

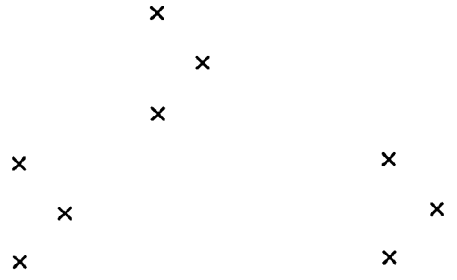


FIGURA 5.7

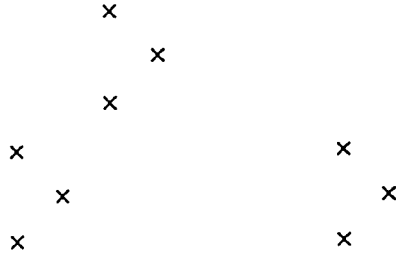


FIGURA 5.8

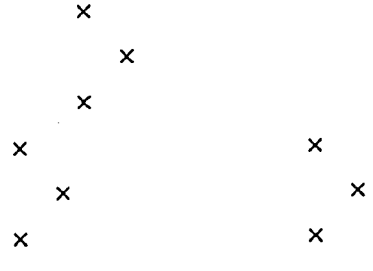


FIGURA 5.9

Como seria de prever, a dificuldade em identificar c=3 como a partição óptima é cada vez maior; a diferença entre $\hat{I}(3)$ e $\hat{I}(2)$ vai diminuindo até que no conjunto da figura 5.9 o valor óptimo identificado para c passa a ser 2.

	c	F	H	$\hat{H}$	$\hat{I}$
Conjunto 5.7	2	0.860	0.250	0.322	0.328
	3	0.892	0.245	0.367	0.194
	4	0.822	0.371	0.668	0.279
Conjunto 5.8	2	0.878	0.227	0.292	0.289
	3	0.867	0.286	0.429	0.239
	4	0.793	0.421	0.757	0.315
Conjunto 5.9	2	0.896	0.200	0.257	0.246
	3	0.859	0.294	0.441	0.264
	4	0.782	0.436	0.785	0.350

TABELA 5.12

No caso do conjunto representado na figura 8, o indicador $\hat{I}$ ao contrário dos outros, aponta para c=3 como sendo c óptimo. Se se atender, neste caso, às partições obtidas para c=2 e c=3, correspondentes à figura 5.10, verifica-se que a distribuição dos pontos em torno dos centróides é mais homogénea para c=3 do que para c=2, propriedade que se encontra ligada à própria concepção de $\hat{I}$.

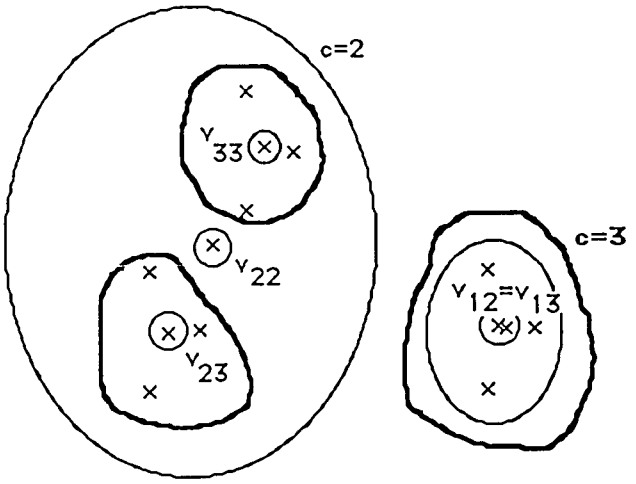


FIGURA 5.10

O indicador $\hat{I}$ goza, todavia da mesma fragilidade dos indicadores anteriormente analisados, pois resulta dos valores de pertença atribuidos a cada ponto por um algoritmo, em vez de resultar dos próprios pontos.

Assim, se para o mesmo conjunto de pontos fosse utilizado o algoritmo FCM, com dois valores do parâmetro m diferente, obter-se-iam duas distribuições de valores de pertença diferentes (é sabido que valores de m menores influenciam o FCM de maneira a este atibuir valores de pertença de forma mais "rígida"). Em conformidade, obteríamos

dois valores de $\hat{I}$ para o mesmo conjunto de dados!
Este, porém, é um mal comum à generalidade dos indicadores que se baseiam nos valores de pertença, e não uma característica exclusiva de $\hat{I}$. Esta apreciação serve, no entanto, para realçar uma vez mais o cuidado que sempre deve presidir à análise e validação dos resultados de um processo de agrupamento.

6. Análise de Resultados

6.1. Introdução

Neste capítulo são apresentados vários conjuntos processados por algoritmos de partição imprecisa e analisam-se os resultados assim obtidos, procurando-se, por um lado, salientar as vantagens que aqueles apresentam relativamente aos de partição rígida no que diz respeito à interpretação das características dos dados e, por outro, fazer um estudo comparativo de alguns desses mesmos processos de obtenção de agrupamentos.

6.2. Conjunto 1

Os métodos rígidos de análise de agrupamentos geram partições tais que cada objecto pertence a um e um só grupo. Contudo, existem situações em que os objectos não podem ser atribuidos exclusivamente a um agrupamento. É o caso de pontos que ocupem uma posição de simetria no conjunto, isto é, "entre" os agrupamentos. Nestas circunstâncias, os métodos imprecisos representam de uma forma mais adequada as estruturas contidas nos dados.

Para ilustrar a diferença entre os resultados obtidos por métodos clássicos e métodos imprecisos de agrupamento, considere-se o conjunto X representado na figura 6.1.[8]

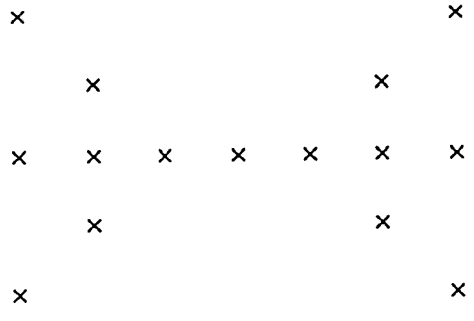


FIGURA 6.1

O conjunto foi processado com o HCM e o FCM já referidos no capítulo 4, tomando c=2. Utilizou-se norma euclideana, ε=0.001 e m=2. Os resultados obtidos conduziram às partições representadas nas figuras 6.2 e 6.3, nesta última por transformação de U^* em U_m, onde os elementos junto dos pontos correspondem aos valores de pertença ao agrupamento 1.

Repare-se que o ponto x_8 é simétrico relativamente aos dois agrupamentos. No caso da partição rígida essa característica dos dados não é traduzida nos resultados, uma vez que o ponto pertence inteiramente ao grupo 1. Pelo contrário, usando o algoritmo de partição imprecisa, este ponto apresenta um valor de pertença igual para os dois grupos, de acordo com a sua posição no conjunto dado.

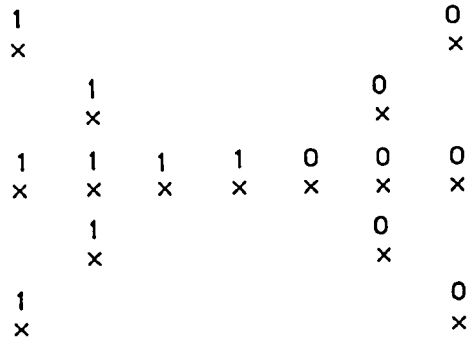


FIGURA 6.2

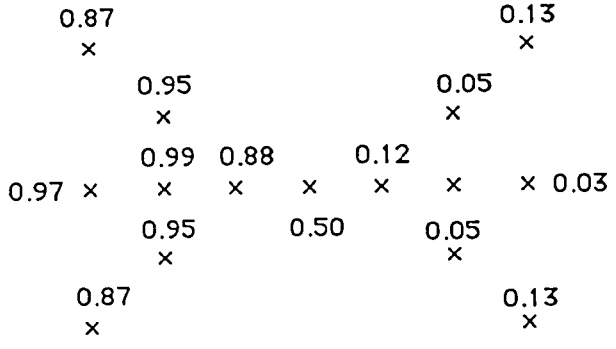


FIGURA 6.3

No caso da partição rígida, os pontos de um grupo têm o mesmo valor de pertença a esse mesmo grupo, como se todos ocupassem posições idênticas relativamente ao centróide. O mesmo não se passa quando se aplica o FCM. Aos pontos "mais ligados" aos centróides, caso dos pontos x_2 e x_5, correspondem valores de pertença maiores.

6.3. Conjunto 2

Um conjunto de dados X pode conter grupos de pontos de formas geométricas diferentes. No Fuzzy c-Means prevê-se a possibilidade de utilizar uma norma que não a norma euclideana, mas essa é fixa para todos os c agrupamentos a identificar, não conduzindo a resultados satisfatórios nestes casos. O algoritmo de Gustafson e Kessel permite uma adaptação da norma, através de uma variação local da mesma, de maneira a identificar alterações na forma dos agrupamentos.

Na figura 6.4 está representado um conjunto X contendo vinte pontos distribuidos no espaço em forma de cruz[11]. Os agrupamentos, que visualmente se identificam, apresentam estruturas distintas, estruturas essas responsáveis pelos maus resultados a que conduz o FCM, mesmo no caso de se utilizar norma diagonal.

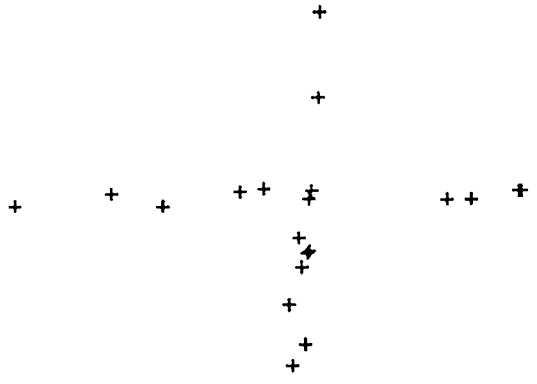


FIGURA 6.4

Na tabela 6.1 encontram-se os valores de pertença dos pontos ao primeiro agrupamento, obtidos quando se aplica o algoritmo de Gustafson e Kessel (coluna 3) e o FCM utilizando norma diagonal (coluna 4) e norma euclideana (coluna 5), tomando em ambos os casos c=m=2.

DADOS		GUSTAFSON e KESSEL	FCM com N_E	FCM com N_D
x_1	(-9.75, -0.15)	0.998	0.752	0.494
x_2	(-6.44, 0.34)	0.997	0.818	0.446
x_3	(-4.69, -0.30)	0.994	0.885	0.463
x_4	(-2.04, 0.37)	0.985	0.916	0.236
x_5	(-1.24, 0.45)	0.951	0.891	0.145
x_6	(0.33, -0.08)	0.236	0.773	0.150
x_7	(5.04, -0.21)	0.995	0.016	0.391
x_8	(5.86, -0.25)	0.996	0.033	0.414
x_9	(7.54, 0.16)	0.998	0.085	0.419
x_{10}	(7.67, 0.24)	0.998	0.089	0.417
x_{11}	(-0.30, -8.07)	0.001	0.700	0.780
x_{12}	(0.13, -7.13)	0.003	0.709	0.813
x_{13}	(-0.37, -5.18)	0.006	0.809	0.906
x_{14}	(0.03. -3.33)	0.001	0.875	0.998
x_{15}	(0.35, -2.63)	0.008	0.866	0.977
x_{16}	(0.23, -2.68)	0.002	0.880	0.984
x_{17}	(-0.05, -2.00)	0.013	0.926	0.890
x_{18}	(0.41, 0.37)	0.169	0.709	0.054
x_{19}	(0.69, 4.75)	0.004	0.426	0.168
x_{20}	(0.74, 8.87)	0.003	0.424	0.289

TABELA 6.1

Como era de prever, a alteração da norma no algoritmo de Bezdek de N_E para N_D, embora conduzindo a resultados mais próximos da estrutura dos dados, não permitiu a identificação dos grupos que visualmente se distinguem em X. Em contrapartida, do algoritmo de Gustafson e Kessel resulta uma partição tal que as funções de pertença aos

dois grupos são bem distintas, como é sugerido pelos próprios dados.

Apresentam-se, em seguida as matrizes de covariância imprecisa dos dois agrupamentos obtidos, C_{f1} e C_{f2}. Estas matrizes confirmam que uma escolha global da norma para os dois grupos não conduziria a resultados satisfatórios. Mais, a composição de ambas sugere que o algoritmo detectou o tipo de estruturas contidas nos dados, ao contrário do que se passou com o FCM.

$$C_{f1}=\begin{bmatrix} 37.817 & -0.019 \\ -0.019 & 0.078 \end{bmatrix} \qquad C_{f2}=\begin{bmatrix} 0.124 & 1.464 \\ 1.464 & 23.994 \end{bmatrix}$$

6.4. Conjunto 3

Podem surgir conjuntos em que as estruturas neles contidas tenham características particulares. É o caso do exemplo descrito na secção anterior, em que os pontos se distribuem ao longo dos eixos coordenados, aproximando-se de duas rectas. Com base nos resultados obtidos para as matrizes C_{f1} e C_{f2} finais correspondentes a este conjunto, fez-se uma adaptação ao algoritmo de Gustafson e Kessel, impondo que as matrizes A_i^* fossem matrizes diagonais. A matriz de partição encontrada apresenta-se na tabela 6.2.

DADOS		u_{1k}	u_{2k}
x_1	(-9.75, -0.15)	0.998	0.002
x_2	(-6.44, 0.34)	0.997	0.003
x_3	(-4.69, -0.30)	0.994	0.006
x_4	(-2.04, 0.37)	0.985	0.015
x_5	(-1.24, 0.45)	0.951	0.049
x_6	(0.33, -0.08)	0.236	0.757
x_7	(5.04, -0.21)	0.995	0.004
x_8	(5.86, -0.25)	0.996	0.004
x_9	(7.54, 0.16)	0.998	0.002
x_{10}	(7.67, 0.24)	0.998	0.002
x_{11}	(-0.30, -8.07)	0.001	0.999
x_{12}	(0.13, -7.13)	0.003	0.997
x_{13}	(-0.37, -5.18)	0.006	0.994
x_{14}	(0.03. -3.33)	0.001	0.999
x_{15}	(0.35, -2.63)	0.008	0.992
x_{16}	(0.23, -2.68)	0.002	0.998
x_{17}	(-0.05, -2.00)	0.013	0.987
x_{18}	(0.41, 0.37)	0.169	0.832
x_{19}	(0.69, 4.75)	0.004	0.996
x_{20}	(0.74, 8.87)	0.003	0.997

TABELA 6.2

Repare-se que em termos da matriz U_m derivada da matriz U^* pela regra da máxima pertença, a partição agora obtida é igual à que se obteve na secção 6.3. No entanto, desta

vez o cálculo da matriz S_{fi}^{-1} foi simplificado já que se transformou S_{fi} numa matriz diagonal. A inversão de uma matriz cheia pode levantar problemas de origem numérica.

Formou-se um outro conjunto, efectuando sobre a cruz da figura 6.4 uma rotação de 45°. A este conjunto foi aplicado o algoritmo de Gustafson e Kessel, com a alteração referida anteriormente. A partição U_m, correspondente aos resultados obtidos quando se toma c=m=2, (tabela 6.3), encontra-se representada na figura 6.5.

DADOS	u_{1k}	u_{2k}
x_1	0.844	0.156
x_2	0.903	0.097
x_3	0.947	0.053
x_4	0.920	0.080
x_5	0.869	0.131
x_6	0.759	0.241
x_7	0.054	0.946
x_8	0.035	0.965
x_9	0.051	0.949
x_{10}	0.055	0.945
x_{11}	0.888	0.112
x_{12}	0.906	0.094
x_{13}	0.955	0.045
x_{14}	0.876	0.024
x_{15}	0.957	0.043
x_{16}	0.964	0.036
x_{17}	0.961	0.039
x_{18}	0.678	0.322
x_{19}	0.130	0.870
x_{20}	0.185	0.815

TABELA 6.3

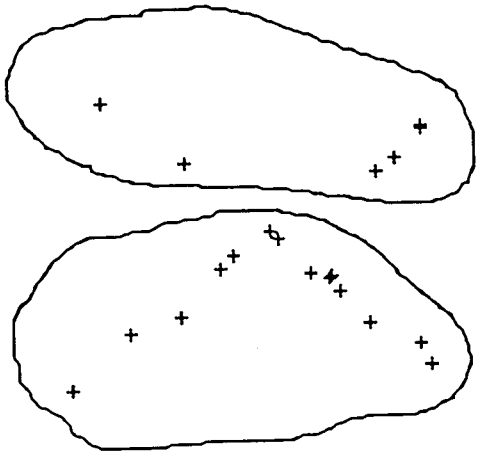


FIGURA 6.5.

Como se pode verificar, os dois agrupamentos que visualmente se distinguem não são identificados por este processo. Note-se que os pontos deixaram de estar distribuidos ao longo dos eixos. Prevê-se que um algoritmo que encontrasse direcções previlegiadas para se definirem os agrupamentos, conduzisse a resultados correspondentes a uma partição adequada do conjunto X.

6.5. Conjunto 4

Como se sabe, é considerado, em geral, desejável em estudos de análise de Sistemas Eléctricos de Energia fazer a classificação dos diagramas de carga desse mesmo sistema e obter diagramas-tipo de cada classe de maneira a poder facilmente "etiquetar" um novo diagrama.

O que se tem feito até agora consiste na obtenção de partições rígidas do conjunto dos diagramas, o que não satisfaz quando surgem curvas que não podem ser atribuidas claramente a um dos grupos já identificados. Sendo assim, pensou-se em aplicar nesta área técnicas baseadas em conceitos de conjuntos imprecisos. Como não foi possível reunir, em tempo útil, um conjunto de diagramas adquiridos a partir de uma amostra significativa de consumidores de energia eléctrica, optou-se por apresentar, um estudo efectuado sobre um conjunto de diagramas de afluências de água a uma barragem, estudo esse que se supõe semelhante ao que se faria para diagramas de energia eléctrica. Este exemplo tem objectivo meramente ilustrativo e não visa a obtenção de conclusões específicas.

Tomou-se um conjunto de diagramas deste tipo, representando as afluências à barragem da Aguieira, sendo cada diagrama definido por doze pontos, correspondentes às afluências nos doze meses do ano. As tabelas, que se seguem, contêm as coordenadas dos trinta e seis pontos que constituem o conjunto de dados.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
OUT	33.5	28.3	14.2	26.1	19.0	34.2	39.5	74.2
NOV	58.9	32.1	36.1	79.4	49.7	445.6	63.3	43.0
DEZ	126.7	116.2	138.0	114.1	106.8	176.0	244.2	149.4
JAN	184.6	689.3	183.2	80.7	245.4	160.9	167.6	53.2
FEV	707.2	312.6	70.6	228.5	531.5	143.3	104.4	103.8
MAR	698.2	173.6	56.1	195.2	784.1	155.6	97.5	319.3
ABR	430.9	140.8	41.5	96.4	200.8	250.1	94.4	100.1
MAI	186.5	135.0	36.8	127.4	186.2	270.6	79.5	94.9
JUN	70.4	77.3	25.0	105.6	115.5	123.1	25.2	43.7
JUL	19.9	18.0	12.8	28.6	42.8	41.3	6.0	10.0
AGO	10.6	18.9	2.8	8.5	16.9	18.5	0.1	5.5
SET	16.6	0.8	6.4	0.7	21.9	26.6	0.0	1.4

	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
OUT	2.3	5.6	34.6	2.3	29.3	40.0	335.7	13.5
NOV	24.9	70.8	23.6	23.2	17.4	204.5	572.0	107.4
DEZ	58.8	440.5	34.7	56.4	262.2	705.4	409.1	326.7
JAN	538.4	348.8	51.9	137.5	225.1	312.5	364.1	580.1
FEV	486.0	159.4	193.2	164.7	104.3	648.5	161.0	137.4
MAR	249.5	604.9	165.2	204.2	262.2	625.6	79.9	311.6
ABR	98.8	368.2	89.4	199.6	166.6	257.2	98.5	188.4
MAI	52.1	176.0	67.7	72.6	111.8	120.5	75.2	105.4
JUN	36.1	81.1	31.1	49.0	49.0	52.8	62.9	58.2
JUL	8.7	24.5	3.9	31.2	9.3	8.9	18.4	17.9
AGO	5.7	23.1	0.1	12.0	2.7	5.0	2.9	6.7
SET	5.3	23.8	0.0	12.8	15.1	5.6	2.9	5.6

	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
OUT	18.8	5.4	12.8	93.1	105.4	7.6	6.8	25.6
NOV	45.3	510.9	7.2	355.0	107.9	43.2	109.8	64.8
DEZ	40.3	405.4	13.5	366.1	82.1	32.6	357.3	69.0
JAN	357.7	134.1	86.0	753.6	143.4	22.5	330.	666.6
FEV	510.3	372.6	99.4	939.9	241.0	259.8	307.3	189.8
MAR	392.8	489.1	209.2	221.2	187.8	145.4	716.3	81.6
ABR	239.8	188.2	63.8	483.0	83.6	124.5	203.9	52.9
MAI	90.8	63.2	32.9	120.6	126.5	139.4	165.2	85.3
JUN	85.3	68.3	8.4	2.8	45.7	34.7	87.0	49.6
JUL	15.8	20.6	1.4	9.8	8.1	5.3	23.9	6.6
AGO	3.6	2.6	0.7	3.0	2.9	1.4	4.8	0.9
SET	3.4	2.4	0.5	5.0	2.8	7.3	17.2	1.1

	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}	x_{32}
OUT	0.7	19.4	34.8	19.0	14.2	2.3	53.2	26.5
NOV	17.0	18.2	111.5	33.7	36.2	15.7	145.7	58.7
DEZ	27.4	21.7	225.0	50.9	34.8	27.1	194.1	329.2
JAN	178.9	90.2	358.9	361.5	80.3	55.3	505.5	230.8
FEV	98.1	497.0	142.7	420.9	152.4	81.9	631.0	614.1
MAR	125.3	251.4	91.1	123.2	326.7	54.6	235.9	435.8
ABR	194.4	92.0	64.8	80.4	101.1	57.1	132.7	129.3
MAI	167.3	78.3	182.1	83.0	56.2	40.0	82.1	171.3
JUN	191.2	48.5	80.4	101.1	38.9	23.6	59.1	70.9
JUL	68.9	26.8	37.5	58.9	6.7	12.1	15.8	11.4
AGO	21.0	2.7	8.8	13.4	0.8	7.6	7.6	1.9
SET	7.7	5.7	7.0	13.2	0.9	7.9	7.9	0.0

	x_{33}	x_{34}	x_{35}	x_{36}
OUT	13.6	57.7	3.6	17.0
NOV	43.3	47.9	37.9	4.5
DEZ	535.8	77.9	27.0	302.3
JAN	351.3	135.5	18.1	273.4
FEV	92.8	146.5	14.3	110.1
MAR	382.9	131.5	34.6	72.7
ABR	355.1	97.9	78.7	53.9
MAI	129.2	87.9	77.8	29.1
JUN	56.4	46.3	29.7	16.8
JUL	12.8	16.5	9.2	2.3
AGO	2.5	14.5	0.0	0.4
SET	0.0	6.2	0.9	1.5

6.5.1. Resultados com o Fuzzy c-Means

O conjunto descrito foi processado com o algoritmo de Bezdek, fixando m=2 e ε=0.001. Tomou-se c=2, c=3 e c=4 e utilizou-se quer norma euclideana quer norma diagonal. Os valores de pertença obtidos são os indicados nas tabelas 6.5 e 6.6.

Os indicadores de validade F, H, $\hat{H}$ e $\hat{I}$, cujos valores correspondem à tabela 6.4, sugerem a partição em dois grupos obtida como sendo a óptima.

	NE				ND			
c	F	H	$\hat{H}$	$\hat{I}$	F	H	$\hat{H}$	$\hat{I}$
2	0.699	0.465	0.492	0.618	0.594	0.592	0.627	0.822
3	0.566	0.759	0.828	0.674	0.435	0.953	1.039	0.865
4	0.449	1.026	1.155	0.765	0.349	1.209	1.361	0.876

TABELA 6.4

dados	c = 2		c = 3			c = 4			
x_1	0.242	0.758	0.145	0.206	0.649	0.555	0.121	0.116	0.208
x_2	0.481	0.519	0.147	0.730	0.123	0.112	0.263	0.136	0.489
x_3	0.926	0.074	0.833	0.119	0.048	0.036	0.324	0.571	0.069
x_4	0.944	0.056	0.898	0.067	0.035	0.027	0.153	0.777	0.043
x_5	0.263	0.737	0.160	0.226	0.614	0.519	0.136	0.129	0.216
x_6	0.661	0.339	0.461	0.323	0.216	0.138	0.384	0.299	0.179
x_7	0.927	0.073	0.816	0.128	0.056	0.036	0.476	0.425	0.063
x_8	0.882	0.118	0.813	0.111	0.076	0.056	0.232	0.643	0.069
x_9	0.393	0.607	0.117	0.757	0.126	0.078	0.117	0.077	0.728
x_{10}	0.352	0.648	0.219	0.290	0.491	0.381	0.248	0.163	0.208
x_{11}	0.933	0.067	0.917	0.055	0.028	0.015	0.068	0.893	0.024
x_{12}	0.944	0.056	0.895	0.072	0.033	0.024	0.149	0.789	0.043
x_{13}	0.855	0.145	0.661	0.228	0.111	0.054	0.584	0.282	0.080
x_{14}	0.222	0.778	0.119	0.192	0.689	0.606	0.126	0.093	0.175
x_{15}	0.546	0.454	0.342	0.394	0.264	0.178	0.356	0.229	0.237
x_{16}	0.479	0.521	0.213	0.587	0.200	0.145	0.414	0.156	0.289
x_{17}	0.255	0.745	0.169	0.504	0.327	0.195	0.135	0.112	0.558
x_{18}	0.362	0.638	0.235	0.270	0.495	0.374	0.234	0.179	0.213
x_{19}	0.917	0.083	0.879	0.080	0.041	0.027	0.126	0.803	0.044
x_{20}	0.289	0.711	0.181	0.387	0.432	0.321	0.182	0.135	0.362
x_{21}	0.943	0.057	0.849	0.107	0.044	0.033	0.232	0.671	0.064
x_{22}	0.892	0.108	0.834	0.106	0.060	0.039	0.143	0.756	0.062
x_{23}	0.253	0.747	0.146	0.221	0.633	0.537	0.161	0.116	0.186
x_{24}	0.617	0.383	0.253	0.618	0.129	0.102	0.362	0.199	0.337
x_{25}	0.883	0.117	0.751	0.173	0.076	0.053	0.299	0.552	0.096
x_{26}	0.682	0.372	0.460	0.319	0.221	0.152	0.226	0.366	0.256
x_{27}	0.820	0.180	0.486	0.410	0.104	0.036	0.727	0.151	0.086
x_{28}	0.651	0.349	0.257	0.635	0.108	0.083	0.275	0.209	0.433
x_{29}	0.882	0.118	0.813	0.116	0.071	0.048	0.172	0.712	0.068
x_{30}	0.906	0.094	0.853	0.098	0.049	0.035	0.172	0.735	0.058
x_{31}	0.217	0.783	0.113	0.635	0.252	0.111	0.086	0.053	0.750
x_{32}	0.131	0.869	0.087	0.165	0.748	0.625	0.098	0.076	0.201
x_{33}	0.224	0.776	0.140	0.255	0.605	0.493	0.138	0.108	0.261
x_{34}	0.979	0.021	0.977	0.016	0.007	0.007	0.059	0.921	0.013
x_{35}	0.884	0.116	0.816	0.120	0.064	0.046	0.209	0.673	0.072
x_{36}	0.839	0.161	0.613	0.281	0.106	0.055	0.554	0.285	0.106

TABELA 6.5

dados	c = 2		c = 3			c = 4			
x_1	0.299	0.701	0.507	0.326	0.167	0.476	0.208	0.107	0.209
x_2	0.429	0.571	0.331	0.445	0.224	0.190	0.329	0.151	0.330
x_3	0.853	0.147	0.079	0.129	0.792	0.045	0.114	0.726	0.115
x_4	0.569	0.431	0.283	0.400	0.317	0.146	0.323	0.205	0.326
x_5	0.296	0.704	0.524	0.316	0.160	0.526	0.189	0.096	0.189
x_6	0.363	0.637	0.454	0.338	0.208	0.385	0.236	0.143	0.236
x_7	0.858	0.142	0.075	0.130	0.795	0.043	0.119	0.718	0.120
x_8	0.838	0.162	0.114	0.184	0.702	0.073	0.189	0.548	0.190
x_9	0.664	0.336	0.207	0.380	0.413	0.110	0.307	0.277	0.306
x_{10}	0.305	0.695	0.510	0.320	0.170	0.496	0.199	0.106	0.199
x_{11}	0.866	0.134	0.059	0.095	0.846	0.029	0.072	0.826	0.073
x_{12}	0.582	0.418	0.297	0.358	0.345	0.180	0.292	0.235	0.293
x_{13}	0.628	0.372	0.262	0.353	0.385	0.150	0.295	0.260	0.295
x_{14}	0.358	0.642	0.408	0.383	0.209	0.286	0.281	0.152	0.281
x_{15}	0.506	0.494	0.319	0.356	0.325	0.213	0.274	0.238	0.275
x_{16}	0.415	0.585	0.296	0.511	0.193	0.139	0.369	0.124	0.368
x_{17}	0.452	0.548	0.294	0.489	0.217	0.143	0.360	0.139	0.358
x_{18}	0.451	0.549	0.347	0.381	0.272	0.219	0.292	0.198	0.291
x_{19}	0.818	0.182	0.093	0.140	0.767	0.051	0.109	0.730	0.110
x_{20}	0.384	0.616	0.380	0.395	0.225	0.260	0.289	0.162	0.289
x_{21}	0.793	0.207	0.150	0.257	0.593	0.086	0.247	0.419	0.248
x_{22}	0.789	0.211	0.144	0.211	0.645	0.091	0.198	0.513	0.198
x_{23}	0.245	0.755	0.560	0.311	0.129	0.530	0.194	0.082	0.194
x_{24}	0.684	0.316	0.196	0.343	0.461	0.107	0.283	0.326	0.284
x_{25}	0.412	0.588	0.405	0.354	0.241	0.311	0.259	0.171	0.259
x_{26}	0.745	0.255	0.183	0.299	0.518	0.103	0.268	0.360	0.269
x_{27}	0.432	0.568	0.348	0.433	0.219	0.187	0.333	0.144	0.336
x_{28}	0.423	0.577	0.376	0.393	0.231	0.247	0.296	0.161	0.296
x_{29}	0.854	0.146	0.077	0.122	0.801	0.044	0.106	0.743	0.107
x_{30}	0.815	0.185	0.111	0.164	0.725	0.069	0.146	0.639	0.146
x_{31}	0.423	0.577	0.261	0.561	0.178	0.112	0.392	0.107	0.389
x_{32}	0.427	0.573	0.332	0.436	0.232	0.182	0.331	0.158	0.329
x_{33}	0.384	0.616	0.376	0.400	0.224	0.246	0.296	0.162	0.296
x_{34}	0.750	0.250	0.189	0.298	0.513	0.108	0.271	0.347	0.274
x_{35}	0.825	0.175	0.091	0.137	0.772	0.051	0.111	0.728	0.110
x_{36}	0.789	0.211	0.123	0.198	0.679	0.071	0.170	0.588	0.171

TABELA 6.6

Os agrupamentos derivados, pela regra da máxima pertença, dos resultados das tabelas 6.5 e 6.6 para c=2 representam-se nas figuras 6.6, 6.7, 6.8 e 6.9.

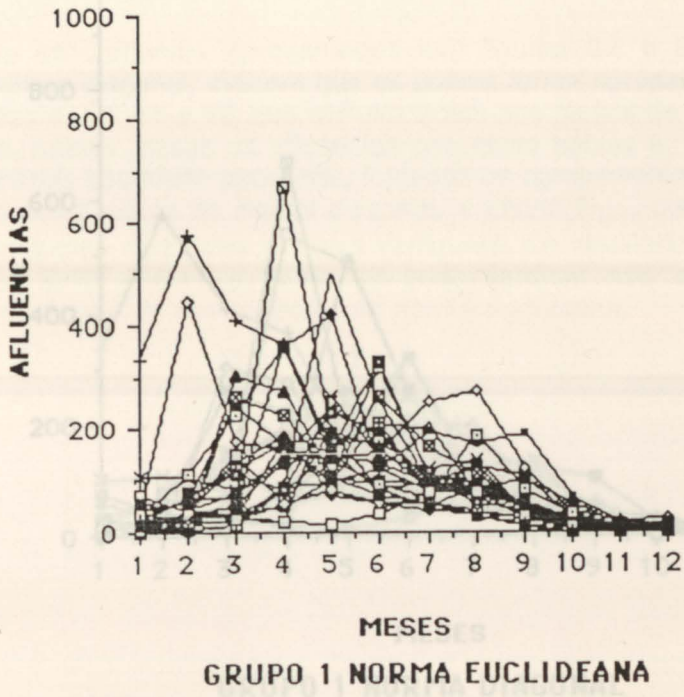


FIGURA 6.6

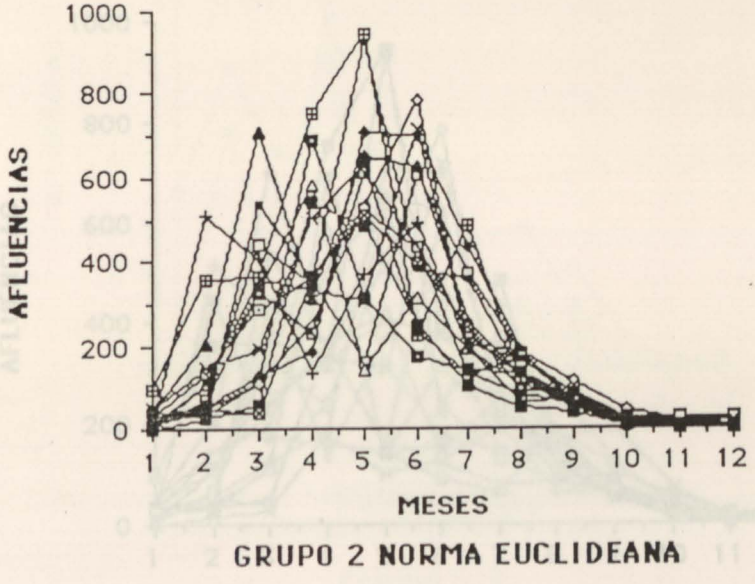
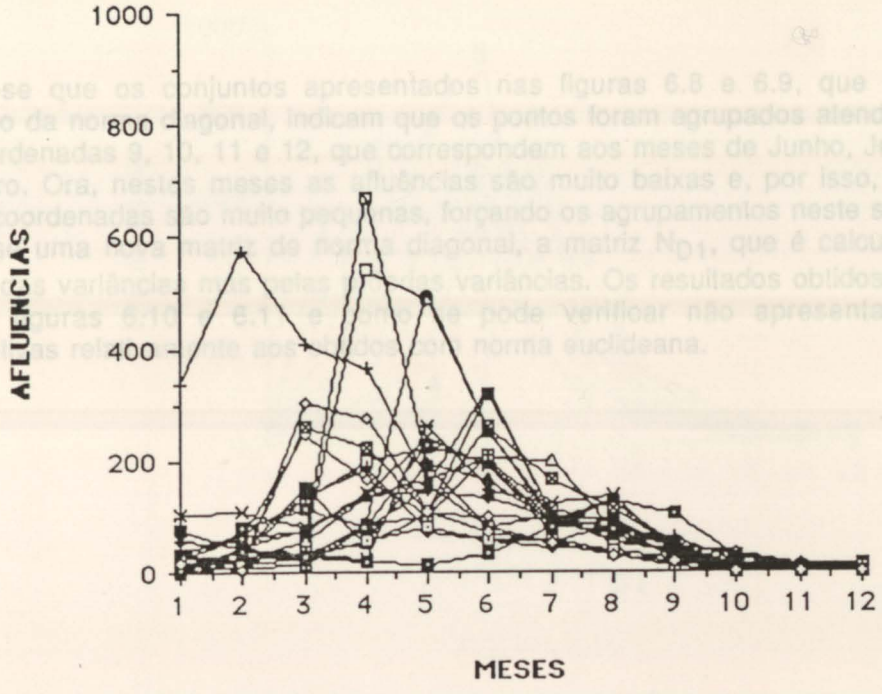
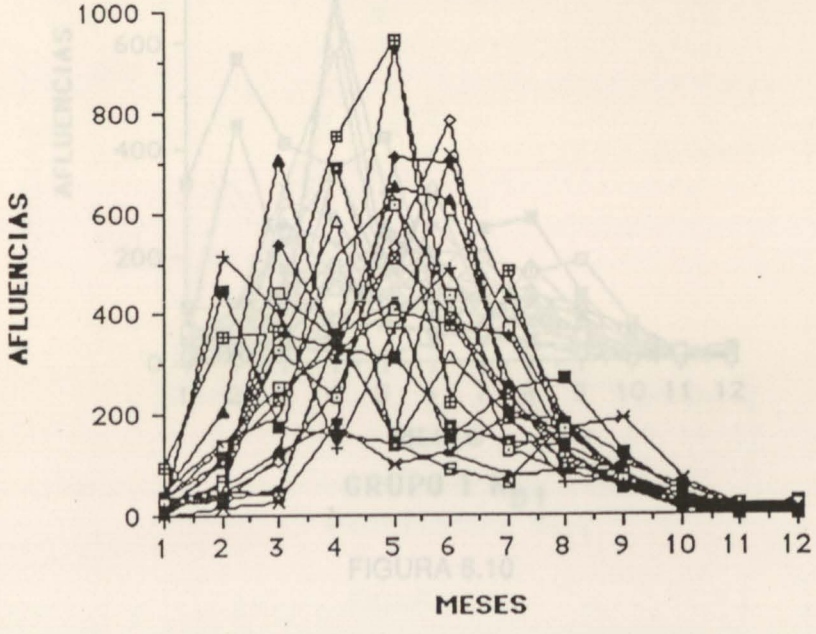


FIGURA 6.7



GRUPO 1 NORMA DIAGONAL

FIGURA 6.8



GRUPO 2 NORMA DIAGONAL

FIGURA 6.9

Repare-se que os conjuntos apresentados nas figuras 6.8 e 6.9, que se referem à utilização da norma diagonal, indicam que os pontos foram agrupados atendendo ao valor das coordenadas 9, 10, 11 e 12, que correspondem aos meses de Junho, Julho, Agosto e Setembro. Ora, nestes meses as afluências são muito baixas e, por isso, as variâncias destas coordenadas são muito pequenas, forçando os agrupamentos neste sentido. Assim, definiu-se uma nova matriz de norma diagonal, a matriz N_{D1}, que é calculada não pelo inverso das variâncias mas pelas próprias variâncias. Os resultados obtidos representam-se nas figuras 6.10 e 6.11 e como se pode verificar não apresentam alterações significativas relativamente aos obtidos com norma euclideana.

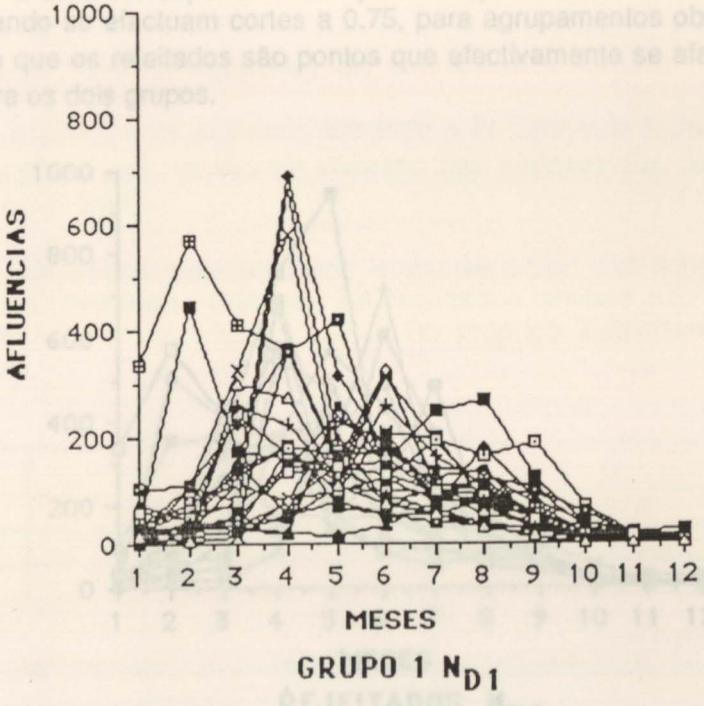


FIGURA 6.10

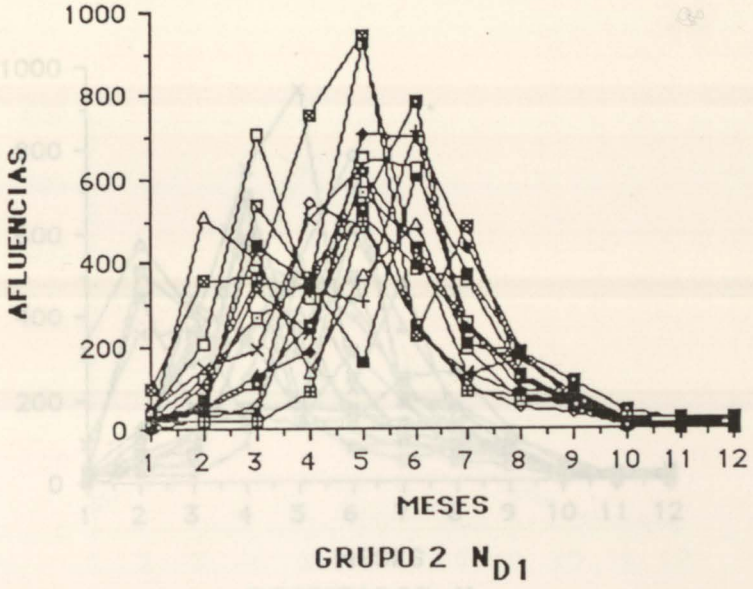


FIGURA 6.11

As figuras 6.12 e 6.13 correspondem aos pontos rejeitados, tal como foram definidos na secção 5.6, quando se efectuam cortes a 0.75, para agrupamentos obtidos utilizando N_E e N_{D1}. Repare-se que os rejeitados são pontos que efectivamente se afastam dos centróides encontrados para os dois grupos.

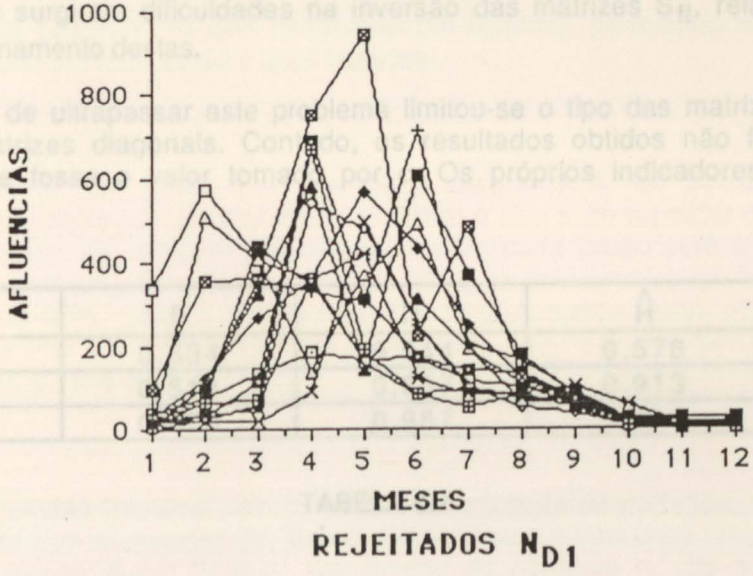


FIGURA 6.12

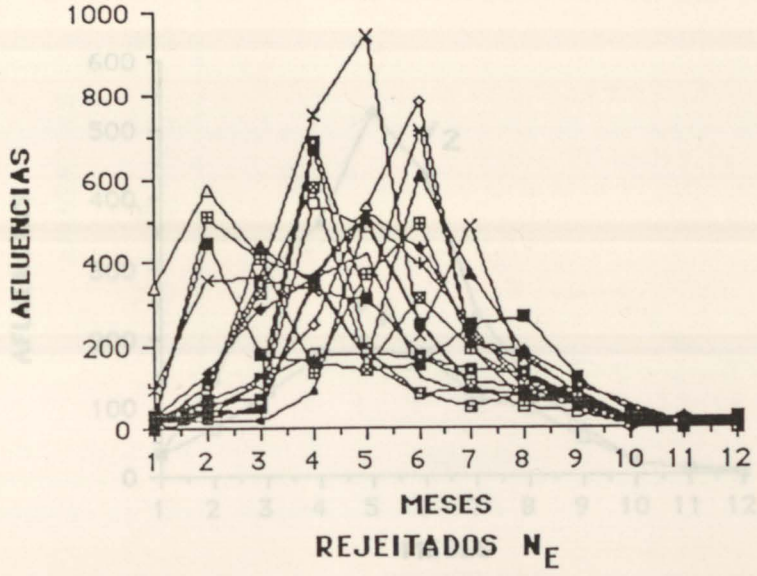


FIGURA 6.13

6.5.2. Resultados com o algoritmo de Gustafson e Kessel

Ao conjunto foi aplicado este algoritmo tomando ρ_i=1 para todo i, m=c=2. Para os outros valores de c surgiram dificuldades na inversão das matrizes S_{fi}, relacionados com um mau condicionamento destas.

Na tentativa de ultrapassar este problema limitou-se o tipo das matrizes que induzem a norma a matrizes diagonais. Contudo, os resultados obtidos não foram satisfatórios, qualquer que fosse o valor tomado por c. Os próprios indicadores (tabela 6.7) isso confirmam.

c	F	H	$\hat{H}$	$\hat{I}$
2	0.634	0.544	0.576	0.744
3	0.514	0.837	0.913	0.752
4	0.472	0.987	1.110	0.738

TABELA 6.7

6.5.3. Distribuições de Possibilidade

Da aplicação do FCM com norma euclideana e tomando c=2 resultaram os agrupamentos das figuras 6.6 e 6.7. Mais, é possível representar cada um destes grupos por um diagrama típico, o centróide. Na figura 6.14 estão representados os protótipos ou centróides para a partição obtida.

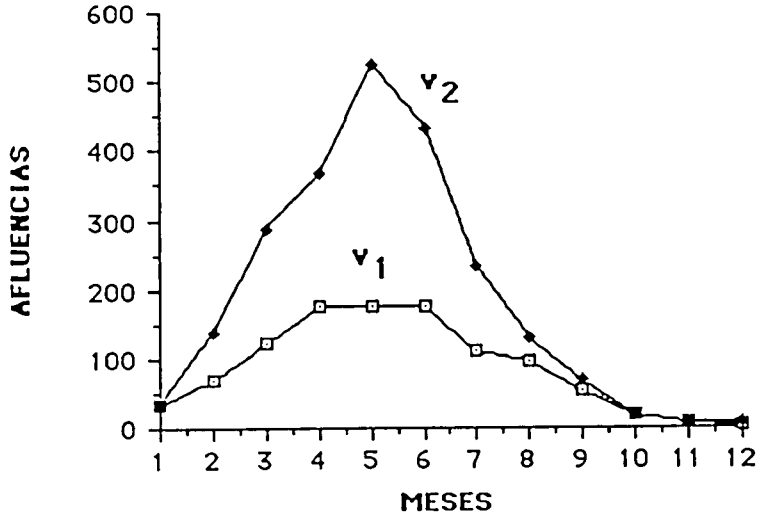


FIGURA 6.14

Como se pode observar, os dois centroides são distintos, permitindo interpretar a partição do conjunto dado em anos secos e anos húmidos.

Mas representar um grupo por um único diagrama é, de certa forma, ignorar a imprecisão associada ao conceito de ano húmido e ano seco e perder muita da informação disponível àcerca do conjunto em estudo. Então, cada grupo será identificado por uma "banda" desenvolvida em torno do centróide. Para se definirem essas "bandas" foram feitos cortes de nível 0.75, como é definido em 5.5. Nos conjuntos assim obtidos, encontraram-se valores máximo e mínimo para cada coordenada e as "bandas" resultantes representam-se nas figuras 6.15 e 6.16.

Da definição destas "bandas" resultará uma nova função de pertença, calculada com base nas distâncias dos diagramas às "bandas" e não aos centróides; diagramas que "caiam" inteiramente dentro duma "banda" têm função de pertença 1, os que caiam fora têm pertença 0 e a todos os outros corresponderão valores de pertença calculados com base nas distâncias definidas desta nova forma. Obtem-se, assim, a tipificação dos diagramas de uma forma imprecisa, sendo o diagrama tipo representado por uma "banda" e não por uma curva única.

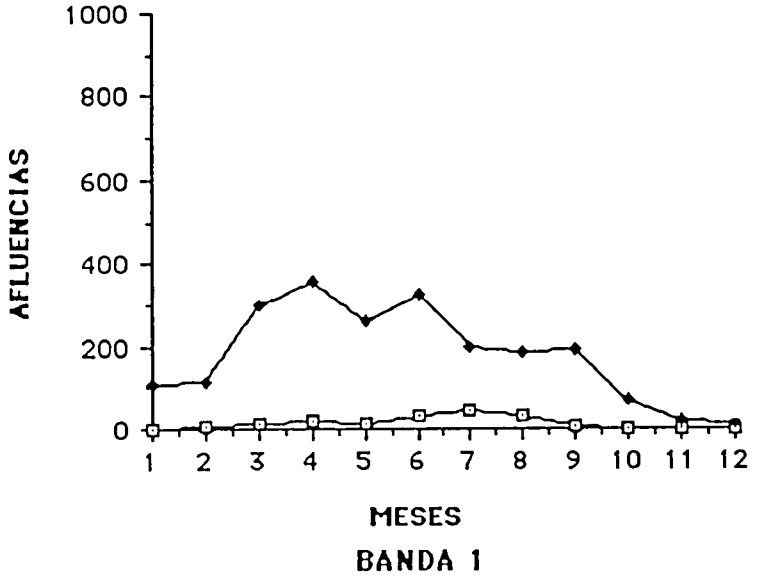


FIGURA 6.15

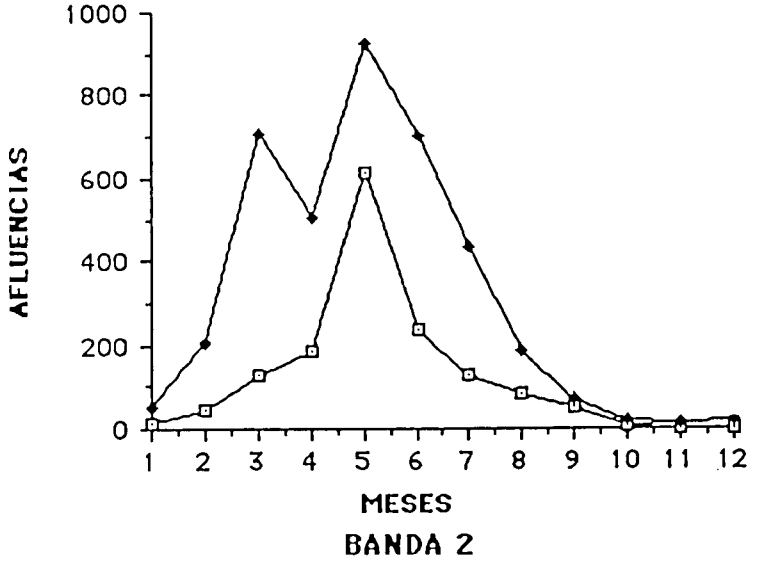


FIGURA 6.16

A aplicação desta metodologia a diagramas de carga de Sistemas Eléctricos encontra-se desenvolvida em [18]. Lamentavelmente, não existem ainda em Portugal, disponíveis para estudos independentes, conjuntos de leituras sobre amostras significativas de consumidores, pelo que se optou por não desenvolver esse exercício com base no exemplo em apresentação, referido a afluências hidrológicas.

7. Conclusões

Nesta dissertação abordou-se o tema da análise e classificação de agrupamentos por modelos imprecisos.

Procurou centrar-se a apreciação na distinção entre as partições obtidas por processos clássicos e as que resultam da aplicação de métodos baseados em teoria dos conjuntos imprecisos, dando-se especial relevo aos aspectos imprecisos que envolvem muitas situações reais.

Zadeh referia em 1969: " *A noção de conjunto impreciso é um bom ponto de partida para a construção de uma ferramenta conceptual que tem semelhança em muitos aspectos com a utilizada no caso dos conjuntos rígidos, mas mais geral do que esta e, potencialmente, mostra ter um campo de aplicação mais amplo, particularmente na área do reconhecimento de formas e processamento da informação. Essencialmente, tal ferrramenta corresponde a um modo natural de lidar com problemas nos quais a fonte de imprecisão é a ausência de critérios de pertença a classes rigorosamente definidos mais do que a presença de variáveis aleatórias*".

A imprecisão pode ser encontrada nas mais diversas áreas. Em particular, naquelas em que a avaliação e decisão humana são importantes.

Surgem situações em que as estruturas a identificar são de tal forma que representá-las de um modo dicotómico faz perder muita da informação de que se dispõe. As partições imprecisas, em que se deixou de classificar os elementos de um conjunto em pertencentes ou não pertencentes a uma classe e passou-se a defini-los como mais ou menos pertencentes, permitem traduzir de um modo mais adequado estes casos, em que a própria linguagem usada para os definir envolve conceitos de certa forma vagos. O exemplo referido em 6.2 ilustra bem esta diferença.

Conseguiu-se uma boa descrição e fundamentada de métodos não hierárquicos relevantes em análise de agrupamentos, em particular do Fuzzy c-Means de Bezdek, já que este processo desenvolve as ideias fundamentais em análise de agrupamentos. Todos os outros podem ser considerados como variações ou aperfeiçoamentos do FCM.

Centrou-se a análise na qualidade daspartições obtidas, como é efectuado em 6.4, onde são utilizadas várias medidas de validação das partições e, depois de uma observação directa dos resultados obtidos, se optou por uma partição em dois grupos.

Foi concebido e ensaiado um novo índice que revelou bons resultados nas avaliações que faz das partições obtidas. Este novo indicador oferece um atractivo especial sobre os outros que foram descritos na dissertação, já que envolve uma visão global sobre as estruturas que se possam identificar, incluindo a consideração explícita de "resíduos" ou "elementos rejeitados", numa abordagem que a alguns poderá recordar certas virtudes dos métodos hierárquicos. Notem-se os resultados obtidos em 5.8. O indicador apresenta, no entanto, uma característica básica comum a outros indicadores: o seu valor não depende directamente dos dados, mas das funções de pertença calculadas por meio de um dos algoritmos disponíveis. É por isso que é, também ele, sensível à metodologia com que se calculam os valores de pertença dos pontos aos agrupamentos e reflecte, portanto, mais a estrutura da matriz de partição imprecisa do que a estrutura intrínseca dos dados (embora, acredita-se, com uma nova ou, pelo menos, mais ampla perspectiva).

Pretendeu dar-se uma cobertura bastante completa do tema e apresentá-lo sob forma

didáctica, de modo que a dissertação possa constituir um bom material de referência para outros que se desenjem iniciar nestes assuntos. Sendo assim, incluiu-se um capítulo com exemplos tratados, para evidenciação das características mais interessantes de cada problema e cada método. Incluiu-se também, em anexo, uma listagem dos programas utilizados, pelo mesmo motivo.

Em síntese: a autora julga que a elaboração desta dissertação, satisfazendo os requisitos exigidos para um trabalho de MESTRADO, cumpriu o objectivo de formação pessoal numa área especializada do conhecimento, que não é habitualmente abordada nas disciplinas do curso, e cumpriu também o objectivo de constituir um trabalho de documentação para terceiros que o queiram prosseguir, abrindo perspectivas de investigação interessantes.

Referências

[1] Zadeh, L. A., Fuzzy Set, Inf. Control, vol. 8 (1965).

[2] Miyamoto, S., Fuzzy Sets in Information Retrieval and Cluster Analysis (1990).

[3] Zadeh, L. A., The Concept of a Linguistic Variable and its Aplication to Aproximate Reasoning Memorandum ERL-M 411 (1973).

[4] Bezdek,J. C., Pattern Recognition with Fuzzy Objective Function Algoritms, New York (1981).

[5] Bezdek, J. C. e Harris, J., Convex Decompositions of Fuzzy Partitions, J. Math. Anal. Apl., vol. 67-2, pp. 490-512 (1979).

[6] Bezdek,J. C., Pattern Recognition with Fuzzy Objective Function Algoritms, New York (1981).

[7] Bellman, R., Kalaba, R. e Zadeh L., Abstraction and Pattern Classification, J. Math. Anal. Apl., vol. 3, pp. 1-7 (1966).

[8] Ruspini, E., Numerical Methods for Fuzzy Clustering, Inf. Sci., vol. 2, pp. 319-350 (1970).

[9] Duda, R. e Hart, P., Pattern Classification and Scene Analysis, Wiley New York (1973).

[10] Bezdek, J. C., Fuzzy Mathematics in Pattern Classification, Ph. D. Thesis, Applied Math. Center, Cornell University, Ithaca (1973).

[11] Gustafson, D. E. e Kessel, W., Fuzzy Clustering with Fuzzy Covariance Matrix, Proc. IEEE-CDC, vol. 2, pp. 761-766, New Jersey (1979).

[12] Bezdek, J. C., Numerical Taxonomy with Fuzzy Sets, J. Math. Biol., vol.1-1, pp. 57-71 (1974).

[13] Bezdek, J. C., Mathematical Models for Systematics and Taxonomy,Proc. Eight Int. Conf. on Numerical Taxonomy, pp. 143-164 (1975).

[14] Windham, M. P., Cluster Validity for Fuzzy Clustering Algoritms, J. Fuzzy Sets Sist., vol. 3, pp.1-9 (1980).

[15] Dunn, J. C., Indices of Partition Fuzziness and the Detection of Clusters in Large Data Sets, Fuzzy Automata and Decision Processes, New York (1977).

[16] Gunderson, R., Application of Fuzzy ISODATA Algorithms to Star Tracker Pointing Sistems, Proc. 7th Trienal World IFAC Congress, pp. 1319-1323 (1978).

[17] Dunn,J. C., Well Separated Clusters and Optimal Fuzzy Partitions, J. Cybern, vol. 4-1, pp. 95-104 (1974).

[18] Miranda, V. e Matos, M., Determinação de Diagramas de Carga Imprecisos Uteis ao Planeamento de Energia Eléctrica, Investigação Operacional, vol.7 nº 2, pp.25-31(1987).

```
PROGRAM PROJ2 (input,output,dados,di,lpr);

type    sa=string[12];

const   zr=0.000000000000001;
        eps=0.001;
        it0=5;
        cd=3;

var     el1,p,it,y,l,col,com,fim,lim,iter,itermax,c,i,j,k,n,nz,dim,nc,k1:integer;
        el,sm,uij,ui,soma1,soma2,soma3,m:real;
        V,X:array[1..50,1..15] of real;
        D,U1,U2:array[1..50,1..50] of real;
        dr,di,dados,result:text;
        s1,r,r1,t,t1:sa;
        s:string[85];
        arred,enc,nulo,outran,resp:boolean;
        tab,rep,op,seg,seq:char;
        A:array[1..200] of sa;
        Peso,Soma:array[1..15] of real;
        Anorm:array[1..15,1..15] of real;
        H,F,HN:array[1..50] of real;
        VC:array[1..50] of integer;

FUNCTION NORMA(k,l:integer):real;
var     norm:real;
        i:integer;

begin
   norm:=0;
   for i:=1 to dim do
       norm:=norm+Peso[i]*(X[k,i]-V[l,i])*(X[k,i]-V[l,i]);
   NORMA:=sqrt(norm)
end;


FUNCTION OUTNORM(k,l:integer):real;
var  i,j:integer;
     soma:real;
     Dist,Dist1:array[1..15] of real;

begin
  for i:=1 to dim do
      Dist[i]:=X[k,i]-V[l,i];
  for j:=1 to dim do
  begin
    soma:=0;
    for i:=1 to dim do
      soma:=soma+(Dist[i]*Anorm[i,j]);
    Dist1[j]:=soma
  end;
  soma:=0;
  for i:=1 to dim do
      soma:=soma+(Dist1[i]*Dist[i]);
  OUTNORM:=sqrt(soma)
end;


FUNCTION POWER(a,b:real):real;

begin
  if a<0 then write('0 valor de a nao esta correcto');
  if a>0 then power:=exp(b*ln(a))
         else power:=0
end;
```

```
PROCEDURE limpecran;
var cod:char;

begin
  cod:=chr(27);
  write(cod,'[2J');
  write(cod,'[H')
end;


PROCEDURE gotoxy(x,y:integer);
var cod:char;

begin
  cod:=chr(27);
  write(cod,'[H');
  write(cod,'[',y,'B');
  write(cod,'[',x,'C')
end;


PROCEDURE Crianome;
var m1,n1,n3:integer;
    st,st1:sa;
    B1:array[1..8] of integer;

PROCEDURE Indiv(num:integer;var i:integer);
var quoc,rest:integer;

begin
  quoc:=num div 10;
  i:=0;
  while quoc<>0 do
  begin
     rest:=num mod 10;
     num:=quoc;
     i:=i+1;
     B1[i]:=rest;
     quoc:=num div 10
  end;
  i:=i+1;
  B1[i]:=num
end;

PROCEDURE Convert(n2:integer;var s2:sa);
var j:integer;
    q:string[1];

begin
  s2:='';q:=' ';
  for j:=n2 downto 1 do
  begin
     q[1]:=chr(48+B1[j]);
     s2:=concat(s2,q)
  end
end;

begin
   m1:=trunc(m*100);
   Indiv(c,n3);
   Convert(n3,st);
   Indiv(m1,n1);
   Convert(n1,st1);
   r:=concat(t,st,st1)
```

```
end;


PROCEDURE Lertab;
var i,j:integer;

begin
   for i:=1 to n do
   begin
      readln(dados,s);
      for j:=1 to dim do
        read(dados,X[i,j]);
      readln(dados,s);readln(dados,s)
   end
end;


PROCEDURE Arredondar;
var i,j:integer;

begin
  for i:=1 to c-1 do
    for j:=1 to n do
    begin
      el:=U2[i,j];
      el1:=round(el*y);
      el:=el1/y;
      U2[i,j]:=el
    end;
  for j:=1 to n do
  begin
    sm:=U2[1,j];
    for i:=2 to c-1 do
      sm:=sm+U2[i,j];
    U2[c,j]:=1.0-sm
  end
end;

                        {* PROGRAMA PRINCIPAL *}


begin
   repeat
      reset(di,'FICHDADOS');
      readln(di,l);
      if l<>0 then begin
                     col:=1;
                     com:=1;
                     lim:=15;
                     repeat
                       limpecran;
                       if col<lim then lim:=col;
                       fim:=com+lim-1;
                       writeln;writeln;
                       writeln('                    FICHEIROS DE DADOS EXISTENTES');
                       writeln;
                       for i:=com to fim do
                       begin
                         readln(di,t1);
                         writeln('                                        ',t1)
                       end;
                       col:=col-lim;
                       com:=fim+1;
                       repeat
                         gotoxy(60,24);
                         write('CONTINUE->RETURN');
```

```
                        readln(tab)
                      until tab=chr(32)
                    until col=0
                  end;
        close(di);
        limpecran;
        write('Qual o ficheiro de dados a utilizar? ');
        readln(t);
        reset(dados,t);readln(dados,s);readln(dados,s);
        readln(dados,s);readln(dados,s);
        readln(dados,n);readln(dados,s);
        readln(dados,s);readln(dados,dim);readln(dados,s);
        Lertab;
        repeat
          write('Qual a norma que pretende utilizar:euclideana(E),diagonal(D),ou outr
' );
          readln(op);
          outran:=false;resp:=true;
          case op of
            'E','e' : for i:=1 to dim do Peso[i]:=1;
            'D','d' : begin
                        writeln('Quais os pesos nas varias direccoes?');
                        for i:=1 to dim do
                        begin
                          write('Peso[',i,']=');readln(Peso[i])
                        end
                      end;
            'O','o' : begin
                        writeln('Indique a matriz definidora da norma');
                        for i:=1 to dim do
                          for j:=1 to dim do
                          begin
                            write('A[',i,',',j,']=');readln(Anorm[i,j])
                          end;
                        outran:=true
                      end;
            otherwise : begin
                          resp:=false;
                          writeln('Responda correctamente')
                        end
          end
        until resp;
        nc:=0;
        repeat
          write('Em quantos grupos pretende dividir o conjunto? ');readln(c);
          if c<>0 then begin
                         write('Qual o valor a atribuir ao parametro m? ');readln(m);
                         write('Qual o numero maximo de iteracoes que pretende? ');re
(itermax);
                         Crianome;
                         reset(dr,'FICHRESULT');
                         readln(dr,p);
                         if p<>0 then
                                   for i:=1 to p do
                                   begin
                                     readln(dr,r1);
                                     A[i]:=r1
                                   end;
                         close(dr);
                         enc:=false;i:=1;
                         while(i<=p) and not enc do
                         begin
                           if r=A[i] then enc:=true;
                           i:=i+1
                         end;
                         if not enc then
```

```
begin
  p:=p+1;
  rewrite(dr,'FICHRESULT');
  writeln(dr,p);
  if p=1 then writeln(dr,r)
         else
         begin
           for i:=1 to p-1 do
           begin
             s1:=A[i];
             writeln(dr,s1)
           end;
           writeln(dr,r)
         end;
  close(dr)
end;

    {*  INICIALIZACAO DA MATRIZ U  *}

for i:=1 to c do
  for j:=1 to n do U1[i,j]:=0;
for i:=1 to c do U1[i,i]:=1;
for j:=c+1 to n do U1[1,j]:=1;


iter:=0;it:=it0;y:=round(exp(cd*ln(10)));
repeat
  arred:=false;
  iter:=iter+1;

    {*  CALCULO DOS CENTROIDES  *}

  for i:=1 to c do
  begin
    soma3:=0;
    for k1:=1 to dim do Soma[k1]:=0;
    for j:=1 to n do
    begin
      uij:=POWER(U1[i,j],m);
      for k1:=1 to dim do
        Soma[k1]:=Soma[k1]+uij*X[j,k1];
      soma3:=soma3+uij
    end;
    for k1:=1 to dim do
      V[i,k1]:=Soma[k1]/soma3
  end;

     {*  CALCULO DA MATRIZ U  *}

  for j:=1 to n do
  begin
    nz:=0;
    for i:=1 to c do
    begin
      if outran then D[i,j]:=OUTNORM(j,i)
                else D[i,j]:=NORMA(j,i);
      if D[i,j]<zr then nz:=nz+1
    end;
    if nz>0 then
            begin
              for i:=1 to c do
                if D[i,j]>zr then U2[i,j]:=0
                             else U2[i,j]:=1/nz
            end
            else
            begin
```

```
                    for i:=1 to c do
                    begin
                      ui:=0;
                      for k:=1 to c do
                      ui:=ui+POWER(D[i,j]/D[k,j],2/(m-1));
                      U2[i,j]:=1/ui
                    end
                  end
  end;

       {*  TESTE DE CONVERGENCIA  *}

  nulo:=true;
  i:=1;
  while(i<=c) and nulo do
  begin
    j:=1;
    while(j<=n) and nulo do
    begin
      if abs (U2[i,j]-U1[i,j])>eps then nulo:=false;
      j:=j+1
    end;
    i:=i+1
  end;
  if nulo=false then
                   for i:=1 to c do
                     for j:=1 to n do U1[i,j]:=U2[i,j];
  if iter=it then
              begin
                it:=it+it0;
                Arredondar;
                arred:=true
              end
until nulo or (iter=itermax);
if not arred then Arredondar;

  {*  CALCULO DOS COEFICIENTES DE VALIDADE  *}

soma1:=0;
soma2:=0;
for i:=1 to c do
  for j:=1 to n do
  begin
    soma1:=soma1+(U2[i,j]*U2[i,j]/n);
    if U2[i,j]<>0 then soma2:=soma2+U2[i,j]*ln(U2[i,j])/n
  end;
nc:=nc+1;
F[nc]:=soma1;
H[nc]:=-soma2;
HN[nc]:=H[nc]/(1-c/n);
VC[nc]:=c;

   {*  APRESENTACAO DE RESULTADOS  *}

col:=n;
com:=1;
lim:=10;
repeat
  limpecran;
  if col<lim then lim:=col;
  fim:=com+lim-1;
  writeln('NUMERO DE ITERACOES EXECUTADAS:',iter);
  writeln;writeln;
  writeln('                          ***  COORDENADAS DOS CENTROI
***');
  writeln;
```

```
                    for i:=1 to c do
                    begin
                      writeln('Coordenadas do centroide do grupo',i);
                      for k1:=1 to dim do write(V[i,k1]:5:3,'  ');
                      writeln;writeln
                    end;
                    writeln;
                    writeln('                              ***  MATRIZ U  ***');
                    writeln;writeln;
                    for i:=1 to c do
                    begin
                      for j:=com to fim do
                        write(U2[i,j]:5:3,'  ');
                      writeln
                    end;
                    col:=col-lim;
                    com:=fim+1;
                    repeat
                      gotoxy(60,24);
                      write('CONTINUE->RETURN');
                      readln(seq)
                    until seq=chr(32)
                  until col=0;
                  limpecran;
                  writeln;writeln;writeln;writeln;writeln;writeln;
                  writeln('             Valor do indicador F');
                  writeln('             F(',c,')=',F[nc]:5:3);
                  writeln;
                  writeln('             Valor do indicador H');
                  writeln('             H(',c,')=',H[nc]:5:3);
                  writeln;
                  writeln('             Valor do indicador HN');
                  writeln('             HN(',c,')=',HN[nc]:5:3);
                  repeat
                    gotoxy(60,24);
                    write('CONTINUE->RETURN');
                    readln(seg)
                  until seg=chr(32);
                  rewrite(result,r);
                  writeln(result,'RESULTADOS OBTIDOS PARA O FICHEIRO ',t);
                  writeln(result);writeln(result);
                  writeln(result,'Numero de grupos em que foi dividido o conju
',c);
                  writeln(result,'Valor do parametro m: ',m:4:2);
                  writeln(result,'Numero de iteracoes executadas: ',iter);
                  case op of
                    'E','e':writeln(result,'Norma utilizada:Norma Euclideana')

                    'D','d':writeln(result,'Norma utilizada:Norma Diagonal');
                    'O','o':writeln(result,'Norma utilizada:Outra Norma');
                    otherwise:begin
                              end
                  end;
                  writeln(result);writeln(result);
                  writeln(result,'                    *** COORDENADAS DOS CENTR
***');
                  writeln(result);
                  for i:=1 to c do
                  begin
                    writeln(result,'Coordenadas do centroide do grupo',i);
                    for k1:=1 to dim do
                      writeln(result,V[i,k1]:5:3,'  ');
                    writeln(result)
                  end;
                  writeln(result);
                  writeln(result,'                              **  MATRIZ U  **'
```

```
                    writeln(result);writeln(result);
                    col:=n;
                    com:=1;
                    lim:=10;
                    repeat
                      if col<lim then lim:=col;
                      fim:=com+lim-1;
                      for i:=1 to c do
                      begin
                        for j:=com to fim do
                          write(result,U2[i,j]:5:3,'  ');
                        writeln(result)
                      end;
                      writeln(result);writeln(result);
                      col:=col-lim;
                      com:=fim+1
                    until col=0;
                    writeln(result);writeln(result);
                    writeln(result,'Valor do indicador F :',F[nc]:5:3);
                    writeln(result);
                    writeln(result,'Valor do indicador H :',H[nc]:5:3);
                    writeln(result);
                    writeln(result,'Valor do indicador HN:',HN[nc]:5:3);
                  end
                until c=0;
                limpecran;
                writeln;writeln;writeln;writeln;
                writeln('               TABELA DOS COEFICIENTES DE VALIDADE');
                writeln;writeln;
                writeln('               c           F          H           HN');
                writeln;
                for j:=1 to nc do
                  writeln('                ',VC[j],'        ',F[j]:5:3,'      '
5:3,'       ',HN[j]:5:3);
                writeln;writeln;writeln;
                write('Quer utilizar outros dados?(S/N)->');readln(rep)
  until (rep='N') or (rep='n')
end.
```

```
PROGRAM PROJ4 (input,output);

type   sa=string[12];
       ma=array[1..24,1..24] of double;

const  zr=0.00000000000000000001;
       eps=0.075;
       it0=5;
       cd=3;

var   i3,j3,i2,j2,p,it,l,col,com,fim,lim,iter,itermax,c,i,j,k,n,nz,dim,dim1,nc,k1,k
4:integer;
      det,soma5,smv,media,prod,soma4,pi,sm,el1,y,el,uij,ui,soma1,soma2,soma3,m,tc:d

       V,X,XI:array[1..50,1..24] of double;
       D,U1,U2:array[1..50,1..50] of double;
       dr,di,dados,result:text;
       s1,r,r1,t,t1:sa;
       s:string[85];
       arred,enc,nulo:boolean;
       tab,rep,op,seq,seg:char;
       A:array[1..100] of sa;
       Dis,Soma:array[1..24] of double;
       Si,Sinv,Anorm,MP:ma;
       H,F,HN:array[1..50] of double;
       VC:array[1..50] of integer;
       Elm,DV,DVN:array[1..24] of double;



FUNCTION NORMA(k,l:integer):double;
var  i,j:integer;
     soma:double;
     Dist,Dist1:array[1..24] of double;

begin
  for i:=1 to dim do
      Dist[i]:=X[k,i]-V[l,i];
  for j:=1 to dim do
  begin
    soma:=0;
    for i:=1 to dim do
      soma:=soma+(Dist[i]*Anorm[i,j]);
    Dist1[j]:=soma
  end;
  soma:=0;
  for i:=1 to dim do
      soma:=soma+(Dist1[i]*Dist[i]);
  NORMA:=sqrt(soma)
end;


FUNCTION POWER(a,b:double):double;

begin
  if a<0 then write('0 valor de a nao esta correcto');
  if a>0 then power:=exp(b*ln(a))
         else power:=0
end;


PROCEDURE limpecran;
var cod:char;

begin
```

```
  cod:=chr(27);
  write(cod,'[2J');
  write(cod,'[H')
end;


PROCEDURE gotoxy(x,y:integer);
var cod:char;

begin
  cod:=chr(27);
  write(cod,'[H');
  write(cod,'[',y,'B');
  write(cod,'[',x,'C')
end;


PROCEDURE Crianome;
var m1,n1,n3:integer;
    st,st1:sa;
    B1:array[1..8] of integer;

PROCEDURE Indiv(num:integer;var i:integer);
var quoc,rest:integer;

begin
  quoc:=num div 10;
  i:=0;
  while quoc<>0 do
  begin
     rest:=num mod 10;
     num:=quoc;
     i:=i+1;
     B1[i]:=rest;
     quoc:=num div 10
  end;
  i:=i+1;
  B1[i]:=num
end;

PROCEDURE Convert(n2:integer;var s2:sa);
var j:integer;
    q:string[1];

begin
  s2:='';q:=' ';
  for j:=n2 downto 1 do
  begin
     q[1]:=chr(48+B1[j]);
     s2:=concat(s2,q)
  end
end;

begin
   m1:=trunc(m*100);
   Indiv(c,n3);
   Convert(n3,st);
   Indiv(m1,n1);
   Convert(n1,st1);
   r:=concat(t,st,st1)
end;


PROCEDURE Lertab;
var i,j:integer;
```

```
begin
   for i:=1 to n do
   begin
      readln(dados,s);
      for j:=1 to dim1 do
        read(dados,XI[i,j]);
      readln(dados,s);readln(dados,s)
   end
end;

PROCEDURE Lertab1;
var i,j:integer;
    z:double;

begin
  for i:=1 to n do
  begin
    k3:=0;
    for j:=1 to dim1 do
    begin
      if j=Elm[k3+1] then
                      begin
                        k3:=k3+1;
                        X[i,k3]:=XI[i,j]
                      end
    end;
    readln(dados,s);readln(dados,s)
  end
end;


PROCEDURE Matinv (var Minv:ma);
var M,L,U,M1,MP:ma;
    Termos:array[1..24] of double;
    Index:array[1..24] of integer;
    i,j,k:integer;
    normm,norml,normu,ks,prod1,soma7:double;

FUNCTION NORMAT(A:ma):double;
var max,soma:double;
    i,j:integer;

begin
  max:=0;
  for i:=1 to dim do
  begin
    soma:=0;
    for j:=1 to dim do
      soma:=soma+abs(A[i,j]);
    if soma>max then max:=soma
  end;
  NORMAT:=max
end;

PROCEDURE Lermat;
var i1,j1:integer;

begin
  for i1:=1 to dim do
    for j1:=1 to dim do
      M[i1,j1]:=Si[i1,j1]
end;

PROCEDURE Invert(var A:ma);
var i,j,k,l,ll,irow,icol:integer;
    pivinv,dum,big:double;
```

```
    indexc,indexr,ipiv:array[1..15] of integer;

begin
  for j:=1 to dim do
    ipiv[j]:=0;
  for i:=1 to dim do
  begin
    big:=0;
    for j:=1 to dim do
    begin
      if ipiv[j]<>1 then
                    begin
                      for k:=1 to dim do
                      begin
                        if ipiv[k]=0 then
                                     begin
                                       if abs(A[j,k])>=big then
                                                      begin
                                                        big:=abs(A[j,k]);
                                                        irow:=j;
                                                        icol:=k
                                                      end
                                     end
                      end
                    end
    end;
    ipiv[icol]:=ipiv[icol]+1;
    if irow<>icol then
                  begin
                    for l:=1 to dim do
                    begin
                      dum:=A[irow,l];
                      A[irow,l]:=A[icol,l];
                      A[icol,l]:=dum
                    end
                  end;
    indexr[i]:=irow;
    indexc[i]:=icol;
    pivinv:=1/A[icol,icol];
    A[icol,icol]:=1.0;
    for l:=1 to dim do
      A[icol,l]:=A[icol,l]*pivinv;
    for ll:=1 to dim do
    begin
      if ll<>icol then
                  begin
                    dum:=A[ll,icol];
                    A[ll,icol]:=0;
                    for l:=1 to dim do
                      A[ll,l]:=A[ll,l]-A[icol,l]*dum
                  end
    end
  end;
  for l:=dim downto 1 do
  begin
    if indexr[l]<>indexc[l] then
                            begin
                              for k:=1 to dim do
                              begin
                                dum:=A[k,indexr[l]];
                                A[k,indexr[l]]:=A[k,indexc[l]];
                                A[k,indexc[l]]:=dum
                              end
                            end
  end
end;
```

```
PROCEDURE Declu;
var i1max,k5,i1,j1:integer;
    dum,sum,mmax:double;
    Scl:array[1..24] of double;

begin
  for i1:=1 to dim do
  begin
    mmax:=0;
    for j1:=1 to dim do
      if abs(M[i1,j1])>mmax then mmax:=abs(M[i1,j1]);
    if mmax=0 then writeln('       A MATRIZ NAO E INVERTIVEL');
    Scl[i1]:=1/mmax
  end;
  for j1:=1 to dim do
  begin
    if j1>1 then
            begin
              for i1:=1 to j1-1 do
              begin
                sum:=M[i1,j1];
                if i1>1 then for k5:=1 to i1-1 do
                                sum:=sum-M[i1,k5]*M[k5,j1];
                M[i1,j1]:=sum
              end
            end;
     mmax:=0;
     for i1:=j1 to dim do
     begin
       sum:=M[i1,j1];
       if j1>1 then for k5:=1 to j1-1 do
                      sum:=sum-M[i1,k5]*M[k5,j1];
       M[i1,j1]:=sum;
       dum:=Scl[i1]*abs(sum);
       if dum>mmax then
                    begin
                      i1max:=i1;
                      mmax:=dum
                    end
     end;
     if j1<>i1max then
                  begin
                    for k5:=1 to dim do
                    begin
                      dum:=M[i1max,k5];
                      M[i1max,k5]:=M[j1,k5];
                      M[j1,k5]:=dum
                    end;
                    det:=-det;
                    Scl[i1max]:=Scl[j1]
                  end;
     Index[j1]:=i1max;
     if M[i1,j1]=0 then M[i1,j1]:=zr;
     if j1<dim then
               begin
                 dum:=1/M[j1,j1];
                 for i1:=j1+1 to dim do M[i1,j1]:=M[i1,j1]*dum
               end
   end
end;


PROCEDURE Subst;
var ii,ll,i1,j1:integer;
```

```
    sum:double;

begin
  ii:=0;
  for i1:=1 to dim do
  begin
    ll:=Index[i1];
    sum:=Termos[ll];
    Termos[ll]:=Termos[i1];
    if ii>0 then
            begin
              if i1>1 then for j1:=ii to i1-1 do
                              sum:=sum-M[i1,j1]*Termos[j1]
            end
            else if sum<>0 then ii:=i1;
    Termos[i1]:=sum
  end;
  for i1:=dim downto 1 do
  begin
    sum:=Termos[i1];
    if i1<n then for j1:=i1+1 to dim do
                   sum:=sum-M[i1,j1]*Termos[j1];
    Termos[i1]:=sum/M[i1,i1]
  end
end;

              {*** COMECA INVERSAO ***}

begin
  det:=1;
  Lermat;
  normm:=NORMAT(M);
  writeln('    NORMA MATRIZ S=',normm);
  Declu;
 { writeln('              MATRIZ DECOMPOSTA');
  writeln;
  col:=dim;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i:=1 to dim do
    begin
      for j:=com to fim do
        write(M[i,j]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for i:=1 to dim do
  begin
    for j:=1 to dim do
    begin
      if j<i then
            begin
              U[i,j]:=0;
              L[i,j]:=M[i,j]
            end
            else
            begin
              U[i,j]:=M[i,j];
              if j=i then L[i,j]:=1
                     else L[i,j]:=0
            end
    end
```

```
end;
{writeln('          MATRIZ L');
writeln;
col:=dim;
com:=1;
lim:=3;
repeat
  if col<lim then lim:=col;
  fim:=com+lim-1;
  for i:=1 to dim do
  begin
    for j:=com to fim do
      write(L[i,j]);
    writeln
  end;
  writeln;writeln;
  col:=col-lim;
  com:=fim+1
until col=0;
writeln('          MATRIZ U');
writeln;
col:=dim;
com:=1;
lim:=3;
repeat
  if col<lim then lim:=col;
  fim:=com+lim-1;
  for i:=1 to dim do
  begin
    for j:=com to fim do
      write(U[i,j]);
    writeln
  end;
  writeln;writeln;
  col:=col-lim;
  com:=fim+1
until col=0;}
for i:=1 to dim do
  for j:=1 to dim do
    M1[i,j]:=L[i,j];
Invert(L);
{writeln('           INVERSA DE L');
writeln;
col:=dim;com:=1;lim:=3;
repeat
  if col<lim then lim:=col;
  fim:=com+lim-1;
  for i:=1 to dim do
  begin
    for j:=com to fim do
      write(L[i,j]);
    writeln
  end;
  writeln;writeln;
  col:=col-lim;
  com:=fim+1
until col=0;}
Invert(U);
{writeln('          INVERSA DE U');
writeln;
col:=dim;com:=1;lim:=3;
repeat
  if col<lim then lim:=col;
  fim:=com+lim-1;
  for i:=1 to dim do
  begin
```

```
      for j:=com to fim do
        write(U[i,j]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;
  for i:=1 to dim do
    for j:=1 to dim do
    begin
      soma7:=0;
      for k:=1 to dim do
      begin
        prod1:=M1[i,k]*L[k,j];
        soma7:=soma7+prod1
      end;
      MP[i,j]:=soma7
    end;
  writeln('   PRODUTO L POR INVL');
  writeln;
  col:=dim;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i:=1 to dim do
    begin
      for j:=com to fim do
        write(MP[i,j]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  norml:=NORMAT(L);
  writeln('      NORMA INVERSA DE L=',norml);
  normu:=NORMAT(U);
  writeln('      NORMA INVERSA DE U=',normu);
  ks:=normm*norml*normu;
  writeln('     FACTOR DE CONDICIONAMENTO=',ks);
  for j:=1 to dim do
  begin
    for i:=1 to dim do Termos[i]:=0;
    Termos[j]:=1;
    Subst;
    for i:=1 to dim do Minv[i,j]:=Termos[i];
    det:=det*M[j,j]
  end
end;


PROCEDURE Arredondar;
var i,j:integer;

begin
  for i:=1 to c-1 do
    for j:=1 to n do
    begin
      el:=U2[i,j];
      el1:=round(el*y);
      el:=el1/y;
      U2[i,j]:=el
    end;
  for j:=1 to n do
  begin
```

```
    sm:=U2[1,j];
    for i:=2 to c-1 do
      sm:=sm+U2[i,j];
    U2[c,j]:=1.0-sm
  end
end;

                        {* PROGRAMA PRINCIPAL *}


begin
   repeat
      reset(di,'FICHDADOS');
      readln(di,l);
      if l<>0 then begin
                     col:=l;
                     com:=1;
                     lim:=15;
                     repeat
                       limpecran;
                       if col<lim then lim:=col;
                       fim:=com+lim-1;
                       writeln;writeln;
                       writeln('                    FICHEIROS DE DADOS EXISTENTES');
                       writeln;
                       for i:=com to fim do
                       begin
                         readln(di,t1);
                         writeln('                                        ',t1)
                       end;
                       col:=col-lim;
                       com:=fim+1;
                       repeat
                         gotoxy(60,24);
                         write('CONTINUE->RETURN');
                         readln(tab)
                       until tab=chr(32)
                     until col=0
                   end;
      close(di);
      limpecran;
      write('Qual o ficheiro de dados a utilizar? ');
      readln(t);
      reset(dados,t);readln(dados,s);readln(dados,s);
      readln(dados,s);readln(dados,s);
      readln(dados,n);readln(dados,s);
      readln(dados,s);readln(dados,dim1);readln(dados,s);
      Lertab;
                   {*** CALCULO DESVIOS PADROES ***}

      for j:=1 to dim1 do
      begin
        smv:=0;
        soma5:=0;
        for i:=1 to n do
          soma5:=soma5+XI[i,j];
        media:=soma5/n;
        for k4:=1 to n do
          smv:=smv+(XI[k4,j]-media)*(XI[k4,j]-media);
        DV[j]:=sqrt(smv/n);
        DVN[j]:=DV[j]/media
      end;
      writeln('          DESVIOS PADROES COORDENADAS');
      for j:=1 to dim1 do
        writeln('Desvio da coordenada ',j,':',DV[j]);
      writeln;writeln;
```

```
        writeln('         DESVIOS PADROES NORMALIZADOS');
        for j:=1 to dim1 do
          writeln('Desvio da coordenada ',j,':',DVN[j]);
        write('Quantas coordenadas pretende utilizar? ');readln(dim);
        for i:=1 to dim do
        begin
          write('Qual a coordenada ',i,'? ');readln(Elm[i])
        end;
        Lertab1;
        nc:=0;
        repeat
          write('Em quantos grupos pretende dividir o conjunto? ');readln(c);
          if c<>0 then begin
                         write('Qual o valor a atribuir ao parametro m? ');readln(m);
                         write('Qual o numero maximo de iteracoes que pretende? ');re
itermax);
                         Crianome;
                         reset(dr,'FICHRESULT');
                         readln(dr,p);
                         if p<>0 then
                                   for i:=1 to p do
                                   begin
                                     readln(dr,r1);
                                     A[i]:=r1
                                   end;
                         close(dr);
                         enc:=false;i:=1;
                         while(i<=p) and not enc do
                         begin
                           if r=A[i] then enc:=true;
                           i:=i+1
                         end;
                         if not enc then
                         begin
                           p:=p+1;
                           rewrite(dr,'FICHRESULT');
                           writeln(dr,p);
                           if p=1 then writeln(dr,r)
                                  else
                                  begin
                                    for i:=1 to p-1 do
                                    begin
                                      s1:=A[i];
                                      writeln(dr,s1)
                                    end;
                                    writeln(dr,r)
                                  end;
                           close(dr)
                         end;

                             {*  INICIALIZACAO DA MATRIZ U  *}

                         for i:=1 to c do
                           for j:=1 to n do U1[i,j]:=0;
                         i:=1;
                         repeat
                           j:=i;
                           repeat
                             U1[i,j]:=1;
                             j:=j+c
                           until j>n;
                           i:=i+1
                         until i>c;

                         iter:=0;it:=it0;y:=round(exp(cd*ln(10)));
                         repeat
```

```
arred:=false;
iter:=iter+1;

  {*  CALCULO DOS CENTROIDES  *}

for i:=1 to c do
begin
  soma3:=0;
  for k1:=1 to dim do Soma[k1]:=0;
  for j:=1 to n do
  begin
    uij:=POWER(U1[i,j],m);
    for k1:=1 to dim do
      Soma[k1]:=Soma[k1]+uij*X[j,k1];
    soma3:=soma3+uij
  end;
  for k1:=1 to dim do
    V[i,k1]:=Soma[k1]/soma3
end;

   {*  CALCULO DA MATRIZ U  *}

for i:=1 to c do
begin
  for i2:=1 to dim do
    for j2:=1 to dim do
      Si[i2,j2]:=0;
  for k:=1 to n do
  begin
    uij:=POWER(U1[i,k],m);
    for j2:=1 to dim do
      Dis[j2]:=X[k,j2]-V[i,j2];
    for i2:=1 to dim do
      for j2:=1 to dim do
        Si[i2,j2]:=Si[i2,j2]+Dis[i2]*Dis[j2]*uij
  end;
  writeln('          ITERACAO ',iter);
  writeln('      MATRIZ S');
  writeln;writeln;
  col:=dim;
  com:=1;
  lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Si[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
 until col=0;
 Matinv(Sinv);
{ writeln('           INVERSA DE S');
 writeln;
 col:=dim;
 com:=1;
 lim:=3;
 repeat
   if col<lim then lim:=col;
   fim:=com+lim-1;
   for i2:=1 to dim do
   begin
```

```
      for j2:=com to fim do
        write(Sinv[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for i3:=1 to dim do
    for j3:=1 to dim do
    begin
      soma4:=0;
      for k3:=1 to dim do
      begin
        prod:=Si[i3,k3]*Sinv[k3,j3];
        soma4:=soma4+prod
      end;
      MP[i3,j3]:=soma4
    end;
  writeln('           MATRIZ PRODUTO');
  writeln;
  col:=dim;
  com:=1;
  lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i3:=1 to dim do
    begin
      for j3:=com to fim do
        write(MP[i3,j3]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;
  writeln('DETERMINANTE DE S=',det);
  writeln;writeln;
  for i2:=1 to dim do
    for j2:=1 to dim do
    begin
      pi:=POWER(det,1/dim);
      Anorm[i2,j2]:=pi*Sinv[i2,j2]
    end;
 { writeln('       MATRIZ NORMA');
  writeln;
  col:=dim;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Anorm[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for j:=1 to n do D[i,j]:=NORMA(j,i)
end;
for j:=1 to n do
begin
  nz:=0;
```

```
    for i:=1 to c do
      if D[i,j]<zr then nz:=nz+1;
    if nz>0 then
            begin
              for i:=1 to c do
                if D[i,j]>zr then U2[i,j]:=0
                             else U2[i,j]:=1/nz
            end
            else
            begin
              for i:=1 to c do
              begin
                ui:=0;
                for k:=1 to c do
                  ui:=ui+POWER(D[i,j]/D[k,j],2/(m-1));
                U2[i,j]:=1/ui
              end
            end
  end;
{ writeln('         MATRIZ DE PERTENCA');writeln;
  col:=n;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i:=1 to c do
    begin
      for j:=com to fim do
        write(U2[i,j]);
      writeln
    end;
    writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}

       {*  TESTE DE CONVERGENCIA  *}

  for i:=1 to c do
  begin
    for j:=1 to n do
    begin
      tc:=abs(U2[i,j]-U1[i,j]);
      writeln('tc(',i,j,')=',tc)
    end
  end;
  nulo:=true;
  i:=1;
  while(i<=c) and nulo do
  begin
    j:=1;
    while(j<=n) and nulo do
    begin
      if abs (U2[i,j]-U1[i,j])>eps then nulo:=false;
      j:=j+1
    end;
    i:=i+1
  end;
  if nulo=false then
                  for i:=1 to c do
                    for j:=1 to n do U1[i,j]:=U2[i,j];
  if iter=it then
             begin
               it:=it+it0;
               Arredondar;
               arred:=true
             end
```

```
until nulo or (iter=itermax);
if not arred then Arredondar;

  {*  CALCULO DOS COEFICIENTES DE VALIDADE  *}

soma1:=0;
soma2:=0;
for i:=1 to c do
  for j:=1 to n do
  begin
    soma1:=soma1+(U2[i,j]*U2[i,j]/n);
    if U2[i,j]<>0 then soma2:=soma2+U2[i,j]*ln(U2[i,j])/n
  end;
nc:=nc+1;
F[nc]:=soma1;
H[nc]:=-soma2;
HN[nc]:=H[nc]/(1-c/n);
VC[nc]:=c;

   {*  APRESENTACAO DE RESULTADOS  *}

col:=n;
com:=1;
lim:=10;
repeat
  limpecran;
  if col<lim then lim:=col;
  fim:=com+lim-1;
  if nulo=true then writeln('CONVERGENCIA ATINGIDA');
  writeln('NUMERO DE ITERACOES EXECUTADAS:',iter);
  writeln;
  writeln('             ***  COORDENADAS DOS CENTROIDES  ***

  writeln;
  for i:=1 to c do
  begin
    writeln('Coordenadas do centroide do grupo',i);
    for k1:=1 to dim do write(V[i,k1]:5:3,'  ');
    writeln;writeln
  end;
  writeln('               ***  MATRIZ U  ***');
  writeln;
  for i:=1 to c do
  begin
    for j:=com to fim do
      write(U2[i,j]:5:3,'  ');
    writeln
  end;
  col:=col-lim;
  com:=fim+1;
  repeat
    gotoxy(60,24);
    write('CONTINUE->RETURN');
    readln(seq)
  until seq=chr(32)
until col=0;
limpecran;
writeln;writeln;writeln;writeln;writeln;writeln;
writeln('             Valor do indicador F');
writeln('             F(',c,')=',F[nc]:5:3);
writeln;
writeln('             Valor do indicador H');
writeln('             H(',c,')=',H[nc]:5:3);
writeln;
writeln('             Valor do indicador HN');
writeln('             HN(',c,')=',HN[nc]:5:3);
```

```
                repeat
                  gotoxy(60,24);
                  write('CONTINUE->RETURN');
                  readln(seg)
                until seg=chr(32);
                rewrite(result,r);
                writeln(result,'RESULTADOS OBTIDOS PARA O FICHEIRO ',t);
                writeln(result);writeln(result);writeln(result);
                writeln(result,'Numero de grupos em que foi dividido o conju
:',c);
                writeln(result,'Valor do parametro m: ',m:4:2);
                writeln(result,'Numero de iteracoes executadas: ',iter);
                writeln(result);
                writeln(result,'          *** COORDENADAS DOS CENTROIDES  **
;
                writeln(result);
                for i:=1 to c do
                begin
                  writeln(result,'Coordenadas do centroide do grupo',i);
                  for k1:=1 to dim do
                    writeln(result,V[i,k1]:5:3,'  ');
                  writeln(result)
                end;
                writeln(result,'             **  MATRIZ U  **');
                writeln(result);
                col:=n;
                com:=1;
                lim:=10;
                repeat
                  if col<lim then lim:=col;
                  fim:=com+lim-1;
                  for i:=1 to c do
                  begin
                    for j:=com to fim do
                      write(result,U2[i,j]:5:3,'  ');
                    writeln(result)
                  end;
                  writeln(result);writeln(result);
                  col:=col-lim;
                  com:=fim+1
                until col=0;
                writeln(result);writeln(result);
                writeln(result,'Valor do indicador F:',F[nc]:5:3);
                writeln(result);
                writeln(result,'Valor do indicador H:',H[nc]:5:3);
                writeln(result);
                writeln(result,'Valor do indicador HN:',HN[nc]:5:3);
              end
            until c=0;
            limpecran;
            writeln;writeln;writeln;writeln;
            writeln('               TABELA DOS COEFICIENTES DE VALIDADE');
            writeln;writeln;
            writeln('               c          F          H          HN');
            writeln;
            for j:=1 to nc do
              writeln('                ',VC[j],'        ',F[j]:5:3,'      '
5:3,'       ',HN[j]:5:3);
            writeln;writeln;writeln;
            write('Quer utilizar outros dados?(S/N)->');readln(rep)
  until (rep='N') or (rep='n')
end.
```

```
PROGRAM PROJ4 (input,output);

type   sa=string[12];
       ma=array[1..24,1..24] of double;

const  zr=0.00000000000000000001;
       eps=0.001;
       it0=5;
       cd=3;

var   i3,j3,i2,j2,p,it,l,col,com,fim,lim,iter,itermax,c,i,j,k,n,nz,dim,dim1,nc,k1,k
3,k4:integer;
      det,soma5,smv,media,prod,soma4,pi,sm,el1,y,el,uij,ui,soma1,soma2,soma3,m,tc:d
le;
       V,X,XI:array[1..50,1..24] of double;
       D;U1,U2:array[1..50,1..50] of double;
       dr,di,dados,result:text;
       s1,r,r1,t,t1:sa;
       s:string[85];
       arred,enc,nulo:boolean;
       tab,rep,op,seq,seg:char;
       A:array[1..100] of sa;
       Dis,Soma:array[1..24] of double;
       Si,Sinv,Anorm,MP:ma;
       H,F,HN:array[1..50] of double;
       VC:array[1..50] of integer;
       Elm,DV,DVN:array[1..24] of double;


FUNCTION NORMA(k,l:integer):double;
var  i,j:integer;
     soma:double;
     Dist,Dist1:array[1..24] of double;

begin
  for i:=1 to dim do
      Dist[i]:=X[k,i]-V[l,i];
  for j:=1 to dim do
  begin
    soma:=0;
    for i:=1 to dim do
      soma:=soma+(Dist[i]*Anorm[i,j]);
    Dist1[j]:=soma
  end;
  soma:=0;
  for i:=1 to dim do
      soma:=soma+(Dist1[i]*Dist[i]);
  NORMA:=sqrt(soma)
end;


FUNCTION POWER(a,b:double):double;

begin
  if a<0 then write('0 valor de a nao esta correcto');
  if a>0 then power:=exp(b*ln(a))
         else power:=0
end;


PROCEDURE limpecran;
var cod:char;

begin
  cod:=chr(27);
```

```
  write(cod,'[2J');
  write(cod,'[H')
end;


PROCEDURE gotoxy(x,y:integer);
var cod:char;

begin
  cod:=chr(27);
  write(cod,'[H');
  write(cod,'[',y,'B');
  write(cod,'[',x,'C')
end;


PROCEDURE Crianome;
var m1,n1,n3:integer;
    st,st1:sa;
    B1:array[1..8] of integer;

PROCEDURE Indiv(num:integer;var i:integer);
var quoc,rest:integer;

begin
  quoc:=num div 10;
  i:=0;
  while quoc<>0 do
  begin
     rest:=num mod 10;
     num:=quoc;
     i:=i+1;
     B1[i]:=rest;
     quoc:=num div 10
  end;
  i:=i+1;
  B1[i]:=num
end;

PROCEDURE Convert(n2:integer;var s2:sa);
var j:integer;
    q:string[1];

begin
  s2:='';q:=' ';
  for j:=n2 downto 1 do
  begin
     q[1]:=chr(48+B1[j]);
     s2:=concat(s2,q)
  end
end;

begin
   m1:=trunc(m*100);
   Indiv(c,n3);
   Convert(n3,st);
   Indiv(m1,n1);
   Convert(n1,st1);
   r:=concat(t,st,st1)
end;


PROCEDURE Lertab;
var i,j:integer;

begin
```

```
   for i:=1 to n do
   begin
      readln(dados,s);
      for j:=1 to dim1 do
        read(dados,XI[i,j]);
      readln(dados,s);readln(dados,s)
   end
end;

PROCEDURE Lertab1;
var i,j:integer;
    z:double;

begin
  for i:=1 to n do
  begin
    k3:=0;
    for j:=1 to dim1 do
    begin
      if j=Elm[k3+1] then
                      begin
                        k3:=k3+1;
                        X[i,k3]:=XI[i,j]
                      end
    end;
    readln(dados,s);readln(dados,s)
  end
end;


PROCEDURE Arredondar;
var i,j:integer;

begin
  for i:=1 to c-1 do
    for j:=1 to n do
    begin
      el:=U2[i,j];
      el1:=round(el*y);
      el:=el1/y;
      U2[i,j]:=el
    end;
  for j:=1 to n do
  begin
    sm:=U2[1,j];
    for i:=2 to c-1 do
      sm:=sm+U2[i,j];
    U2[c,j]:=1.0-sm
  end
end;

                      {* PROGRAMA PRINCIPAL *}


begin
   repeat
      reset(di,'FICHDADOS');
      readln(di,l);
      if l<>0 then begin
                     col:=1;
                     com:=1;
                     lim:=15;
                     repeat
                       limpecran;
                       if col<lim then lim:=col;
                       fim:=com+lim-1;
```

```
                    writeln;writeln;
                    writeln('                    FICHEIROS DE DADOS EXISTENTES');
                    writeln;
                    for i:=com to fim do
                    begin
                      readln(di,t1);
                      writeln('                                        ',t1)
                    end;
                    col:=col-lim;
                    com:=fim+1;
                    repeat
                      gotoxy(60,24);
                      write('CONTINUE->RETURN');
                      readln(tab)
                    until tab=chr(32)
                  until col=0
                end;
        close(di);
        limpecran;
        write('Qual o ficheiro de dados a utilizar? ');
        readln(t);
        reset(dados,t);readln(dados,s);readln(dados,s);
        readln(dados,s);readln(dados,s);
        readln(dados,n);readln(dados,s);
        readln(dados,s);readln(dados,dim1);readln(dados,s);
        Lertab;
                    {*** CALCULO DESVIOS PADROES ***}

        for j:=1 to dim1 do
        begin
          smv:=0;
          soma5:=0;
          for i:=1 to n do
            soma5:=soma5+XI[i,j];
          media:=soma5/n;
          for k4:=1 to n do
            smv:=smv+(XI[k4,j]-media)*(XI[k4,j]-media);
          DV[j]:=sqrt(smv/n);
          DVN[j]:=DV[j]/media
        end;
        writeln('          DESVIOS PADROES COORDENADAS');
        for j:=1 to dim1 do
          writeln('Desvio da coordenada ',j,':',DV[j]);
        writeln;writeln;
        writeln('          DESVIOS PADROES NORMALIZADOS');
        for j:=1 to dim1 do
          writeln('Desvio da coordenada ',j,':',DVN[j]);
        write('Quantas coordenadas pretende utilizar? ');readln(dim);
        for i:=1 to dim do
        begin
          write('Qual a coordenada ',i,'? ');readln(Elm[i])
        end;
        Lertab1;
        nc:=0;
        repeat
          write('Em quantos grupos pretende dividir o conjunto? ');readln(c);
          if c<>0 then begin
                        write('Qual o valor a atribuir ao parametro m? ');readln(m);
                        write('Qual o numero maximo de iteracoes que pretende? ');re
termax);
                        Crianome;
                        reset(dr,'FICHRESULT');
                        readln(dr,p);
                        if p<>0 then
                                  for i:=1 to p do
                                  begin
```

```
            readln(dr,r1);
            A[i]:=r1
          end;
close(dr);
enc:=false;i:=1;
while(i<=p) and not enc do
begin
  if r=A[i] then enc:=true;
  i:=i+1
end;
if not enc then
begin
  p:=p+1;
  rewrite(dr,'FICHRESULT');
  writeln(dr,p);
  if p=1 then writeln(dr,r)
         else
         begin
           for i:=1 to p-1 do
           begin
             s1:=A[i];
             writeln(dr,s1)
           end;
           writeln(dr,r)
         end;
  close(dr)
end;

    {*  INICIALIZACAO DA MATRIZ U  *}

for i:=1 to c do
  for j:=1 to n do U1[i,j]:=0;
i:=1;
repeat
  j:=i;
  repeat
    U1[i,j]:=1;
    j:=j+c
  until j>n;
  i:=i+1
until i>c;

iter:=0;it:=it0;y:=round(exp(cd*ln(10)));
repeat
  arred:=false;
  iter:=iter+1;

    {*  CALCULO DOS CENTROIDES  *}

  for i:=1 to c do
  begin
    soma3:=0;
    for k1:=1 to dim do Soma[k1]:=0;
    for j:=1 to n do
    begin
      uij:=POWER(U1[i,j],m);
      for k1:=1 to dim do
        Soma[k1]:=Soma[k1]+uij*X[j,k1];
      soma3:=soma3+uij
    end;
    for k1:=1 to dim do
      V[i,k1]:=Soma[k1]/soma3
  end;

    {*  CALCULO DA MATRIZ U  *}
```

```
for i:=1 to c do
begin
  for i2:=1 to dim do
    for j2:=1 to dim do
      Si[i2,j2]:=0;
  for k:=1 to n do
  begin
    uij:=POWER(U1[i,k],m);
    for j2:=1 to dim do
      Dis[j2]:=X[k,j2]-V[i,j2];
    for i2:=1 to dim do
      for j2:=1 to dim do
        Si[i2,j2]:=Si[i2,j2]+Dis[i2]*Dis[j2]*uij
  end;
  {writeln('          ITERACAO ',iter)};
   {writeln('      MATRIZ S');
  writeln;writeln;
  col:=dim;
  com:=1;
  lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Si[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for i2:=1 to dim do
    for j2:=1 to dim do
    begin
      if i2<>j2 then Si[i2,j2]:=0
    end;
  {writeln('     MATRIZ S DIAGONAL');
  writeln;writeln;
  col:=dim;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Si[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for i2:=1 to dim do
    for j2:=1 to dim do
    begin
      if i2=j2 then Sinv[i2,j2]:=1/Si[i2,j2]
               else Sinv[i2,j2]:=0
    end;
  {writeln('          INVERSA DE S');
  writeln;
  col:=dim;
  com:=1;
  lim:=3;
  repeat
```

```
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Sinv[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  det:=1;
  for i2:=1 to dim do
      det:=det*Si[i2,i2];
  {writeln('DETERMINANTE DE S=',det);
  writeln;writeln;}
  for i2:=1 to dim do
    for j2:=1 to dim do
    begin
      pi:=POWER(det,1/dim);
      Anorm[i2,j2]:=pi*Sinv[i2,j2]
    end;
  {writeln('       MATRIZ NORMA');
  writeln;
  col:=dim;com:=1;lim:=3;
  repeat
    if col<lim then lim:=col;
    fim:=com+lim-1;
    for i2:=1 to dim do
    begin
      for j2:=com to fim do
        write(Anorm[i2,j2]);
      writeln
    end;
    writeln;writeln;
    col:=col-lim;
    com:=fim+1
  until col=0;}
  for j:=1 to n do D[i,j]:=NORMA(j,i)
end;
for j:=1 to n do
begin
  nz:=0;
  for i:=1 to c do
    if D[i,j]<zr then nz:=nz+1;
  if nz>0 then
          begin
            for i:=1 to c do
              if D[i,j]>zr then U2[i,j]:=0
                           else U2[i,j]:=1/nz
          end
          else
          begin
            for i:=1 to c do
            begin
              ui:=0;
              for k:=1 to c do
                ui:=ui+POWER(D[i,j]/D[k,j],2/(m-1));
              U2[i,j]:=1/ui
            end
          end
end;
{writeln('       MATRIZ DE PERTENCA');writeln;
col:=n;com:=1;lim:=3;
repeat
```

```
      if col<lim then lim:=col;
      fim:=com+lim-1;
      for i:=1 to c do
      begin
        for j:=com to fim do
          write(U2[i,j]);
        writeln
      end;
      writeln;
      col:=col-lim;
      com:=fim+1
    until col=0;}

        {*  TESTE DE CONVERGENCIA  *}

    {for i:=1 to c do
    begin
      for j:=1 to n do
      begin
        tc:=abs(U2[i,j]-U1[i,j]);
        writeln('tc(',i,j,')=',tc)
      end
    end;}
    nulo:=true;
    i:=1;
    while(i<=c) and nulo do
    begin
      j:=1;
      while(j<=n) and nulo do
      begin
        if abs (U2[i,j]-U1[i,j])>eps then nulo:=false;
        j:=j+1
      end;
      i:=i+1
    end;
    if nulo=false then
                    for i:=1 to c do
                      for j:=1 to n do U1[i,j]:=U2[i,j];
    if iter=it then
               begin
                 it:=it+it0;
                 Arredondar;
                 arred:=true
               end
until nulo or (iter=itermax);
if not arred then Arredondar;

    {*  CALCULO DOS COEFICIENTES DE VALIDADE  *}

soma1:=0;
soma2:=0;
for i:=1 to c do
  for j:=1 to n do
  begin
    soma1:=soma1+(U2[i,j]*U2[i,j]/n);
    if U2[i,j]<>0 then soma2:=soma2+U2[i,j]*ln(U2[i,j])/n
  end;
nc:=nc+1;
F[nc]:=soma1;
H[nc]:=-soma2;
HN[nc]:=H[nc]/(1-c/n);
VC[nc]:=c;

    {*  APRESENTACAO DE RESULTADOS  *}

col:=n;
```

```
com:=1;
lim:=10;
repeat
  limpecran;
  if col<lim then lim:=col;
  fim:=com+lim-1;
  if nulo=true then writeln('CONVERGENCIA ATINGIDA');
  writeln('NUMERO DE ITERACOES EXECUTADAS:',iter);
  writeln;
  writeln('              ***  COORDENADAS DOS CENTROIDES  ***

  writeln;
  for i:=1 to c do
  begin
    writeln('Coordenadas do centroide do grupo',i);
    for k1:=1 to dim do write(V[i,k1]:5:3,'  ');
    writeln;writeln
  end;
  writeln('               ***  MATRIZ U  ***');
  writeln;
  for i:=1 to c do
  begin
    for j:=com to fim do
      write(U2[i,j]:5:3,'  ');
    writeln
  end;
  col:=col-lim;
  com:=fim+1;
  repeat
    gotoxy(60,24);
    write('CONTINUE->RETURN');
    readln(seq)
  until seq=chr(32)
until col=0;
limpecran;
writeln;writeln;writeln;writeln;writeln;writeln;
writeln('              Valor do indicador F');
writeln('              F(',c,')=',F[nc]:5:3);
writeln;
writeln('              Valor do indicador H');
writeln('              H(',c,')=',H[nc]:5:3);
writeln;
writeln('              Valor do indicador HN');
writeln('              HN(',c,')=',HN[nc]:5:3);
repeat
  gotoxy(60,24);
  write('CONTINUE->RETURN');
  readln(seg)
until seg=chr(32);
rewrite(result,r);
writeln(result,'RESULTADOS OBTIDOS PARA O FICHEIRO ',t);
writeln(result);writeln(result);writeln(result);
writeln(result,'Numero de grupos em que foi dividido o conju
,c);

writeln(result,'Valor do parametro m: ',m:4:2);
writeln(result,'Numero de iteracoes executadas: ',iter);
writeln(result);
writeln(result,'          *** COORDENADAS DOS CENTROIDES  **

writeln(result);
for i:=1 to c do
begin
  writeln(result,'Coordenadas do centroide do grupo',i);
  for k1:=1 to dim do
    writeln(result,V[i,k1]:5:3,'  ');
  writeln(result)
```

```
                    end;
                    writeln(result,'              **  MATRIZ U  **');
                    writeln(result);
                    col:=n;
                    com:=1;
                    lim:=10;
                    repeat
                      if col<lim then lim:=col;
                      fim:=com+lim-1;
                      for i:=1 to c do
                      begin
                        for j:=com to fim do
                          write(result,U2[i,j]:5:3,'  ');
                        writeln(result)
                      end;
                      writeln(result);writeln(result);
                      col:=col-lim;
                      com:=fim+1
                    until col=0;
                    writeln(result);writeln(result);
                    writeln(result,'Valor do indicador F:',F[nc]:5:3);
                    writeln(result);
                    writeln(result,'Valor do indicador H:',H[nc]:5:3);
                    writeln(result);
                    writeln(result,'Valor do indicador HN:',HN[nc]:5:3);
                  end
                until c=0;
                limpecran;
                writeln;writeln;writeln;writeln;
                writeln('                TABELA DOS COEFICIENTES DE VALIDADE');
                writeln;writeln;
                writeln('                c           F          H           HN');
                writeln;
                for j:=1 to nc do
                  writeln('                ',VC[j],'        ',F[j]:5:3,'      '
]:5:3,'        ',HN[j]:5:3);
                writeln;writeln;writeln;
                write('Quer utilizar outros dados?(S/N)->');readln(rep)
  until (rep='N') or (rep='n')
end.
```