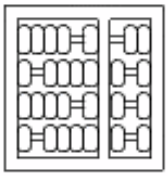


INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Aspectos básicos de *clustering*: conceitos e técnicas

Ricardo Luís Lachi
Heloísa Vieira da Rocha

Technical Report - IC-05-003 - Relatório Técnico

February - 2005 - Fevereiro

Aspectos básicos de *clustering*: conceitos e técnicas

Ricardo Luís Lachi
ricardo.lachi@ic.unicamp.br

Heloísa Vieira da Rocha
heloisa@ic.unicamp.br

Núcleo de Informática Aplicada à Educação (Nied)
Instituto de Computação – Universidade Estadual de Campinas
CEP: 13083-970 – Caixa Postal 6176 – Campinas – SP – Brasil
Telefone: +55 (19) 3788-5866

Resumo

Clustering é uma forma de organizar dados por meio do agrupamento destes em conjuntos, a partir da maior similaridade existente entre os dados de um mesmo conjunto que os de outro, com base em algum critério pré-determinado. Neste relatório são apresentados os aspectos gerais que devem ser observados quando se pretende aplicar alguma técnica de *clustering* na resolução de um problema. Os aspectos gerais apresentados são: definição da forma de representação do conjunto de dados a serem agrupados, definição de uma medida adequada de semelhança entre os dados e a definição de qual técnica de *clustering* utilizar para a construção dos *clusters*. Além disso, uma farta quantidade de referências bibliográficas é disponibilizada permitindo ao leitor o aprofundamento em todos os tópicos presentes no texto.

Palavras-chave: *Clustering*, formas de representação de dados, medidas de similaridade, técnicas de *clustering* hierárquicas e particionais.

Abstract

Clustering is a kind of data organization that groups data into sets where elements of a set are more similar to each other than are similar to the elements of another set, according to some criteria. In this report general aspects are presented that should be observed when it is intended to apply some clustering technique to solve a problem. The general aspects presented are: definition of the data representation type among data to be grouped, definition of the most suitable similarity measure among data and definition of which clustering technique to use to build the clusters. Besides that, a lot of bibliographic references are available to readers in order to make possible deep studies about all topics presented in the text.

Keywords: Clustering, types of data representation, similarity measures, hierarchical and partitional clustering techniques.

1. Introdução

Este relatório surgiu como resultado dos estudos realizados para a elaboração de uma proposta de tese de doutorado. Nele foi feito um primeiro estudo sobre *clustering* visando conhecer os principais aspectos e conceitos envolvidos.

Apesar do texto não se deter em apresentar um absoluto formalismo matemático, ele não evita de apresentar um certo rigor quando se faz necessário. Por exemplo, quando da apresentação da fórmula algébrica da função erro-quadrado, a notação utilizada está absolutamente dentro do mais puro formalismo algébrico.

No entanto, visando escrever um texto de fácil compreensão, procurou-se sempre apresentar as fórmulas matemáticas em conjunto com uma descrição pormenorizada dos seus significados e de figuras contendo a representação visual de cada uma delas.A partir disso, este relatório pode ser considerado como um bom ponto de partida para uma pessoa que esteja tomando contato pela primeira vez com o tema *clustering*.

Além disso, uma farta quantidade de referências bibliográficas é disponibilizada, possibilitando ao leitor o aprofundamento em todos os tópicos discutidos no texto.

Neste relatório são apresentados os aspectos gerais que devem ser observados quando se pretende aplicar alguma técnica de *clustering* na resolução de um problema. Os aspectos gerais apresentados são: definição da forma de representação do conjunto de dados a serem agrupados, definição de uma medida adequada de semelhança entre os dados e a definição de qual técnica de *clustering* utilizar para a construção dos *cluster*s.

Este texto está organizado da seguinte forma: na seção 2 é apresentada uma definição geral sobre *clustering*; na seção 3 são apresentadas as principais definições e notações utilizadas neste texto; na seção 4 são apresentados os problemas envolvidos na representação dos dados em padrões visando alcalçar um bom *clustering*; na seção 5 são apresentadas as medidas de similaridades existentes na literatura; na seção 6 são apresentadas diversas técnicas de *clustering*; na seção 7 são feitas algumas considerações finais; na seção 8 são apresentadas as referências bibliográficas.

2. Clustering

Clustering, ou agrupamento em português, é uma forma de modelagem de dados que se baseia na construção de *clusters*. *Clusters* são conjuntos de dados que exibem a seguinte propriedade: os elementos pertencentes a um mesmo conjunto apresentam maior semelhança entre si que os elementos pertencentes a qualquer outro conjunto, com relação a um certo critério de similaridade.

Na **Figura 1** é apresentado um exemplo ilustrativo de um *clustering*. Na parte **(a)** é apresentado um conjunto de dados de entrada, onde cada elemento é representado pelo símbolo 'x'. Na parte **(b)** é apresentado o resultado do *clustering* realizado sobre esse conjunto de dados de entrada, com cada elemento 'x' sendo rotulado com um número identificador do conjunto a que pertence ao final do *clustering*.

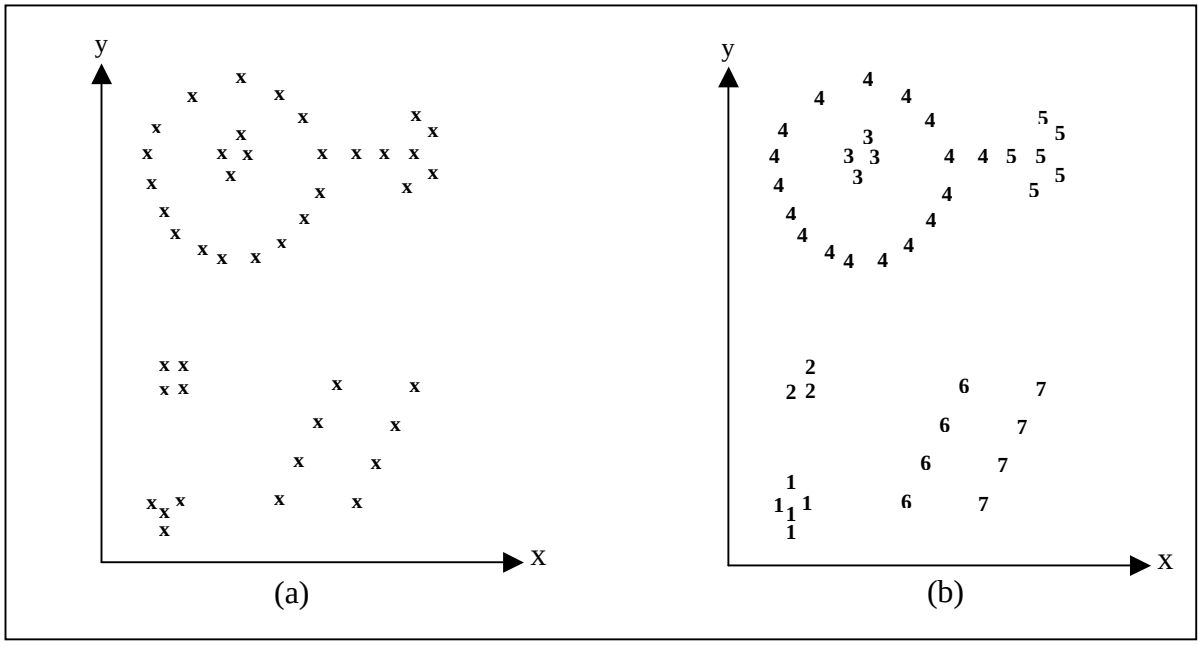


Figura 1 - Os dados de entradas são apresentados na parte (a) e os 7 *clusters* construídos são apresentados na parte (b). Na parte (b), pode-se ver que os dados pertencentes a um mesmo *cluster* apresentam o mesmo rótulo (Jain et al., 1999, p.266).

De acordo com Jain and Dubes (1988[1] apud Jain et al. 1999), são em número de 5, os passos que devem ser observados para a utilização de alguma técnica de *clustering*:

1. Definição da forma de representação dos dados de entrada a serem agrupados;

2. Definição de uma medida adequada de aproximação entre os dados;

3. Definir qual a técnica de *clustering* a ser aplicada para a construção dos *cluster*s;

4. Definição de uma *abstração dos dados*. Este passo envolve a definição de uma forma de abstração dos dados que permita uma representação simples e compacta do conjunto de dados. Esta simplicidade deve se dar tanto do ponto de vista humano quanto da máquina. Do ponto de vista humano, a representação deve ser intuitiva e fácil de ser compreendida. Do ponto de vista da máquina, essa abstração dos dados deve permitir um processamento posterior eficiente. Na concepção dos autores, este é um passo opcional e não, necessariamente, precisa ser pensado e executado quando se pretende aplicar alguma técnica de *clustering* sobre um conjunto de dados. O seu planejamento vai depender diretamente da aplicação a ser desenvolvida.;

5. Avaliação do resultado do *clustering*. O que se procura neste passo é, basicamente, avaliar quão bom foi o agrupamento dos dados conseguido pela técnica de *clustering* aplicada. Um problema levantado neste passo é que, geralmente e infelizmente, a análise da otimalidade do *clustering* se baseia em algum critério específico definido subjetivamente. Talvez, devido a essa subjetividade, este passo também seja considerado opcional pelo autores, à semelhança do passo anterior.

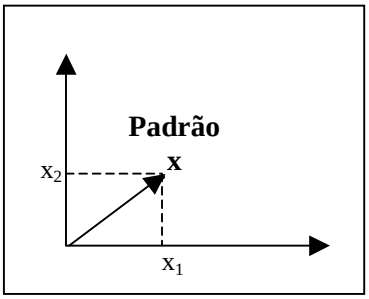
Os três primeiros passos apresentados serão discutidos nas seções 3, 4 e 5 Já os outros dois passos não serão discutidos pelos seguintes motivos: serem considerados opcionais pelos autores (ambos os passos), transcender o escopo deste texto (passo 4), subjetividade e inexistência de resultados consolidados a respeito (passo 5).

Antes da discussão sobre os passos 3, 4 e 5, é feita na próxima seção (seção 2) uma apresentação rápida sobre os termos e definições adotados neste texto, em virtude da grande diversidade de notações e definições existentes nas mais diversas comunidades que utilizam *clustering*.

[1] Jain, A. K., Dubes, R. C. *Algorithms for Clustering Data*. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ. 1988.

3. Definições e notações

A primeira definição a ser dada é a definição do termo *padrão*[2]. Um padrão **x** nada mais é do que um dado simples usado pelos algoritmos de *clustering*. Geralmente, é representado por um vetor com *d* características: $\mathbf{x} = (x_1, x_2, ..., x_d)$. No **Exemplo 1** a seguir é apresentado um esquema de um padrão **x** representado por um vetor com duas características: x_1 e x_2.



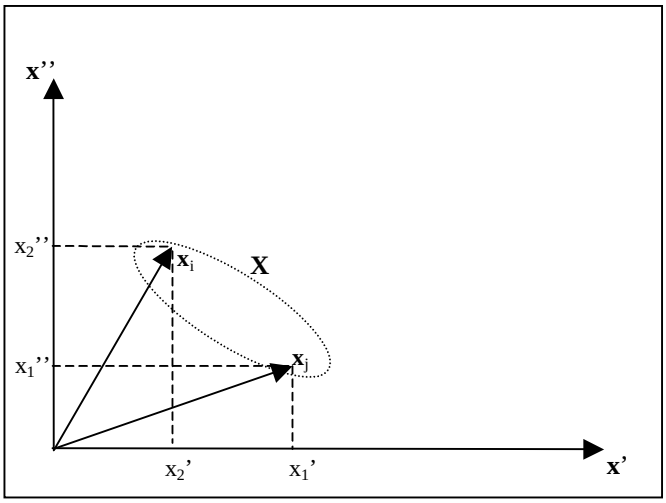
Exemplo 1. Padrão $\mathbf{x} = (x_1, x_2)$.

Uma observação importante que deve ser feita é que até este momento foi empregado no texto o termo *dado* no lugar do termo *padrão*. Isso foi feito para facilitar o entendimento inicial desse termo, deixando a apresentação da definição formal deste termo em um momento mais oportuno (esta seção). No entanto, deste ponto em diante no texto, será utilizado o termo *padrão* ao invés do termo *dado*, uma vez que é essa é a forma mais comumente utilizada na área de *clustering*.

Completando a definição do termo *padrão*: cada componente escalar, x_i, presente no *vetor de características* do padrão **x** é denominado característica *i* do padrão **x**. Essa componente também pode ser denominada como sendo um *atributo i* do padrão **x**. Além disso, o valor *d* presente na definição de **x** é denominada a *dimensão* do padrão e representa o número de características que o padrão **x** possui. No **Exemplo 1** apresentado antes, o padrão **x** é um exemplo de um padrão bidimensional, pois é representado por um vetor contendo duas características.

Se existir um *conjunto de padrões*, este será denotado por **X**. Logo, tem-se que $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ onde, cada padrão $\mathbf{x_i}$ pertencente a este conjunto, representa, por sua vez, um vetor de características $\mathbf{x_i} = (x_1, x_2, ..., x_d)$ de dimensão *d*. Conseqüentemente, o conjunto de padrões **X** pode ser visualizado como uma matriz *n x d*. No **Exemplo 2** a seguir é apresentado um esquema ilustrativo dessa situação.

[2] *Pattern* no original em inglês.



Exemplo 2. São apresentados dois padrões: $\mathbf{x}_i$ e $\mathbf{x}j$. O padrão $\mathbf{x}_i$ possui as características x_2' e x_2''; e o padrão x_j possui as características x_1' e x_1''. A união dos dois padrões $\mathbf{x}_i$ e $\mathbf{x}_j$ forma o conjunto de padrões **X**.

Existem dois modos de particionamento dos padrões em *clusters*, um denominado *hard* e outro *fuzzy*. As técnicas de *clustering* que particionam os padrões de forma *hard*[3] atribuem para cada padrão $\mathbf{x_i}$ uma classe denominada l_i. O conjunto de todas as classes existentes para um determinado *conjunto de padrões* **X** é representada por $\mathbf{L} = \{l_1, ...l_n\}$. onde cada elemento l_i de **L** assume um dos valores do conjunto: {1, ..., *k*}, onde *k* é o número de *clusters* gerado pela técnica de *clustering hard* empregada. Já as técnicas de *clustering* do tipo *fuzzy*[4] atribuem a cada padrão $\mathbf{x_i}$, um valor f_{ij}, representando o grau de do aptidão do padrão $\mathbf{x_i}$ ao *cluster j*.

Com isso, foram apresentados os principais termos e notações que são empregados neste texto. Caso haja necessidade de alguma outra notação, a mesma será explicada diretamente no próprio local em que for empregada.

Na próxima seção são abordados os problemas envolvidos na representação de padrões.

---

[3] No modo *hard* os padrões são particionados em grupos onde, cada dado, pertence *exclusivamente* a um único *cluster*.

[4] No modo *fuzzy*, cada padrão tem associado a si um certo grau de *aptidão* para pertencer a um certo grupo, isto é, um padrão *x* pode pertencer com uma probabilidade de 53% a um *cluster A* e de 47% a um *cluster B.*

4. Representação de padrões

Infelizmente, não há nenhuma regra geral que possa ser seguida e que permita a um usuário definir os padrões e características mais apropriadas de serem utilizadas para a sua situação específica.

Tendo isso em vista, pode-se dizer que o usuário[5] tem um papel fundamental a desempenhar, pois é sua função primordial coletar o maior número possível de fatos a respeito dos dados de entrada e fazer uma cuidadosa investigação sobre as características disponíveis visando conseguir um bom *clustering*.

Concluindo, uma má definição da forma como cada padrão será representado pode levar à construção de um *clustering* complexo, isto é, pode levar ao agrupamento dos dados de entrada de um modo que não permite uma fácil visualização do porquê deles terem sido agrupados de tal modo. Na **Figura 2** é mostrado um exemplo simples dessa dificuldade.

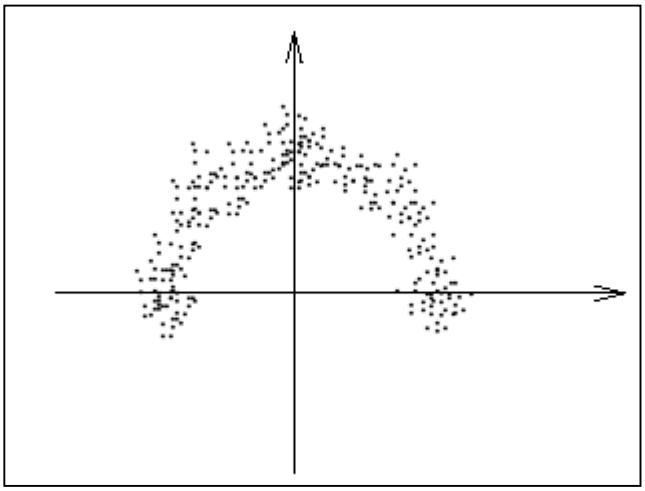


Figura 2 – Um *cluster* curvilíneo cujos pontos são praticamente eqüidistantes da origem (Jain et al., 1999, p.271).

Na **Figura 2**, os pontos no espaço (dados de entrada) podem ser facilmente visualizáveis como sendo um conjunto de pontos que se encontram a uma distância aproximadamente igual à origem sendo possível, conseqüentemente, agrupá-los em um único *cluster* curvilíneo. No entanto, se alguém escolher representar os padrões por coordenadas cartesianas, provavelmente, muitos algoritmos de *clustering* vão fragmentar o *cluster* em dois ou mais *clusters* (um *cluster* para cada quadrante, por exemplo). Agora, se forem usadas coordenadas polares, provavelmente, uma solução com um só *cluster* será encontrada. Na **Figura 3** é apresentado um esquema ilustrando essas duas representações.

[5] Pessoa responsável pelo desenvolvimento da aplicação que utilizará técnicas de *clustering*.

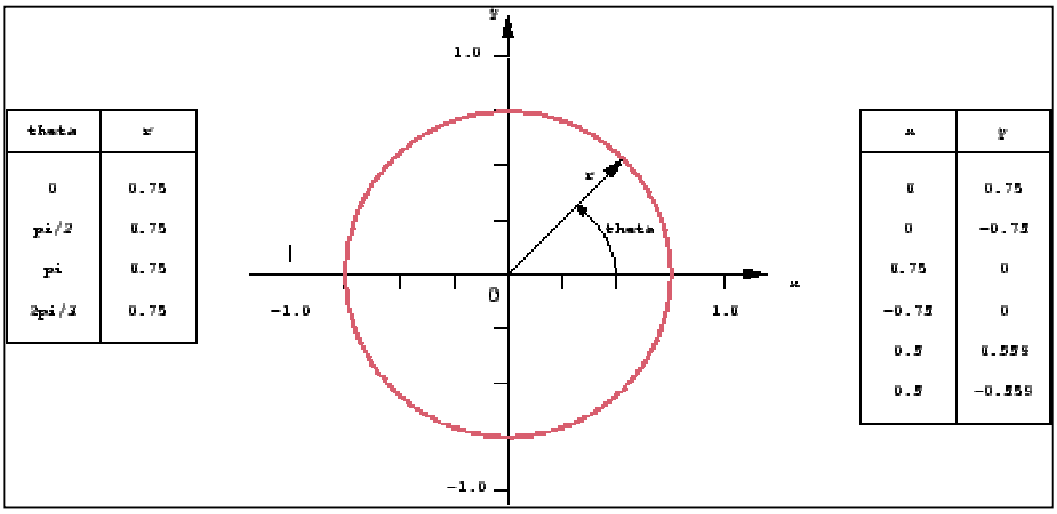


Figura 2 – Em coordenadas polares, cada padrão é representado por dois valores: sua distância a partir da origem e pelo ângulo *theta*. Em coordenadas cartesianas cada padrão é representado por dois valores: x e y.

Um *padrão* pode medir tanto um objeto físico (exemplo, um computador) quanto um conceito abstrato (exemplo, tipo de sorriso). Como já foi apresentado, um padrão **x** é representado convencionalmente por vetores multidimensionais, onde cada dimensão (componente x_i) representa uma única característica do padrão (Duda e Hart, 1973[6] apud Jain et al., 1999). Essa característica pode ser qualitativa ou quantitativa. Por exemplo, se *altura* e *idade* forem duas das características usadas então, (1.75, 18) é a representação de uma pessoa com 1.75m de altura e 18 anos. Na **Figura 4** é apresentado como ficaria a representação esquemática dessa situação.

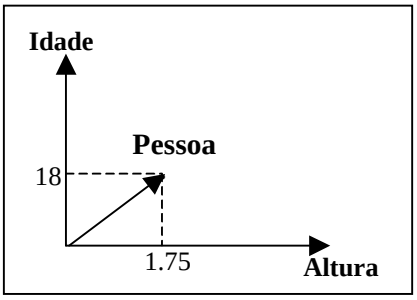


Figura 4. Pessoa = (altura, idade).

[6] Duda, R. O., Hart, P. E. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., New York, NY. 1973.

De acordo com Gowda e Diday (1992[7] apud Jain et al., 1999), as características podem ser subdivididas nas seguintes categorias:

1. Características quantitativas:
   a) **Valores contínuos** (exemplo: altura);
   b) **Valores discretos** (exemplo: número de computadores);
   c) **Intervalo de valores** (exemplo: duração de um evento).

2. Características qualitativas:
   a) **Nominal ou desordenada**: uma característica nominal é uma característica que pode assumir um número pequeno de valores possíveis. Trata-se de uma generalização dos atributos booleanos, onde o número de valores assumidos é dois. Um exemplo de atributo nominal seria "Cor", podendo assumir cinco valores: vermelho, amarelo, verde, azul e rosa. Em geral, seja M o número de valores que pode assumir um atributo nominal. Ao invés de denotar os valores por *strings* (vermelho, amarelo, etc), normalmente são associados a eles números inteiros 1, 2,..., M[8].

   b) **Ordinal**: uma característica ordinal é semelhante a uma característica nominal, exceto que os valores assumidos são ordenados, o que não acontece com as características nominais. Por exemplo, a característica "TipoMedalha" pode assumir os valores nominais Bronze, Prata e Ouro. A estes valores são associados os números 0, 1, 2 respectivamente. A ordem entre os números estabelece uma ordem entre os valores Bronze, Prata e Ouro.

Na **Figura 4**, foi construído um vetor com duas características quantitativas (*altura* e *idade*), mas isso não significa que não se possa utilizar um número maior de características abrangendo as diferentes categorias enumeradas. No entanto, uma ressalva deve ser feita, o projetista deve sempre procurar usar o mínimo possível de características necessárias para o seu problema, pois isso permitirá um *clustering* mais eficiente e passível de ser visualmente inspecionado por seres humanos.

Na próxima seção são apresentadas as principais medidas de similaridade existentes que permitem comparar dois padrões distintos.

---

[7] Gowda, K. C., Diday, E. Symbolic clustering using a new dissimilarity measure. *IEEE Trans. Syst. Man Cybern*. 22, 368-378. 1992.

[8] O fato de se ter associado números inteiros aos valores do atributo, não significa que uma ordem entre estes valores foi determinada. O objetivo desta associação é simplesmente de poder tratar valores nominais como sendo números inteiros. A ordem não é considerada. Esta é a diferença fundamental entre atributos nominais e atributos ordinais.

5. Medidas de similaridade

O conceito de similaridade é um conceito fundamental para a construção de um *cluster*, pois, se dois padrões são similares de acordo com algum critério utilizado pela técnica de *clustering* empregada, então serão agrupados em um mesmo *cluster*, caso contrário, serão agrupados em *clusters* diferentes. Por isso a definição de medidas que permitam comparar padrões pertencentes a um mesmo espaço de características é essencial para a maioria dos processos de *clustering*.

Nesta seção serão abordadas somente as medidas de similaridade entre características contínuas pois são as mais bem conhecidas e utilizadas. No entanto, há divesos trabalhos na literatura propondo medidas de similaridades envolvendo os outros tipos de características possíveis: Wilson e Martinez (1997) propuseram uma métrica relacionando características *contínuas* e *nominais*; Diday e Simon (1976) e Ichino e Yaguchi (1994) propuseram uma forma de computar similaridades entre padrões representados tanto por características quantitativas quanto qualitativas; Baeza-Yates (1992) descreve diversas medidas de similaridades entre *strings*; Zhang (1995) fez um bom resumo de medidas de similaridade entre árvores.

A métrica mais popular para calcular similaridades entre características contínuas é a *distância euclidiana*[9]:

$$d_2(\mathbf{x_i}, \mathbf{x_j}) = \|\Sigma(x_{i,k} - x_{j,k})\| \qquad \text{para } k = 1, ..., d$$

Onde,

d_2 = distância entre os padrões $\mathbf{x_i}$ e $\mathbf{x_j}$ no espaço de dimensão 2

x_{ik} = componente k do padrão $\mathbf{x_i}$

x_{jk} = componente k do padrão $\mathbf{x_j}$

Essa forma de calcular a similaridade entre padrões é bastante intuitiva e é largamente utilizada em espaços de dimensões 2 e 3. No entanto, essa métrica tem a tendência de fazer com que as características que tenham os maiores valores dominem o resultado[10]. Esse problema pode ser facilmente resolvido utilizando-se esquemas de normalização sobre os valores das características, ou então, aplicando pesos diferentes para cada uma das características.

---

[9] A distância euclidiana é um conceito matemático que representa a menor distância existente entre dois pontos na Geometria Euclidiana. Esta geometria foi construída pelo matemático grego Euclides (Ávila, 2001).

[10] Como exemplo veja o resultado da aplicação da fórmula de distância euclidiana sobre os seguintes padrões: $\mathbf{x_i} = (1000, 0.3)$ e $\mathbf{x_j} = (2000, 0.8)$.

$d_2(\mathbf{x_i}, \mathbf{x_j}) = ((1000 - 2000)^2 + (0.3 - 0.8)^2)^{1/2} = (1000^2 + 0.5^2)^{1/2} = 1000.000125$

Veja como o resultado da diferença entre os valores da componente 1 dos padrões, domina o resultado calculado para a diferença entre os valores da componente 2 dos padrões.

Um ponto importante a ser observado na aplicação dessa métrica é a necessidade de se calcular a diferença entre os *n* padrões existentes, dois a dois. Isso implica no cálculo de de $n(n-1)/2$ valores[11], por isso é muito útil e comum a utilização de uma matriz $n \times d$ simétrica[12] para o armazenamento do cálculo desses valores. Essa matriz é denominada *matriz de proximidade*, pois contém a menor distância entre todos os *n* padrões existentes.

Gowda e Krishna (1977); Jarvis e Patrick (1973) descrevem uma outra medida de similaridade que leva em consideração o efeito do pontos vizinhos a um certo ponto. Esses pontos vizinhos são denominados *contexto* por Michalski e Stepp (1983).

Nessa outra medida proposta, a similaridade entre dois padrões $\mathbf{x_i}$ e $\mathbf{x_j}$, é dada pela seguinte expressão:

$$s(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i, \mathbf{x}_j, \mathbf{E})$$

Onde,
E é o *contexto* (conjunto de padrões vizinhos).

Em outras palavras, a similaridade *s* a ser calculada entre os padrões $\mathbf{x}_i$ e $\mathbf{x}_j$, é função dos próprios padrões $\mathbf{x}_i$, $\mathbf{x}_j$ e de um contexto **E**. O contexto **E** de um padrão $\mathbf{x}_i$ é definido **sempre em relação a algum outro padrão** $\mathbf{x}_j$. Isso porque o contexto de um padrão $\mathbf{x}_i$ em relação a um padrão $\mathbf{x}_j$ é definido como sendo o conjunto de todos os padrões vizinhos a $\mathbf{x}_i$ que se encontram a uma distância euclidiana *menor ou igual* que a distância existente entre $\mathbf{x}_i$ e $\mathbf{x}_j$.

A métrica definida usando esse conceito de *contexto* é denominada *distância mútua de vizinhos* (MND) e é dada pela seguinte expressão:

$$MND(\mathbf{x}_i, \mathbf{x}_j) = NN(\mathbf{x}_i, \mathbf{x}_j) + NN(\mathbf{x}_j, \mathbf{x}_i)$$

Onde,
$NN(\mathbf{x_i}, \mathbf{x_j})$ é número de vizinhos de $\mathbf{x_j}$, cuja distância é menor ou igual à distância entre $\mathbf{x_i}$ e $\mathbf{x_j}$

Um exemplo da aplicação dessa medida de similaridade pode ser visto nas **Figura 5** e **Figura 6**.

---

[11] O cálculo a ser efetuado é uma *combinação* de *n* elementos, dois a dois. Aplicando a fórmula de combinação de termos: $C_{n,2} = n!/(2!\,(n-2)!\,)$, o resultado $n(n-1)/2$ é direto.

[12] A matriz é simétrica porque, para qualquer elemento $\mathbf{m_{i,j}}$ da matriz, tem-se que $\mathbf{m_{i,j}} = \mathbf{m_{j,i}}$, onde, $\mathbf{i} = 1, ..., n$ e $\mathbf{j} = 1, ..., d$.

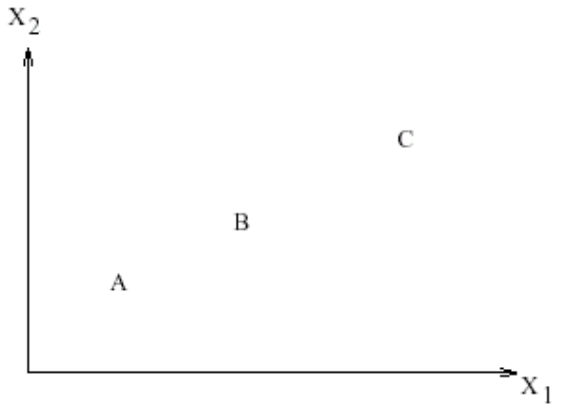


Figura 5 – A e B são mais similares que A e C (Jain et al., 1999, p.273).

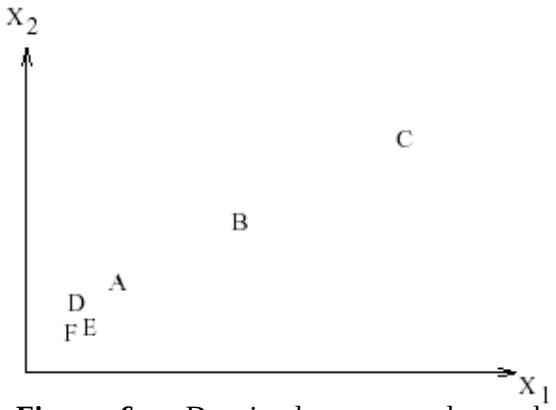


Figura 6 – Depois de uma mudança de contexto, B e C são mais similares que B e A. A e B são mais similares que A e C (Jain et al., 1999, p.273).

Na **Figura 5**, pode-se ver que o vizinho mais próximo de A é B, e que o vizinho mais próximo de C é B. Por isso, *NN*(**A**, **B**) = 1 e *NN*(**B**, **A**) = 1, logo *MND*(**A**, **B**) = 2. No entanto, em relação aos pontos B e C o resultado é um pouco diferente. *NN*(**B**, **C**) = 1, mas *NN*(**C**, **B**) = 2 (os vizinhos mais próximos de B em relação a C, são os pontos A e o próprio ponto C). Com isso, *MND*(**B**, **C**) = 3.

Na **Figura 6**, foram inseridos três novos pontos **D**, **E** e **F**. Com isso, tem-se que *MND*(**B**, **C**) = 3 (igual à **Figura 3**), mas *MND*(**A**, **B**) [13] = 5. Em outras palavras, o valor de *MND* entre os pontos **A** e **B** foi aumentado mesmo os pontos **A** e **B** não tendo sido movidos!

Pode-se concluir a partir desse fato observado no exemplo da **Figura 6**, que é possível tornar similares quaisquer dois padrões, simplesmente acrescentando a eles um número suficiente de características. Isso é um problema, pois pode acabar resultando na impossibilidade de diferenciação entre dois padrões que estejam em uma situação dessas, gerando a seguinte dúvida: a qual *cluster* pertencem esses padrões?

Uma proposta para superar essa dificuldade pode ser vista no trabalho de Michalski e Stepp (1983). Nesse trabalho, os autores propõem que uma forma de contornar a situação apresentada seria por meio do acréscimo de informações relativas ao domínio da aplicação de forma a ter como definir a similaridade ou não entre dois padrões nessa situação.

Na próxima seção são apresentadas as principais técnicas de *clustering* existentes, que utilizam alguma dessas medidas de similaridade apresentadas para a construção dos agrupamentos.

[13] *MND*(**A**, **B**) = *NN*(**A**, **B**) + *NN*(**B**, **A**) = 1 + 4 = 5

6. Técnicas de *Clustering*

Na literatura há diversas taxonomias de classificação das técnicas de *cl*ustering disponíveis. Nesta seção será apresentada a taxonomia apresentada em Jain et al. (1999), **com algumas pequenas alterações na mesma**[14], pois ela engloba os dois principais conjuntos de algoritmos de clustering existentes: **hierárquicos** e **particionais**. Na **Figura 7** é mostrado um esquema dessa taxonomia.

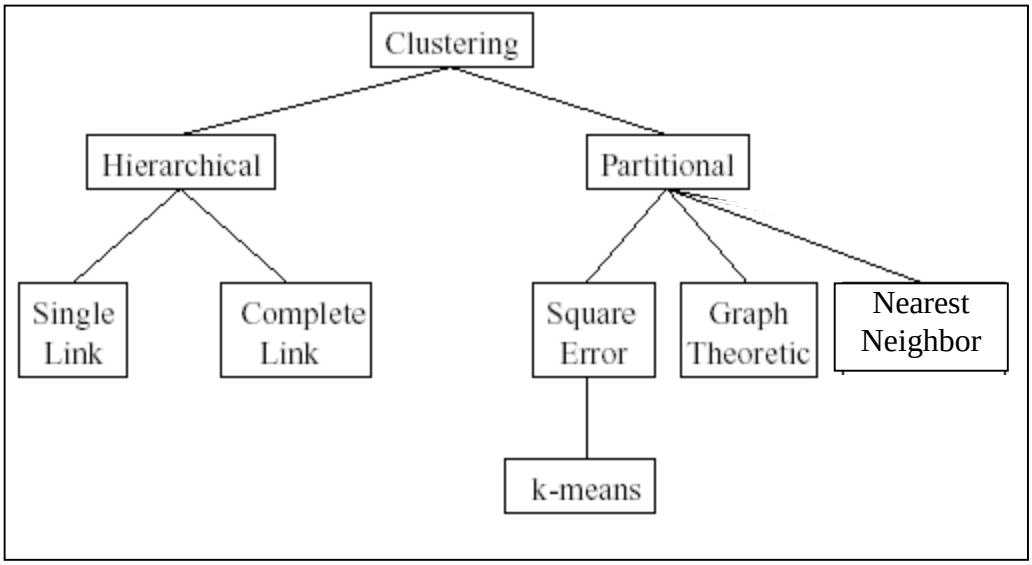


Figura 7 – Uma taxonomia para as diferentes técnicas de *clustering* disponíveis.

No topo dessa taxonomia proposta é feita a separação entre os dois principais ramos envolvendo as técnicas de *clustering*: ramo **hierárquico** e ramo **particional**.

Nas próximas subseções são apresentadas o conceito por trás dessas técnicas e os algoritmos de *clustering* derivados a partir delas.

6.1. Algoritmos de clustering hierárquico

Algoritmos de *clustering* hierárquicos constroem uma hierarquia de *clusters*, isto é, uma árvore de *clusters*, conhecida como **dendograma**. Nessa estrutura, cada *cluster* pode conter outros *clusters*, denominados **filhos**; se um *cluster* não tiver nenhum filho, ele é denominado uma **folha** do dendograma.

A maioria dos algoritmos de *clustering* hierárquicos são variações do algoritmo **single-link** (Sneath e Sokal, 1973), **complete-link** (King, 1967) e **minimum-variance** (Ward, 1963; Murtagh, 1984). Dentre esses os algoritmos **single-link** e **complete-link** são os mais populares, por isso eles serão detalhados nas subseções seguintes.

[14] Foi simplificada, em relação à figura original, o desmembramento dos algoritmos de *clustering* particionais. Na figura original, havia mais uma subdivisão do algoritmo *nearest neighbor* na taxonomia apresentada.

6.1.1. Algoritmo de clustering Single-link

Este é um algoritmo do tipo **aglomerativo**, isto é, inicia-se com todos os padrões sendo *clusters* individuais e, depois, vai-se, recursivamente, agrupando-os. A principal característica deste algoritmo é o fato dele calcular a distância entre 2 *clusters* como sendo a **menor** das distâncias existentes entre **todos** os pares de padrões pertencentes aos 2 *clusters* (1 padrão retirado de um dos *clusters* e 1 outro padrão retirado do outro *cluster*).

Esquematicamente, os passos para a aplicação deste algoritmo são:

- Defina cada padrão como sendo um *cluster*;
- Construa uma lista das distâncias entre padrões, para todos os pares de padrões;
- Ordene essa lista em ordem crescente;
- Percorra essa lista de distâncias do seguinte modo: para cada valor d_k da lista de distâncias, construa um grafo onde, os pares de padrões cujos valores são mais próximos de d_k são conectados por uma aresta;
- Se todos os padrões são membros de um grafo conexo, então, pare. Caso contrário repita todos esses passos.

Veja na **Figura 9** a operação deste algoritmo sobre os padrões da **Figura 8**.

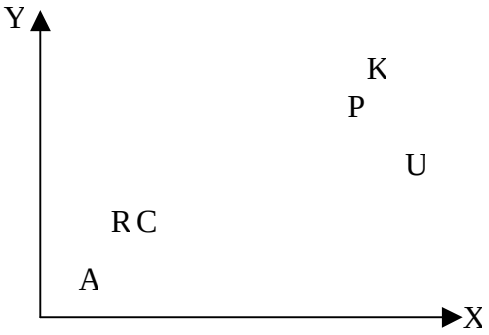


Figura 8 – Conjunto de padrões.

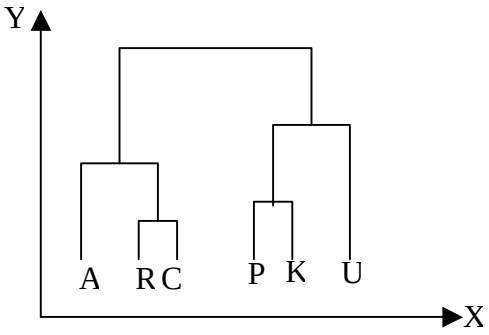


Figura 9 – O dendograma obtido usando o algoritmo **single-link**.

Detalhando o processo que ocorreu. Inicialmente, todos os padrões (**A**, **C**, **P**, **K**, **R**, **U**) viraram *clusters* individuais. Foi calculado a distância euclidiana para todos os pares de padrões: d(**R**, **C**) = 1; d(**P**, **K**) = 2; d(**A**, **R**) = 3, d(**A**, **C**) = 4; d(**U**, **P**) = 4; d(**U**, **K**) = 5; ... .[15]

[15] Além destas distâncias apresentadas também existem diversas outras distâncias, tais como, d(**A**, **P**), d(**R**, **U**), ... que não foram explicitadas para não tornar muito grande o exemplo. O que se deve notar é que todas essas outras distâncias omitidas são maiores que d(**U**, **K**)!

Essa lista foi ordenada em ordem crescente, ficando do seguinte modo: $d(\mathbf{R}, \mathbf{C}) = 1$; $d(\mathbf{P}, \mathbf{K}) = 2$; $d(\mathbf{A}, \mathbf{R}) = 3$, $d(\mathbf{A}, \mathbf{C}) = 4$; $d(\mathbf{U}, \mathbf{P}) = 4$; $d(\mathbf{U}, \mathbf{K}) = 5$; ...

Depois, percorre-se essa lista unindo-se na primeira interação os padrões, **R** e **C**, em um mesmo *cluster* (denominemos de cluster $\mathbf{C_1}$).

Novamente, se calcula a menor distância entre todos os *clusters* e se ordena a lista resultante: $d(\mathbf{P}, \mathbf{K}) = 2$; $d(\mathbf{A}, \mathbf{C_1}) = 3^{*}$; $d(\mathbf{U}, \mathbf{P}) = 4$; $d(\mathbf{U}, \mathbf{K}) = 5$; ...

Percorre-se novamente essa lista e se une os padrões **P** e **K**, em um mesmo *cluster* (denominemos de cluster $\mathbf{C_2}$).

Mais uma vez, se calcula a menor distância entre todos os *clusters* e se ordena a lista resultante: $d(\mathbf{A}, \mathbf{C_1}) = 3$; $d(\mathbf{U}, \mathbf{C_2}) = 4^{**}$; ...

Esse processo se repete até todos os padrões formarem um grafo conexo.

Na próxima subseção será apresentado um algoritmo semelhante a este só que, ao invés da distância entre 2 *clusters* quaisquer ser calculada como sendo o **valor mínimo** entre todas as distâncias de pares de padrões dos 2 *clusters*, será o **valor máximo**.

6.1.2. Algoritmo de clustering Complete-link

Este também é um algoritmo do tipo aglomerativo, no mesmo estilo do **single-link**, a diferença é que este algoritmo utiliza a **maior** das distâncias existentes em dois *clusters* no processo de concatenação de *clusters*.

Esquematicamente, os passos para a aplicação deste algoritmo são iguais ao anterior exceto pela ordenação da lista contendo a **maior distância** entre dois *clusters*:

- Defina cada padrão como sendo um *cluster*;
- Construa uma lista das distâncias entre padrões, para todos os pares de padrões;
- Ordene essa lista em ordem crescente;
- Percorra essa lista de distâncias do seguinte modo: para cada valor d_k da lista de distâncias, construa um grafo onde, os pares de padrões cujos valores são mais próximos de d_k são conectados por uma aresta;
- Se todos os padrões são membros de um grafo conexo, então, páre. Caso contrário repita todos esses passos.

---

[*] Perceba neste momento, que a distância calculada entre o *cluster* que contém o padrão **A** e o *cluster* $\mathbf{C_1}$, é o ***mínimo*** de todas as distâncias entre os dois *clusters*. Como $d(\mathbf{A}, \mathbf{R}) = 3$ e $d(\mathbf{A}, \mathbf{C}) = 4$, então $d(\mathbf{A}, \mathbf{C_1}) = 3$!

[**] Perceba neste momento, que a distância calculada entre o *cluster* que contém o padrão U e o *cluster* $\mathbf{C_2}$, é o ***mínimo*** de todas as distâncias entre os dois *clusters*. Como $d(\mathrm{U}, \mathbf{P}) = 4$ e $d(\mathbf{U}, \mathbf{K}) = 5$, então $d(\mathbf{U}, \mathbf{C_2}) = 4$.

Veja na **Figura 10** e na **Figura 11** um exemplo do resultado da aplicação dos algoritmos **single-link** e **complete-link** sobre o mesmo conjunto de padrões.

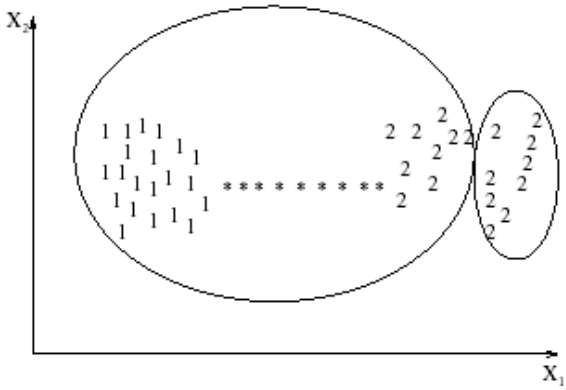


Figura 10 – Aplicação do algoritmo **single-link** construiu 2 *clusters* (1 e 2). Eles estão conectados por uma cadeia de ruído de padrões (símbolo *) (Jain et al., 1999, p.277).

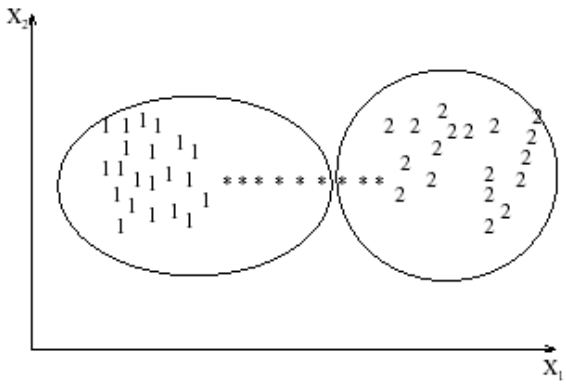


Figura 11 – A aplicação do algoritmo **complete-link** construiu 2 *clusters* (1 e 2). Eles estão conectados por uma cadeia de ruído de padrões (símbolo *)(Jain et al., 1999, p.277).

Concluindo, o algoritmo *complete-link* produz *clusters* mais compactos (Baeza-Yates, 1992), já o algoritmo *single-link* produz *clusters* mais alongados (Nagy, 1968). Mas, independente do formato final dos *clusters* construídos, foi observado por Jain e Dubes (1988) que o algoritmo *complete-Link* produz hierarquias mais úteis que o algoritmo *single-Link* e, portanto, é considerado o melhor algoritmo dos dois.

Na próxima seção são apresentados a outra classe de algoritmos existentes: algoritmos de *clustering* **particionais.**

6.2. Algoritmos de clustering particionais

Algoritmos de *clustering* particionais constroem uma partição simples dos padrões ao invés de uma estrutura de *clustering* como é feita, por exemplo, pelos algoritmos hierárquicos. Os métodos particionais apresentam vantagens nas aplicações que envolvem um grande número de conjuntos pois, nestes casos, a construção de um dendograma é computacionalmente proibitiva.

As técnicas particionais, normalmente, produzem *clusters* por meio da otimização de uma função. Nas próximas subseções são apresentadas algumas dessas técnicas particionais de *clustering*.

6.2.1. Algoritmo de clustering Square-error

A mais freqüente função utilizada para particionar um conjunto de padrões é a função **erro-quadrado**, pois ela tende a funcionar bem para *clusters* isolados[16] e compactos[17]. A função que é otimizada neste algoritmo é a função erro-quadrado *e* dada pela seguinte fórmula:

$$e^2(X,L)=\sum_{j=1}^{K}\sum_{i=1}^{n_j}\left\|x_i^{(j)}-c_j\right\|^2$$

Na equação, o valor $x_i^{(j)}$ é o i-ésimo padrão pertencente ao j-ésimo *cluster* e c_j é o centróide[18] do j-ésimo *cluster*. Além disso, ela considera como entrada um conjunto de padrões **X** e um número inteiro **K** representando o número de *clusters* que se deseja construir. A partir desses dados de entrada é construído um *clustering* **L**.

Explicando com maiores detalhes essa fórmula. Primeiramente, temos o cálculo do centróide de um *cluster* **j**. Uma vez calculado esse valor, calcula-se a distância entre todos os pontos desse *cluster* **j** (valor $x_i^{(j)}$) em relação ao centróide (c_j) e soma todas elas. Nas **Figuras 12** e **13** são apresentados esquema representando esquematicamente cada um desse cálculos.

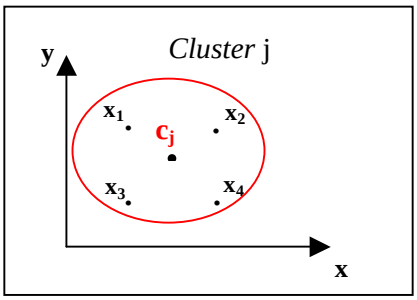


Figura 12. Centróide c_j do *cluster* j.

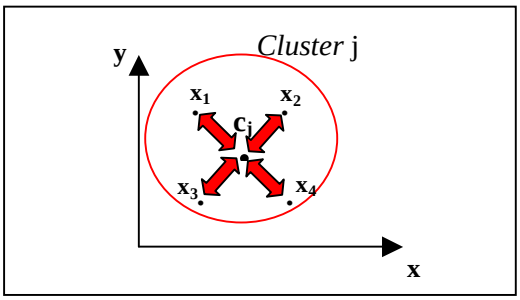


Figura 13. Estão destacadas todas as distâncias representadas pelo termo: $\sum_{i=1}^{n_j}\left\|x_i^{(j)}-c_j\right\|^2$.

Por fim, soma-se todos esses valores obtidos para todos os **K** *clusters* (veja **Figura 14**).

---

[16] Os padrões pertencentes a um *cluster* estão longe dos padrões pertencentes a outro *cluster*.

[17] Os padrões pertencentes a um *cluster* estão próximos entre si.

[18] Centróide é o ponto que representa o centro de todos os pontos pertencentes a um *cluster*.

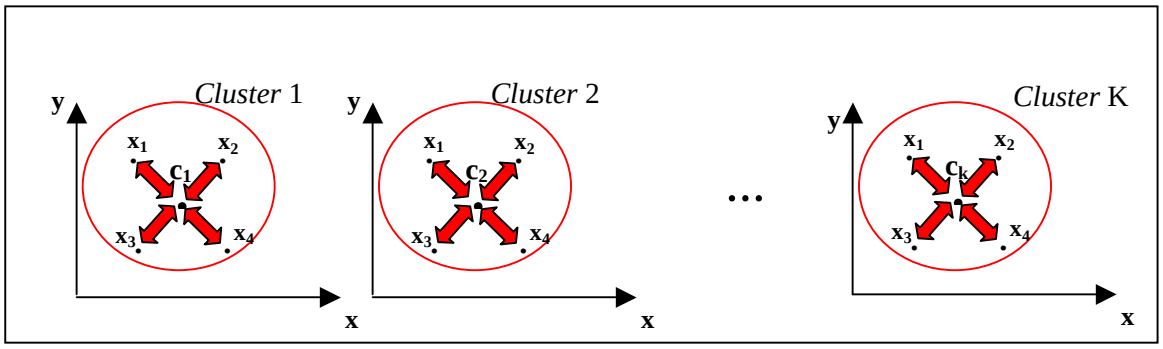


Figura 14. Estão destacadas todas as distâncias representadas pelo termo: $e^2(X, L) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2$.

A partir dessa fórmula é que se baseia o algoritmo de clustering denominado **k-means**. Esquematicamente, esse algoritmo funciona do seguinte modo:

- Escolhe *k* centros de *clusters* para coincidir com *k* padrões aleatoriamente escolhidos[19];
- Atribui cada padrão ao centro de *cluster* mais próximo dele;
- Recomputa os centros dos *clusters* usando os membros atualmente existentes nos *cluster*;
- Repita esses passos (a partir do passo de atribuição) até o critério de convergência ser alcançado. Normalmente, os critérios de parada utilizados são: a ocorrência de um decrescimento mínimo na função *e* de erro-quadrado, ou a não re-atribuição de um padrão a um novo *cluster*.

O algoritmo *k-means* é muito popular porque é fácil de ser implementado e sua complexidade é *O(n)* onde *n* é o número de padrões. No entanto, um grande problema deste algoritmo é que ele é sensível a seleção das partições iniciais e o valor da função *e* pode convergir para um mínimo local se as partições iniciais não forem apropriadamente escolhidas. Veja na **Figura 15** um exemplo da aplicação desse algoritmo sobre um conjunto de padrões de dimensão 2.

[19] Também podem ser definidos *k* pontos aleatórios dentro do hipervolume contendo o conjunto de padrões **X**.

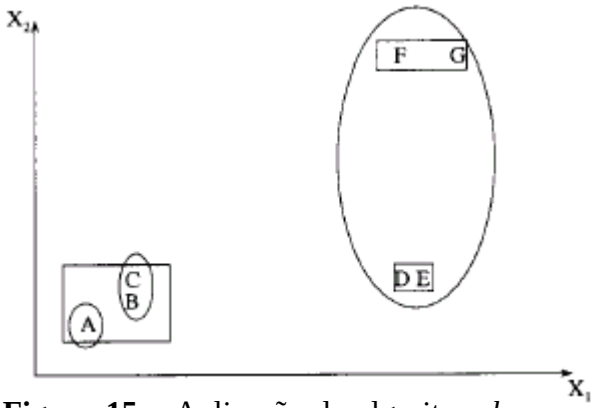


Figura 15 – Aplicação do algoritmo *k-means* (Jain et al., 1999, p.279).

Supondo que os padrões **A**, **B** e **C** sejam os padrões escolhidos para serem os centros de *clusters* iniciais. Conseqüentemente, o algoritmo termina com as 3 partições seguintes: (**A**), (**B**, **C**), (**D**, **E**, **F**, **G**). Estas partições são apresentadas na **Figura 15** circuladas por uma elipse.

O valor da função *e* calculada para a partição anterior é muito maior que o valor que é calculado para a melhor partição possível: (**A**, **B**, **C**), (**D**, **E**), (**F**, **G**). Estes *clusters* estão representados na **Figura 15** circulados por retângulos. Este particionamento ótimo seria calculado se os pontos **A**, **D** e **F** fossem escolhidos inicialmente como os centros de *clusters*. Um último detalhe que deve ser observado é que esse é o melhor particionamento considerando que se quer construir 3 *clusters*, por isso o nome *k-means* ! Caso se queira um outro número de *clusters* é necessário executar o algoritmo novamente com *k* sendo esse número de *clusters* desejado.

Na literatura é apresentada uma variação desse algoritmo básico *k-Means*, denominado ISODATA (Ball e Hall, 1965), que permite garantir que será feita uma partição ótima, independente dos pontos aleatórios escolhidos para o particionamento inicial.

Na próxima sub-seção é apresentado outro algoritmo de *clustering* baseado em conceitos da teoria de grafos.

6.2.2. Algoritmo de clustering Graph-Theoretic

Os algoritmos de clustering deste tipo se utilizam da teoria de grafos para o seu funcionamento. O melhor algoritmo conhecido dentre todos é baseado na construção de uma MST (*Minimal Spanning Tree*) ligando o conjunto de padrões (Zahn, 1971). Uma vez construída a MST, vai-se removendo as arestas com o maior valor (maior distância

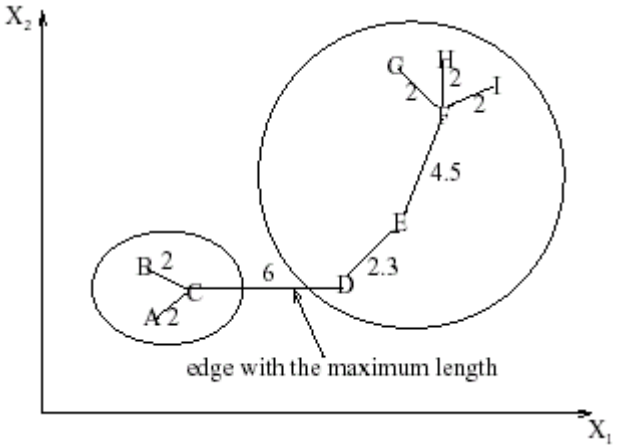


Figura 16 – Exemplo de aplicação de uma MST para construi clusters (Jain et al., 1999, p.279).

euclidiana) para a geração do número de *clusters* desejados. Veja na **Figura 16** um exemplo deste algoritmo sobre um conjunto de padrões de duas dimensões.

Na **Figura 16**, é removida inicialmente a aresta de valor 11 (aresta que liga os padrões **C** e **D**), resultando em dois *clusters*. Depois, caso se queira, pode-se particionar novamente removendo-se a aresta de peso 4.5 (aresta que liga os padrões **E**, **F**).

Na próxima sub-seção é apresentado um algoritmo simples de clustering, denominado *nearest neighbor clustering*.

6.2.3. Algoritmo de clustering nearest neighbor

Neste algoritmo o ponto fundamental é a proximidade entre os padrões, pois essa medida é uma noção intuitiva básica que o ser humano tem a respeito de um *cluster*.

De forma bem simples, Lu e Fu (1978) descrevem um algoritmo que implementa esta técnica. No algoritmo proposto por eles, o passo inicial é rotular alguns padrões e constituí-los como *clusters*. A partir desse momento, os padrões ainda não rotulados vão sendo incorporados paulatinamente aos seus *clusters* mais próximos (se essa distância estiver abaixo de um certo valor definido *a priori*). Esse processo é repetido até o momento em que não haja mais padrões não rotulados.

Na próxima seção são apresentadas algumas considerações finais a respeito de *clustering*.

7. Considerações Finais

As técnicas de *clustering* são amplamente utilizadas em diversas áreas, tais como, aprendizagem por observação (Michalski e Stepp, 1983), recuperação de informação (Salton, 1991; Rasmussen, 1992; Cutting et al., 1992; Steinbach et al., 2000; Dhillon et al., 2001), segmentação de imagens (Jain e Flynn, 1996) e classificação de padrões (Anderberg, 1973).

Neste texto procurou-se focar em mostrar uma visão geral sobre a área de *clustering*, os conceitos, as notações envolvidas e algumas das principais técnicas utilizadas. No entanto, além do que foi apresentado há uma farta bibliografia que pode ser consultada: Hartigan (1975), Spath (1980), Jain e Dubes (1988), Dubes (1993), Everitt (1993), Mirkin (1996), Fasulo (1999), Kolatch (2001), Han e Kamber (2001), Han et al. (2001), Ghosh (2002).

Além dessas, há outras referências importantes relacionadas a temas específicos da área de *clustering*: *Fuzzy clustering* (Zadeh, 1965), utilização de redes neurais para *clustering* (Sethi e Jain, 1991; Jain e Mao, 1994), *clustering* baseado em algoritmos evolucionários (Holland, 1975; Goldberg, 1989; Schwefel, 1981; Fogel et al, 1965), algoritmos que incorporam restrições de domínios no *clustering* (Watanabe, 1985).

Um ponto importante que deve ser ressaltado é fato de que o foco deste texto foi o funcionamento das diversas técnicas de *clustering* existentes, por isso não foram discutidos aspectos como a complexidade de tempo e/ou espaço dos algoritmos descritos. No entanto, uma discussão sobre esses aspectos pode ser vista em Kurita (1991), Choudhury e Muty (1990) e Anderberg (1973).

Outro ponto diz respeito ao desempenho dos humanos comparativamente aos algoritmos de *clustering*: os seres humanos são comparáveis aos procedimentos automáticos de *clustering* para espaços de até 2 dimensões. No entanto, os problemas mais realísticos envolvem a aplicação de *clustering* para casos com dimensões maiores que 2. Nesses casos, é difícil para os seres humanos obterem uma interpretação intuitiva dos dados envolvidos nesses espaços de muitas dimensões. Para dificultar ainda mais, esses dados multi-dimensionais dificilmente seguem estrutura ideais (hiper-esferas, ou padrões lineares). Esse fato explica o grande número de algoritmos de *clustering* que existem e que continuam a aparecer na literatura.

Concluindo, há duas linhas de pesquisa não muito exploradas relacionadas a área de *clustering*. A primeira linha diz respeito a falta de definição de um padrão de avaliação **geral** da efetividade de um *clustering* que foi realizado. Existem algumas formas de avaliação, mas somente para alguns sub-domínios muito bem especificados. A outra linha de pesquisa não muito desenvolvida está relacionada a falta de um estudo mais aprofundado da tendência dos dados de serem ou não passíveis de serem agrupados em *clusters* antes de se aplicar alguma técnica de *clustering*.

8. Referências Bibliográficas

Anderberg, M. R. Cluster Anlysis for Applications. Academic Press, Inc., New York, NY. 1973.

Ávila, G. Euclides, Geometria e Fundamentos. *Revista do Professor de Matemática,* 45, 2001. Capturado em 17 de mar. 2004. Online. Disponível em: http://www.bibvirt.futuro.usp.br/textos/hemeroteca/rpm/rpm45/rpm45_01.pdf.

Baeza-Yates, R. A. Introduction to data structures and algorithms related to information retrieval. In *Information Retrieval: Data Structures and Algorithms*, W. B. Frakes and R. Baeza-Yates, Eds. Prentice-Hall, Inc., Upper Saddle River, NJ, 13-27. 1992.

Ball, G. H.; Hall, D. J. ISODATA, a novel method of data analysis and classification. *Tech. Rep. Stanford University*, Stanford, CA. 1965.

Choudhury, S.; Murty, M. N. A divisive scheme for constructing minimal spanning trees in coordinate space. *Pattern Recognition. Lett. 11*, 6 (Jun. 1990), 385-389. 1990.

Cutting, D.; Karger, D.; Pedersen, J., Tukey, J. Scatter/gather: a cluster-based approach to browsing large document collection. In *Proceedings of the 15th ACM SIGIR Conference*, 318-329, Copenhagen, Denmark. 1992.

Dhillon, I.; Fan, J.; Guan, Y. Efficient clustering of very large document colections. In Grossman, R. L., Kamath, C., Kegelmeyer, P., Kumar, V., Namburu, R. R. (Eds.) *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishers. 2001.

Diday, E.; Simon, J. C. Clustering analysis. In *Digital Pattern Recognition*, K. S. Fu, Ed. Springer-Verlag, Secaucus, NJ, 47-94. 1976.

Dubes, R. C. Cluster Analysis and Related Issues. In Chen, C. H., Pau, L. F., and Wang, P. S. (Eds.) *Handbook of Pattern Recognition and Computer Vision*, 3-32, World Scientific Publishing Co., River Edge, NJ. 1993.

Everitt, B. Cluster Analysis (3rd ed.). Edward Arnold, London, UK. 1993.

Fasulo, D. An analysis of recent work on clustering algorithms. *Technical Report UW-CSE01-03-02*, University of Washington. 1999.

Fogel, L. J.; Owens, A. J.; Walsh, M. J. *Artificial Intelligence Through Simulated Evolution*. John Wiley and Sons, Inc., New York, NY. 1965.

Ghosh, J. Scalable Clustering Methods for Data Mining. In Nong Ye (Ed.) *Handbook of Data Mining*, Lawrence Erlbaum. 2002.

Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Inc., Redwood City, CA. 1989.

Gowda, K. C.; Krishna, G. Agglomerative clustering using the concept of mutual nearest neghborhood. *Pattern Recognition*. 10, 105-112. 1992.

Han, J.; Kamber, M. *Data Mining*. Morgan Kaufmann Publishers. 2001.

Han, J.; Kamber, M.; Tung, A. K. H. Spatial clustering methods in data mining: A survey. In Miller, H. and Han, J. (Eds.) *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis. 2001.

Hartigan, J. *Clustering Algorithms*. John Wiley & Sons, New York, NY. 1975.

Holland, J. H. *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI. 1975.

Ichino, M.; Yaguchi, H. Generalized Minkowski metrics for mixed feature-type data analysis. *IEEE Trans. Syst. Man Cybern*. 24, 698-708. 1994.

Jain, A., Dubes, R. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ. 1988.

Jain, A. K.; Flynn, P. J. Image segmentation using clustering. In Advances in Image Understanding: A Festschrift for Arzriel Rosenfeld, N. Ahuja and K. Bowyer, Eds, IEEE Press, Piscataway, NJ, 65-83. 1996.

Jain, A. K.; Mao, J. Artificial neural networks: A tutorial. *IEEE Computer 29* (Mar.), 31-44. 1996.

Jain, A. K.; Murty, M. N.; Flynn, P. J. Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 3, September 1999.

Jarvis, R. A.; Patrick, E. A. Clustering using a similarity method based on shared near neighbors. *IEEE Trans. Comput*. C-22, 8 (Aug.), 1025-1034. 1973.

Kaufman, L.; Rousseeuw, P. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, NY. 1990.

King, B. Step-wise clustering procedures. *J. Am. Stat. Assoc*. 69, 86-101. 1967.

Kolatch, E. *Clustering Algorithms for Spatial Databases: A Survey*. PDF is available on the Web. 2001.

Kurita, T. An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition. 24*, 3 (1991), 205-209. 1991.

Lu, S. Y.; Fu, K. S. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Trans. Syst. Man Cybern*. 8, 381-389. 1978.

Michalski, R.; Stepp, R. E.; Diday, E. Automated construction of classifications: conceptual clustering versus numerical taxonomy. *IEEE Trans. Pattern Anal. Mach*. Intell, PAMI-5, 5 (Sept.), 396-409. 1983.

Mirkin, B. *Mathematic Classification and Clustering*. Kluwer Academic Publishers. 1996.

Murtagh, F. A survey of recent advances in hierarchical clustering algorithms which use cluster centers. *Comput. J*. 26, 354-359. 1984.

Rasmussen, E. Clustering algorithms. In Information Retrieval: Data Structures and Algorithms, W. B. Frakes and R. Baeza-Yates, Eds. Prentice-Hall, Inc., Upper Saddle River, NJ, 419-442. 1992.

Salton, G. Developtments in automatica text retrieval. *Science* 253, 974-980. 1991.

Schwefel, H. P. *Numerical Optimization of Computer Models*. John Wiley and Sons, Inc., New York, NY. 1981.

Sethi, I.; Jain, A. K. Eds. *Artificial Neural Networks and Pattern Recognition: Old and New Connections*. Elsevier Science Inc., New York, NY. 1991.

Sneath, P. H. A.; Sokal, R. R. *Numerical Taxonomy*. Freeman, London, UK. 1973.

Spath, H. *Cluster Analysis Algorithms*. Ellis Horwood, Chichester, England. 1980.

Steinbach, M.; Karypis, G.,;Kumar, V. A comparison of document clustering techniques. *6th ACM SIGKDD, World Text Mining Conference*, Boston, MA. 2000.

Ward, J. H. Jr. Hierarchical grouping to optimize na objective function. *J. Am. Stat. Assoc.* 58, 236-244. 1963.

Watanabe, S. *Pattern Recognition: Human and Mechanical*. John Wiley and Sons, Inc., New York, NY. 1985.

Wilson, D. R.; Martinez, T. R. Improved heterogeneous distance function. *J. Artif. Intell. Res.* 6, 1-34. 1997.

Zadeh, L. A. Fuzzy sets. *Inf. Control 8*, 338-353. 1965.

Zahn, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput*. C-20 (Apr.), 68-86. 1971.

Zhang, K. Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recognition*. 28, 463-474. 1995.