Revista de Sistemas de Informação da FSMA

n. 4 (2009) pp. 18-36

SISTEMAS DE INFORMAÇÃO

http://www.fsma.edu.br/si/sistemas.html

TUTORIAL

Técnicas de Agrupamento

Ricardo Linden

Resumo

Este capítulo descreve métodos de agrupamento que permitem que se extraiam características interessantes destes dados, separando-os em grupos funcionais ou hierarquizando-os para estudos posteriores.

Palavras Chave *Agrupamento; Heurísticas; K-Means; Redes Neurais; Grafos*

Abstract

This chapter describes some clustering techniques for feature extraction are described. These methods work well for functional group separation and can be used for further studies.

Keywords *Clustering; Heuristics; K-Means; Neural Nets; Graphs*

1. Introdução

Nesta seção serão introduzidos os principais conceitos associados ao problema de agrupamento. O leitor perceberá a dificuldade associada ao processo e entenderá os usos principais da técnica. Além disto será apresentado o ferramental matemático básico para que o leitor possa compreender os algoritmos introduzidos na seção 2.2.

1.1 Agrupamento: Conceitos Básicos

Análise de agrupamento, ou clustering, é o nome dado para o grupo de técnicas computacionais cujo propósito consiste em separar objetos em grupos, baseando-se nas características que estes objetos possuem. A idéia básica consiste em colocar em um mesmo grupo objetos que sejam similares de acordo com algum critério pré-determinado.

O critério baseia-se normalmente em uma função de dissimilaridade, função esta que recebe dois objetos e retorna a distância entre eles. As principais funções serão discutidas na seção 2.1.2. Os grupos determinados por uma métrica de qualidade devem apresentar alta homogeneidade interna e alta

separação (heterogeneidade externa). Isto quer dizer que os elementos de um determinado conjunto devem ser mutuamente similares e, preferencialmente, muito diferentes dos elementos de outros conjuntos.

Os objetos também são denominados exemplos, tuplas e/ou registros. Cada objeto representa uma entrada de dados que pode ser constituída por um vetor de atributos que são campos numéricos ou categóricos (categórico é um tipo de campo que pode assumir um entre um conjunto de valores pré-definidos). Exemplos de dados numéricos incluem idade (inteiro), temperatura (real) e salário (real), entre outros, enquanto que exemplos de dados categóricos incluem bases de DNA (um dentre os valores A, C, G ou T), patente militar (soldado, cabo, sargento, tenente, etc) ou se a pessoa está doente (valor Booleano, podendo assumir os valores verdadeiro ou falso) (GORDON, 1981).

A análise de agrupamento é uma ferramenta útil para a análise de dados em muitas situações diferentes. Esta técnica pode ser usada para reduzir a dimensão de um conjunto de dados, reduzindo uma ampla gama de objetos à informação do centro do seu conjunto. Tendo em vista que clustering é uma técnica de aprendizado não supervisionado (quando o aprendizado é supervisionado, o processo é denominado de classificação), pode servir também para extrair características escondidas dos dados e desenvolver as hipóteses a respeito de sua natureza.

Infelizmente o problema de agrupamento apresenta uma complexidade de ordem exponencial. Isto quer dizer que métodos de força bruta, como enumerar todos os possíveis grupos e escolher a melhor configuração, não são viáveis. Por exemplo, se quisermos separar 100 elementos em 5 grupos, existem $5^{100} \approx 10^{70}$ possíveis agrupamentos. Mesmo em um computador que possa testar 10^9 configurações diferentes por segundo, seriam necessários mais de 10^{53} anos para terminar a tarefa (cerca de 40 ordens de grandeza maior do que a idade da Terra). Obviamente, é necessário então buscar uma heurística eficiente que permita resolver o problema antes do Sol esfriar e a vida na Terra se extinguir.

1.2 Medidas de Dissimilaridade

Todos os métodos de agrupamento que descreveremos na próxima sessão assumem que todos os relacionamentos relevante entre os objetos podem ser descritos por uma matriz contendo uma medida de *dissimilaridade* ou de *proximidade* entre cada par de objetos.

Cada entrada p_{ij} na matriz consiste em um valor numérico que demonstra quão próximos os objetos i e j são. Algumas métricas calculam a similaridade, outras calculam a dissimilaridade, mas em essência elas são idênticas.

Todos os coeficientes de dissimilaridade são funções $d : \Gamma X \Gamma \Rightarrow \Re$, onde Γ denota o conjunto de objetos com o qual estamos trabalhando. Basicamente, estas funções permitem realizar a transformação da matriz de dados, dada por:

$$\Gamma = \begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \quad (1)$$

em uma matriz de distâncias, dada por:

$$d = \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix} \quad (2)$$

onde a entrada d(i,j) representa exatamente a distância entre os elementos i e j.

Todas as funções de dissimilaridade devem obedecer os seguintes critérios básicos:

- $d_{ij} \geq 0, \forall i,j \in \Gamma$
- $d_{ij} = d_{ji}, \forall i,j \in \Gamma$. Esta regra afirma que a distância entre dois elementos não varia, não importando o ponto a partir do qual ela é medida. Por isto, a matriz de (2.2) é mostrada sendo triangular inferior. Tendo em vista que ela é simétrica, os valores acima da diagonal estão implicitamente definidos.
- $d_{ij} + d_{jk} \geq d_{ik}, \forall i,j,k \in \Gamma$. Esta é conhecida como a desigualdade triangular, e basicamente especifica que a menor distância entre dois pontos é uma reta.

Se, além de todas as propriedades listadas acima, a métrica também possui a propriedade d(ax,ay)=|a|d(x,y), então ela é chamada de uma norma. Veremos agora as principais medidas de dissimilaridade usadas nos algoritmos de agrupamento.

A primeira métrica a ser mencionada é a chamada métrica de Minkowski. Ela é dada pela seguinte fórmula:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)} \quad (3)$$

Como se pode ver facilmente na fórmula, as métricas de Minkowski são uma norma. Para verificar isto, basta multiplicar todos os elementos por *a*. Dentro da raiz, o valor a^q pode ser colocado em evidência e posteriormente retirado da raiz, assumindo o valor de *a*, como exige a definição.

O caso especial mais conhecido desta métrica é quando *q=2*. Neste caso, temos a conhecida distância euclidiana entre dois pontos, que é dada por:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)} \quad (4)$$

Existem duas versões das métrica de Minkovski que são populares. Quando *q* assume valor igual a 1, temos chamada distância Manhattan e quando q vai a infinito, temos a distância de Chebyshev. Neste último caso, temos que a distância é dada por:

$$d(i,j) = \max_k |x_{ik} - x_{jk}| \quad (5)$$

A escolha do valor de *q* depende única e exclusivamente da ênfase que se deseja dar a distâncias maiores. Quanto maior o valor *q* maior a sensitividade da métrica a distâncias maiores. Um exemplo simples é quando queremos calcular a distância entre os pontos (0,2,1) e (1,18,2). Usando diferentes valores de q, temos os seguintes valores para as distâncias:

- q=1 → d=[|1-0| + |18-2| + |2-1|] = 18
- q=2 → d=[$(1-0)^2 + (18-2)^2 + (2-1)^2$] $^{1/2}$ = $258^{1/2}$ = 16,06
- q=3 → d=[$(1-0)^3 + (18-2)^3 + (2-1)^3$] $^{1/3}$ = $4098^{1/3}$ = 16,002
- q=∞ → d=[$(1-0)^\infty + (18-2)^\infty + (2-1)^\infty$]$^{1/\infty}$ = 16

A figura 1 mostra a diferença de interpretação entre as várias versões da métrica de Minkovsky quando estamos em duas dimensões.

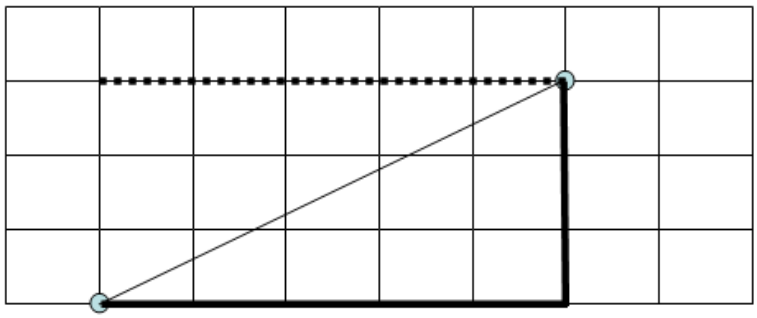

Figura 1. Exemplo de distância calculada pelas distintas métricas. A distância euclidiana (linha fina) é calculada através de uma linha reta entre os pontos. A distância Manhattan é calculada um quarteirão (unidade) de cada vez. O fato de não irmos reto não muda a distância (comprove!!!). A distância de Chebyschev (linha tracejada) é dada pela maior das duas dimensões da distância

Outra métrica de distância cujo cálculo é extremamente simples é a distância Camberra. Esta métrica calcula a soma das diferenças fracionárias entre as coordenadas de pares de objetos. A fórmula para esta métrica é dada por:

$$d(\vec{x},\vec{y}) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{|x_i| + |y_i|} \qquad (6)$$

Se uma das coordenadas é igual a zero, o termo se torna igual a 1. Em todos os outros casos, o termo pertence ao intervalo [0,1]. O somatório total sempre oferece um número pertencente ao intervalo [0,p], onde p é a dimensão dos vetores.

Se ambos os vetores têm uma coordenada *i* igual a zero, estabelece-se por definição que o termo *i* é igual a zero (na verdade ele é igual a 0/0). Um dos problemas desta métrica é que a distância é muito sensível a pequenas variações quando ambas as coordenadas estiverem muito próximas de zero.

Outro problema associado à métrica de Canberra é que coordenadas que cuja distância for a mesma, mas que tiverem módulo diferente apresentarão contribuições distintas. Por exemplo, imagine que a coordenada *i* apresenta os valores 1 e 4. Logo a contribuição do termo *i* será $|1-4| \big/ |1|+|4| = 0{,}6$. Entretanto, se a coordenada *j* apresenta os valores 30 e 33, sua contribuição será $|30-33| \big/ |30|+|33| \approx 0{,}05$. Isto é, as contribuições são de ordens de grandeza diferentes, apesar das diferenças serem idênticas.

A distância de Mahalanobis é uma métrica que difere da distância Euclidiana por levar em consideração a correlação entre os conjutos de dados. A fórmula para distância de Mahalanobis entre dois vetores da mesma distribuição que possuam uma matriz de covariância **Σ** é dada por:

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T \Sigma^{-1} (\vec{x}-\vec{y})} \qquad (7)$$

Se a matriz de covariância é a matriz identidade, esta fórmula se reduz à distância euclidiana. Caso a matriz de covariância seja diagonal, a distância de Mahalanobis se reduz à distância Euclidiana normalizada, e sua fórmula é dada por:

$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{P} \frac{(x_i - y_i)^2}{\sigma_i^2}} \qquad (8)$$

Nesta fórmula, p é a dimensão dos vetores e σ_i é o desvio padrão de x_i no espaço amostral.

Pode ser mostrado que as superfícies onde a distância de Mahalanobis são constantes são elipsóides centradas na média do espaço amostral. No caso especial em que as características são não correlacionadas, estas superfícies são esferódeis, tal qual no caso da distância Euclidiana.

O uso da distância de Mahalanobis corrige algumas das limitações da distância Euclidiana, pois leva em consideração automaticamente a escala dos eixos coordenados e leva em consideração a correlação entre as características. Entretanto, como não existe almoço grátis, há um preço (alto) a se pagar por estas vantagens. As matrizes de covariância podem ser difíceis de determinar e a memória e o tempo de computação crescem de forma quadrática com o número de características.

Pode-se usar também a métrica de Mahalanobis para medir a distância entre um elemento *x* e um cluster de elementos cuja média é dada por $\mu = \{\mu_1, \mu_2, ..., \mu_n\}$ e que possua uma matriz de covariância dada por Σ. Neste caso, a distância é dada pela fórmula $d = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$ .Conceitualmente, é como se estivéssemos avaliando a pertinência de um elemento não só por sua distância ao centro do cluster, mas também pela distribuição do mesmo, determinando assim a distância do elemento *x* em termos de uma comparação com o desvio padrão do cluster. Quanto maior for o valor desta métrica, maior o número de desvios padrões que um elemento esta distante do centro do cluster, e menor sua chance de ser um elemento do mesmo.

A detecção de outliers é um dos usos mais comuns da distância de Mahalanobis, pois um valor alto determina que um elemento está a vários desvios padrões do centro (já computadas todas as diferenças axiais) e, por conseguinte, é provavelmente um outlier. Se fizermos a comparação com vários grupos existentes, cujas distribuições sejam conhecidas, esta métrica pode ser usada como maneira de determinar a qual grupo um elemento mais provavelmente deve pertencer.

Outra métrica interessante é a correlação. Esta mede a força do relacionamento de duas variáveis, logo é uma medida de similaridade. O coeficiente de correlação é representado pela letra "r" e não possui unidades. Sua faixa de valores é [-1, 1], onde a magnitude da correlação representa sua força e o seu sinal representa o tipo de relacionamento: sinais positivos indicam associação direta (quando uma cresce a outra também cresce) enquanto que sinais negativos indicam associação inversa (quando uma cresce a outra diminui).

A correlação é calculada usando-se a seguinte fórmula:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right) \qquad (9)$$

Nesta fórmula, $\bar{x}$ e S_x representam as média e os desvios padrões para os valores de *x* e $\bar{y}$ e S_y para os valores de *y*, e *n* e o número de indivíduos.

1.3 Representação de Grupos

Existem várias formas distintas de representar os grupos criados pelos algoritmos de agrupamento explicados neste capítulo. As maneiras mais usuais são demonstradas na figura 2. Alguns formatos estão associados a métodos específicos, como por exemplo, os dendogramas (fig. 2d), que estão associados aos métodos hierárquicos, enquanto que outros são mais genéricos, como partições e os diagramas de Venn (fig. 2a e b, respectivamente). As tabelas (fig. 2c) são mais úteis no caso de métodos que usam lógica nebulosa, mas podem ser usados em outros casos, tendo poder de representação similar aos diagramas de Venn, por exemplo. Cada formato será explicado com detalhes e usado quando discutirmos as heurísticas de agrupamento, na seção 2.

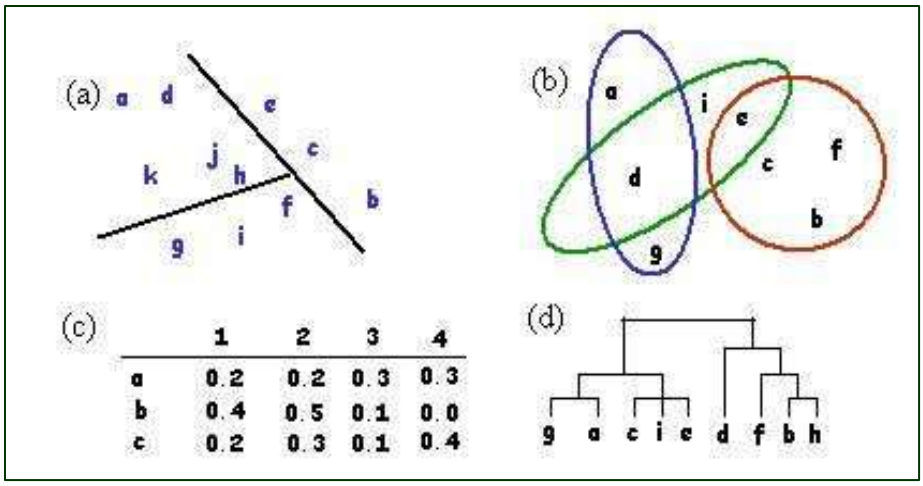

Figura 2. Exemplos de várias formas de representação de grupos.

1.4 Aplicações de Técnicas de Agrupamento

Técnicas de agrupamento têm sido usadas em várias áreas distintas. Algumas das aplicações específicas, além da área de bioinformática, que serão descritas mais adiante, incluem:

- *Marketing:* Pode auxiliar pessoas ligadas a áreas de marketing a descobrir grupos distintos em suas bases de clientes, para que este conhecimento seja usado então para desenvolver programas de marketing direcionados (CHIANG, 2003).
- *Uso de terras*: Identificação de possibilidade de alocação de uso da terra para fins agrários e/ou urbanos em uma base de dados de observação via satélite de todo o planeta Terra (LEVIA JR, 2000).
- *Seguro:* Identificar grupos de pessoas que possuam seguro de carro com um custo elevado de sinistralidade (YEOH, 2001).
- *World Wide Web*: agrupamento de documentos de acordo com similaridades semânticas, de forma a melhorar o resultados oferecidos por sites de busca (HAMMOUDA, 2002).
- *Estudos do terremoto*: Análise de dados reais e sintéticos de terremotos para extração de características que permitam a previsão de eventos precursores de abalos sísmicos (DZWINNEL, 2005).

2. Heurísticas de Agrupamento

Nesta seção serão apresentados alguns dos algoritmos mais usados na área de agrupamento, sendo que cada algoritmo será descrito de forma completa, com uma dissertação sobre suas principais forças e fraquezas.

2.1. K-Means

O K-Means é uma heurística de agrupamento não hierárquico que busca minimizar a distância dos elementos a um conjunto de *k* centros dado por $\chi=\{x_1,x_2,...,x_k\}$ de forma iterativa. A distância entre um ponto p_i e um conjunto de clusters, dada por $d(p_i,\chi)$, é definida como sendo a distância do ponto ao centro mais próximo dele. A função a ser minimizada então, é dada por:

$$d(P,\chi)=\frac{1}{n}\sum_{i=1}^{n} d(p_i,\chi)^2$$

O algoritmo depende de um parâmetro (k=número de clusters) definido de forma *ad hoc* pelo usuário. Isto costuma ser um problema, tendo em vista que normalmente não se sabe quantos clusters existem *a priori*.

O algoritmo do K-Means pode ser descrito da seguinte maneira:

1. Escolher k distintos valores para centros dos grupos (possivelmente, de forma aleatória)
2. Associar cada ponto ao centro mais próximo
3. Recalcular o centro de cada grupo
4. Repetir os passos 2-3 até nenhum elemento mudar de grupo.

Este algoritmo é extremamente veloz, geralmente convergindo em poucas iterações para uma configuração estável, na qual nenhum elemento está designado para um cluster cujo centro não lhe seja o mais próximo. Um exemplo da execução do algoritmo de K-Means pode ser visto na figura 3.

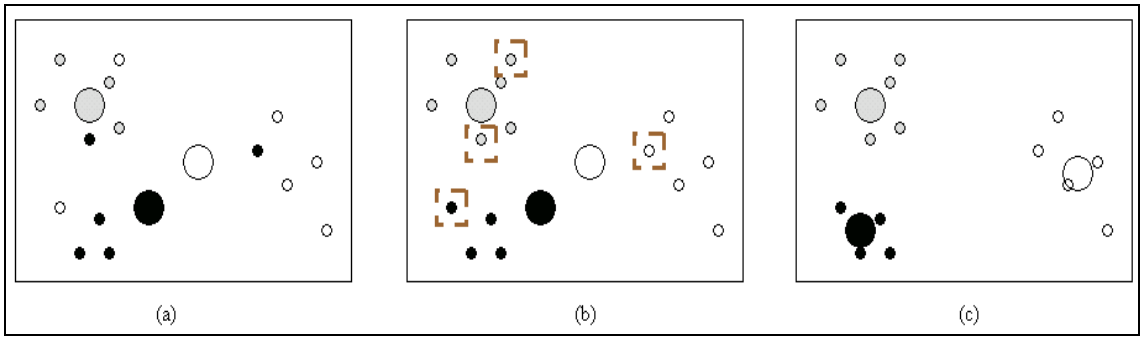

Figura 3. Exemplo de execução do algoritmo de K-Means. (a) Cada elemento foi designado para um dos três grupos aleatoriamente e os centróides (círculos maiores) de cada grupo foram calculados. (b) Os elementos foram designados agora para os grupos cujos centróides lhe estão mais próximos (c) Os centróides foram recalculados. Os grupos já estão em sua forma final. Caso não estivessem, repetiríamos os passos (b) e (c) até que estivessem.

Este algoritmo iterativo tende a convergir para um mínimo da função de energia definida acima. Um eventual problema é que esta condição enfatiza a questão da homogeneidade e ignora a importante questão da boa separação dos clusters. Isto pode causar uma má separação dos conjuntos no caso de uma má inicialização dos centros, inicialização esta que é feita de forma arbitrária (aleatória) no início da execução. Na figura 4 pode-se ver o efeito de uma má inicialização na execução de um algoritmo de K-Means.

Outro ponto que pode afetar a qualidade dos resultados é a escolha do número de conjuntos feita pelo usuário. Um número pequeno demais de conjuntos pode causar a junção de dois clusters naturais, enquanto que um número grande demais pode fazer com que um cluster natural seja quebrado artificialmente em dois.

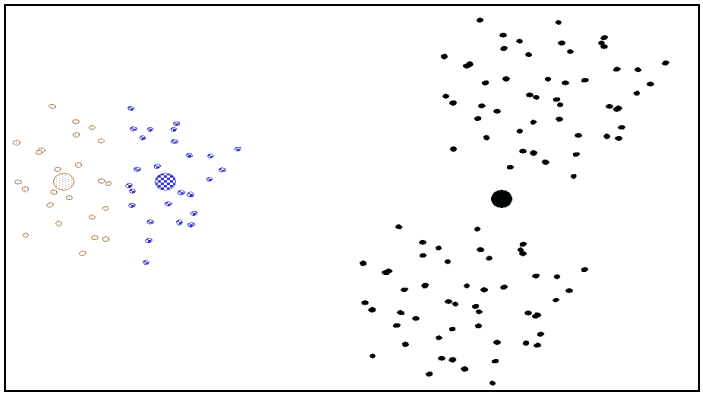

Figura 4. Exemplo do efeito de uma má inicialização do algoritmo K-Means. Existem três clusters naturais neste conjunto, dois dos quais foram atribuídos ao grupo de cor sólida. O problema é que o terceiro cluster é bem separado dos outros dois clusters, e se dois conjuntos forem inicializados daquele lado da figura, eles ficarão daquele lado.

A grande vantagem desta heurística é a sua simplicidade. Um programador experiente pode implementar uma versão própria em cerca de uma hora de trabalho. Além disto, vários demonstrativos estão disponíveis em vários sites, dentre os quais podemos destacar o localizado no endereço dado por http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html (último acesso em novembro/2009).

2.2. Algoritmos Hierárquicos

Os algoritmos hierárquicos criam uma hierarquia de relacionamentos entre os elementos. Eles são muito populares na área de bioinformática e funcionam muito bem, apesar de não terem nenhuma justificativa teórica baseada em estatística ou teoria da informação, constituindo uma técnica *ad-hoc* de alta efetividade.

Existem duas versões: a aglomerativa, que opera criando conjuntos a partir de elementos isolados, e a divisiva, que começa com um grande conjunto e vai quebrando-o em partes até chegar a elementos isolados.

A versão aglomerativa funciona de acordo com o seguinte algoritmo:

1. Faça um cluster para cada elemento
2. Encontre os pares de clusters mais similares, de acordo com uma medida de distância escolhida
3. Funda-os em um cluster maior e recalcule a distância deste cluster para todos os outros elementos
4. Repita os passos 2 e 3 até sobrar um único cluster.

Podemos fazer um exemplo simples para visualizar o funcionamento dos algoritmos hierárquicos. Para tanto, peguemos o conjunto de pontos dados por S={2; 4; 6,3;9;11,6} e façamos o *clustering* destes elementos.

Primeiro calculamos a distância entre os pontos, diretamente usando, por exemplo a distância euclidiana:

$$D = \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 4{,}3 & 2{,}3 & 0 & & \\ 7 & 5 & 2{,}7 & 0 & \\ 9{,}6 & 7{,}6 & 5{,}3 & 2{,}6 & 0 \end{bmatrix},$$

Como podemos ver, os elementos mais próximos são o primeiro (2) e o segundo (4). Realizamos então o agrupamento mostrado na figura 5 (b). As novas distâncias podem ser recalculadas substituindo-se os dois elementos agrupados pela sua média (3) e são as seguintes:

$$D = \begin{bmatrix} 0 & & & \\ 3{,}3 & 0 & & \\ 6 & 2{,}7 & 0 & \\ 8{,}6 & 5{,}3 & 2{,}6 & 0 \end{bmatrix}$$

Neste momento, os dois mais próximos são o terceiro (9) e o quarto (11,6) e realizamos o agrupamento mostrado na figura 5(c). As novas distâncias são recalculadas, obtendo-se então os seguintes valores:

$$D = \begin{bmatrix} 0 & & \\ 3{,}3 & 0 & \\ 7{,}3 & 4 & 0 \end{bmatrix}$$

Realizamos o próximo agrupamento entre o primeiro (3) e o segundo (6,3) elementos, obtendo-se os conjuntos mostrados na figura 5(d). Por último, resta agrupar os dois últimos grupos restantes, obtendo-se o agrupamento final, mostrado na figura 5(e)

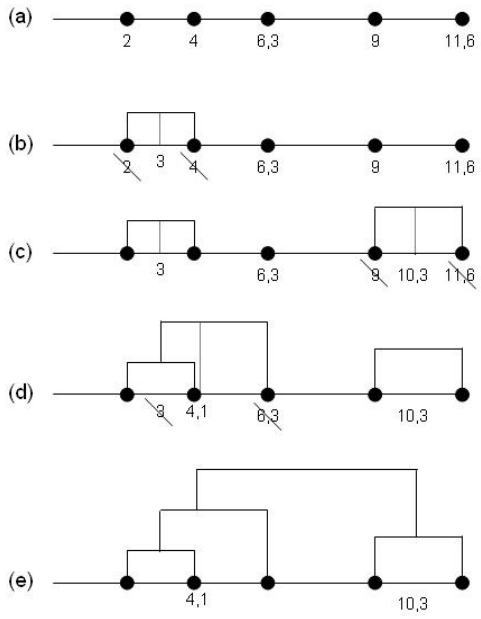

Figura 5. Representação do passo a passo do exemplo de utilização de algoritmo hierárquico.

Note-se que a estrutura hierárquico formada pela união entre os elementos foi representada através de um dendograma, que é a forma mais usual de representação dos resultados de algoritmos hierárquicos e mostra intuitivamente a ordem do agrupamento. Quanto mais alta a linha ligando dois *clusters*, mais tarde foi feito seu agrupamento. Logo, a altura da linha ligando dois clusters é proporcional à sua distância.

Existem três maneiras diferentes de medir a distância entre dois *clusters*, que são descritas a seguir e exemplificadas na figura 6.

- Single Link: A distância entre dois clusters é dada pela distância entre os seus pontos mais próximos, também chamada de "agrupamento de vizinhos" ("neighbour clustering"). É um método guloso, que prioriza elementos mais próximos e deixando os mais distantes em segundo plano.
- Average Link: A distância é dada pela distância entre seus centróides. O problema recalcular centróides e distâncias entre clusters cada vez que um cluster muda.
- Complete Link: A distância entre clusters é a distância entre seus pontos mais distantes.

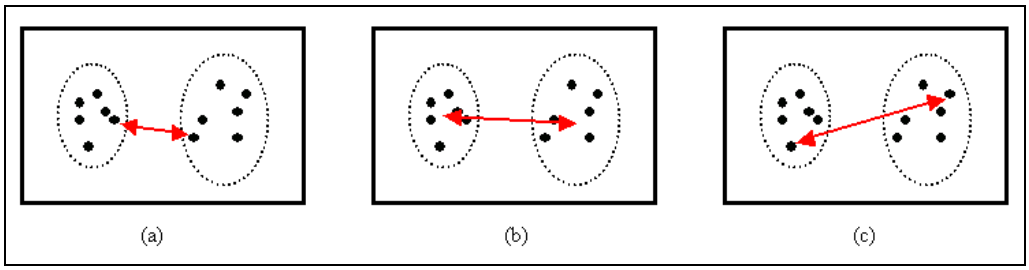

Figura 6. Medidas de distância entre dois clusters para uso em algoritmos hierárquicos. (a) single link, onde a distância entre os clusters é dada pela distância entre seus pontos mais próximos. (b) average link, onde a distância entre os clusters é dada pela distância entre seus centróides e (c) complete link, onde a distância entre os clusters é dada pela distância entre seus pontos mais distantes.

Estes três métodos de medir distância não são totalmente equivalentes. Escolhas distintas entre estas formas de medir distâncias podem gerar resultados diferentes no agrupamento, como pode ser visto na figura.7.

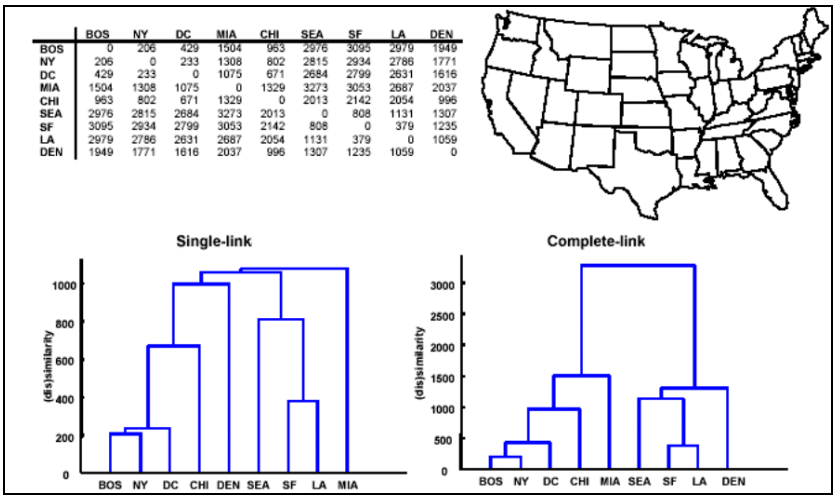

Figura 7. Exemplo de aplicação de algoritmo hierárquico com duas formas diferentes de se medir distância. Dada as distâncias entre cidades dos Estados Unidos, construiu-se a hierarquia usando-se a medida de distância single-link (à esquerda) e complete-link (à direita). Note que os conjuntos gerados são distintos. No single-link, Miami é a última cidade a ser adicionada, enquanto que no complete link ela é rapidamente adicionada

ao conjunto da esquerda (Exemplo retirado do site http://www.analytictech.com/networks/hiclus.htm, último acesso em Novembro/2009) .

O problema deste algoritmo é que com ele obtém-se apenas uma hierarquia de relacionamentos, mas não grupos específicos. Para obter *k* grupos, basta apenas cortar as *k-1* arestas mais altas do dendograma. Um exemplo pode ser visto na figura 8.

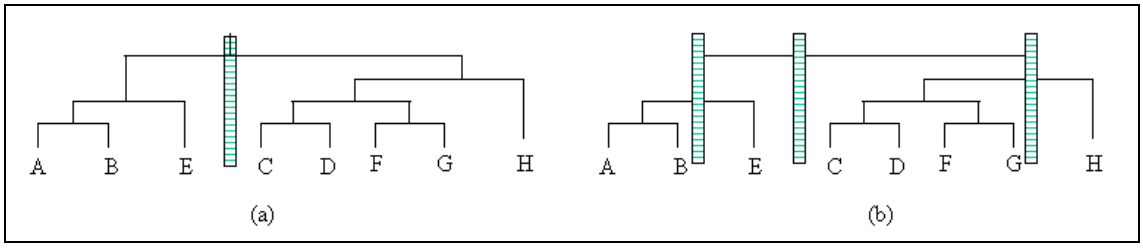

Figura 8. Exemplo de corte de um dendograma obtido através do algoritmo hierárquico para obtenção de clusters. Na figura A foi feito 1 corte e foram obtidos dois grupos. Já na figura (b), foram feitos três cortes e obtidos 4 grupos ({A,B},{E},{C,D,F,G},{H}).

Existe uma versão divisiva do algoritmo hierárquico de agrupamento. Esta versão se inicia com um único cluster contendo todos os elementos e começa a dividi-lo. Em cada iteração, utiliza-se um algoritmo *flat* (como o K-Means, por exemplo), de forma a separar o conjunto corrente em *clusters* menores, repetindo-se o processo recursivamente até que se tenha apenas conjuntos compostos de um único elemento ou até que um critério de parada previamente estabelecido seja atendido.

Note-se que é habitual dividir-se o cluster selecionado em dois clusters menores pois isto faz com que esta técnica pareça ser a inversa dos algoritmos aglomerativos. Entretanto, o numero de clusters gerados não precisa necessariamente ser igual a dois. Uma forma de escolher o número de *clusters* a ser gerados é utilizar uma métrica de otimização de custo de cortes. Esta utiliza dois custos fundamentais, denominados similaridades intra-cluster e similaridade extra-cluster e busca maximizar a relação entre os dois de forma a obter o particionamento no qual os clusters gerados tenham maior coesão. Matematicamente temos o seguinte:

Dados

- $V = \{v_1, v_2, ..., v_n\}$, o conjunto de todos os objetos existentes
- $\{C_1, C_2, ..., C_k\} \mid V = \bigcup_{i=1}^{k} C_i$, um particionamento dos objetos , então temos que:
- similaridade intra-cluster dada por $\text{int}ra_p = \sum_{v_i, v_j \in C_p} sim(v_i, v_j)$
- similaridade extra-cluster dada por $extra_p = \sum_{v_i \in C_p, v_j \notin C_p} sim(v_i, v_j)$

Então, o custo do particionamento com *k* conjuntos é dado por $custo_k = \sum_{p=1,2,...k} \frac{\text{int}ra_p}{extra_p}$, escolhendo-se então o particionamento para o qual o valor do custo seja o menor possivel.

Esta técnica é extremamente custosa em termos computacionais, pois exige que se faça um grande número de particionamentos em cada iteração, o que pode levar um longo tempo se o número de elementos for grande. O custo pode ser limitado, ao se colocar um teto para o valor de , mas ainda assim este algoritmo pode ter um tempo de execução longo demais para ser aceitável.

O funcionamento da versão divisiva é mais complexo do que o do algoritmo aglomerativo, pois depende de um algoritmo *flat* tal como o K-Means para seu funcionamento. Entretanto, ele tem duas vantagens principais:

1. Não é necessário recalcular todas as distâncias entre os *clusters* a cada passo e podemos interromper o procedimento antes de chegarmos ao último nível da árvore. Assim, o algoritmo é potencialmente mais rápido do que sua versão aglomerativa.
2. Começamos usando as informações sobre o todo, ao invés de tomar apenas decisões baseadas em informações locais, como no caso aglomerativo. Assim, os agrupamentos obtidos tendem a ser mais fiéis quanto à real distribuição da informação nos elementos.

Um fato interessante é que esta é uma das maneiras possíveis de se determinar quantos conjuntos definiremos para a técnica de K-Means. Ao invés de definirmos a priori, realizamos o seguinte algoritmo:

```
Defina um conjunto com todos os elementos e insira-o em uma fila de conjuntos a tratar;
Enquanto houver conjuntos na fila faça
        Retire o primeiro conjunto da fila;
        Calcule a perda de informação relacionada a este conjunto;
        Se o valor da perda de informação for maior que um limite pré-estabelecido
        Realize o K-Means com k=2;
        Coloque os dois conjuntos obtidos na fila de conjuntos a tratar;
        Fim Se
Fim Enquanto
```

Pode parecer que apenas substituimos uma arbitrariedade por outra (afinal, à primeira vista apenas trocamos o *k* pelo limite de perda de informação). Entretanto, note-se que agora, ao invés de definir arbitrariamente o número de conjuntos que desejamos, definimos uma métrica de qualidade para cada conjunto que deve ser definida de acordo com o domínio do problema que estamos tentando resolver. Isto é, o nosso problema impõe uma qualidade inerente aos conjuntos e é esta que vamos buscar.

Outra forma de se realizar um algoritmo hierárquico divisivo é partir da *minimal spanning tree*. Esta pode ser definida como a árvore que conecta todos os elementos do conjunto de tal forma que a soma das distâncias de todas as arestas existentes é mínima. Uma vez fornecida esta árvore, para cuja determinação existe um algoritmo que é praticamente linear em relação ao número de elementos, pode-se dividi-la em dois conjuntos usando-se uma das seguintes maneiras:

- Removendo a maior de todas as arestas;
- Removendo a aresta de comprimento l tal que $l >> l'$, onde l' representa o comprimento médio das arestas do grupo. Este método tem como vantagem permitir que clusters mais densos também tenham seus *outliers* retirados, o que poderia demorar mais, posto que sendo densos todas as suas arestas devem ter um comprimento relativamente pequeno.

Usando-se qualquer uma das técnicas obteremos dois clusters para o cluster sendo dividido. Repetimos então o processo até que seja atingido o número de clusters desejado.

No passado, os algoritmos hierárquicos eram extremamente populares, mas com o avanço das técnicas flat, estas ganharam aceitação. Logo, a decisão sobre usar algoritmos hierárquicos é complexa para tomá-la, devemos levar em consideração uma série de importantes fatores.

O primeiro fator é o consumo de memória. Para usar um algoritmo hierárquico devemos armazenar $O(n^2)$ distâncias. Mesmo com o preço das memórias caindo de forma extremamente rápida, se temos 10.000 elementos, teremos que armazenar um número de distâncias da ordem de 10^8. Este valor ainda está dentro da ordem de grandeza de um computador pessoal, mas já é um valor respeitável e deve ser gerenciado com carinho (e eficiência).

Os algoritmos não hierárquicos normalmente dependem de uma série de fatores que são determinados de forma arbitrária pelo pesquisador, como número de conjuntos e as *seeds* de cada

conjunto. Isto pode causar impacto negativo na qualidade dos conjuntos gerados. Os algoritmos hierárquicos aglomerativos não são sujeitos a estes fatores, sendo totalmente determinísticos e independentes de parametrização.

A principal vantagem dos algoritmos hierárquicos é o fato deles oferecerem não só os clusters obtidos, mas também toda a estrutura dos dados e permitirem obter facilmente sub-conjuntos dentro destes dados. Ademais, eles permitem visualizar diretamente através do dendograma, a forma como os dados se ligam e a verdadeira semelhança entre dois diferentes pontos, pertençam estes ao mesmo cluster ou não.

2.3 Mapas de Características Auto-Organizáveis

Um sistema é um grupo de partes que interagem e que funcionam como um todo, possuindo propriedades coletivas distintas de cada uma das partes. Os sistemas possuem uma organização para obter uma função específica. A organização pode ser de dois tipos:

- externa, quando é imposta por fatores externos, como máquinas, paredes, etc.
- auto-organização: quando o sistema evolui para uma forma organizada por conta própria. Exemplos: cristalização, cérebro, economias, etc.

O cérebro claramente é um sistema organizado que aprende espontaneamente, sem um professor. Antigamente, a visão científica dominante era que havia homúnculos dentro do cérebro, orientando o aprendizado. Hoje em dia, acredita-se mais na regra de aprendizado de Hebb. Esta postula que quando um neurônio A repetida e persistentemente ajuda a disparar o neurônio B, a eficácia da associação entre as duas células aumenta (HEBB, 1949).

Estas mudanças ocorrem no cérebro de três maneiras possíveis: aumentando o número de transmissores emitidos pela célula A, aumentando o força da ligação sináptica ou formando novas sinapses. Isto nos permite chegar à conclusão de que o cérebro é um sistema auto-organizado, podendo aprender modificando as interconexões entre neurônios.

O principal resultado da auto-organização cerebral é a formação de mapas de características de topologia linear ou planar no cérebro. Dois exemplos reais são o mapa tonotópico (freqüências sonoras são mapeadas espacialmente no córtex em uma progressão das freqüências mais baixas até as mais altas) e o mapa retinotópico (campo visual é mapeado no córtex visual com maior resolução para o centro).

Existe um tipo de rede neural que implementa os conceitos de auto-organização cerebral. Estas redes são conhecidas como Redes Auto-Organizáveis = Self-Organizing Feature Maps (SOFMs). Outros nomes comuns para elas são memórias associativas de filtro competitivas ou Redes de Kohonen, em homenagem ao seu criador, Dr. Eng. Teuvo Kohonen.

Uma rede de Kohonen consiste em duas camadas (uma de entrada e uma competitiva), como podemos ver na figura 9. Cada neurônio da camada de entrada representa uma dimensão do padrão de entrada e distribui seu componente vetorial para a camada competitiva.

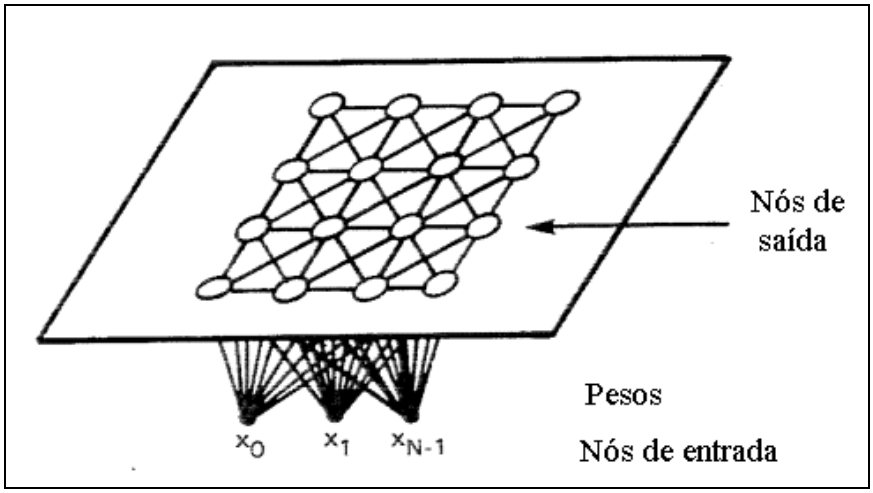

Figura 9. Exemplo da topologia de uma rede de Kohonen. Os neurônios da camada de entrada estão ligados a todos os neurônios da camada de saída (competitiva) através de pesos. Já os neurônios da camada de saída estão ligados a alguns neurônios da mesma camada, e estes constituem a sua vizinhança (BRAGA, 2000).

Cada neurônio na camada competitiva recebe a soma ponderada das entradas e tem uma vizinhança de *k* neurônios, vizinhança esta que pode ser organizada em 1, 2 ou n dimensões Ao receber uma entrada, alguns neurônios serão excitados o suficiente para disparar. Cada neurônio que dispara pode ter um efeito excitatório ou inibitório em sua vizinhança.

Após a inicialização dos pesos sinápticos, três processos básicos se seguem

- Competição: O maior valor da função é selecionado (vencedor). O neurônio vencedor é dado por: $i(x) = \arg\min \|x-w_j\|$, $j = 1,2,3,..,l$. O vencedor determina a localização do centro da vizinhança dos neurônios a serem treinados (excitados).
- Cooperação: Os vizinhos do neurônio vencedor são selecionados e excitados através de uma função de vizinhança.
- Adaptação Sináptica: Os neurônios excitados ajustam seus pesos sinápticos quando um neurônio vence uma competição. Não apenas ele, mas também todos os nodos localizados em sua vizinhança são ajustados, de acordo com uma função mostrada na figura 10.

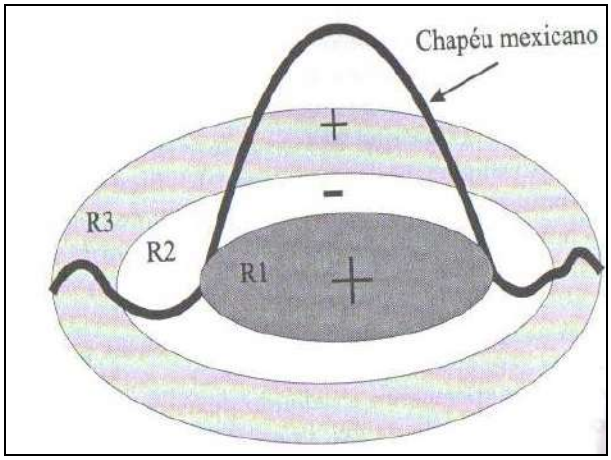

Figura 10. Saída dos elementos da vizinhança do neurônio vencedor, para efeito de ajuste dos pesos sinápticos. Note que os elementos mais próximos são ajustados

positivamente enquanto que os mais distantes são inibidos. Os elementos muito distantes recebem um pequeno ajuste positivo de forma a evitar que os neurônios outliers nunca sejam ajustados (BRAGA, 2000).

O algoritmo pode ser dado então pelos seguintes passos:

Comece com pesos aleatórios pequenos

Repita até estabilizar ou até o número de iterações exceder τ

Normalize os pesos

Aplique uma entrada I^u para a rede

Encontre o neurônio vencedor , i^*

O ajuste dos pesos do neurônio i* e de sua vizinhança é dado por:

$$\Delta w_j = \eta O(I_j^u - w_j),$$

O é a saída do neurônio calculada de acordo com o seguinte critério:

- a saída de i* é 1,
- a de sua vizinhaça é dada pela função de chapéu mexicano da figura 2.10),

j é o índice da coordenada, j=1,2,...,p.

As redes de Kohonen são excelentes para problemas de classificação. O problema delas é que o sistema é uma caixa preta, sem garantia de convergência para redes de dimensões maiores, além do que o treinamento pode ser extremamente longo para problemas grandes de classificação. Por outro lado, elas têm a vantagem de não só descobrir os conjuntos como também mostrar uma ordenação entre eles, dada a noção de vizinhança entre neurônios (OJA *et al*, 2002), o que pode ser útil em muitas aplicações.

2.4 Métodos baseados em grafos

Grafo é um modelo matemático que representa relações entre objetos. Um grafo é um conjunto dado por *G=(V, E)*, onde *V* é um conjunto finito de pontos, normalmente denominados de nós ou vértices e *E* é uma relação entre vértices, ou seja, um conjunto de pares em *V*x*V*. Na figura 2.11 podemos ver alguns exemplos de grafos. No exemplo (a), temos um grafo com 5 nós, e o conjunto V={1,2,3,4,5}. Estes nós estão ligados entre si por algumas arestas, definindo o conjunto E={ (1,1), (1,2), (3,4), (3,5), (4,5) }. Um grafo pode ser representado por uma matriz *A* de adjacências, onde A_{ij}=0 se não existe a aresta (i,j) e 1, caso a aresta em questão exista, como pode ser visto na figura 11.

O número de arestas que saem ou entram em um determinado vértice é denominado o grau ou a conectividade deste vértice. Por exemplo, na figura 11(a), o vértice 4 tem conectividade 2, enquanto que na figura (b), o vértice 4 tem conectividade igual a 3.

Os vértices de um grafo podem receber valores ou rótulos, e neste caso o grafo é denominado de rotulado. Os rótulos podem representar a força da conexão entre dois elementos. Os exemplos da figura 12 (a) e (b) são não rotulados, enquanto o exemplo da figura 11 (c) é um grafo rotulado, no qual, por exemplo, a aresta entre os nós 3 e 4 tem rótulo igual a 8.

As arestas de um grafo podem ter uma direção, e neste caso um grafo é chamado de direcionado. Caso contrário, o grafo é denominado não-direcionado. Para determinar a direção, representam-se as arestas com setas na ponta destino. No caso de um grafo não-direcionado, as arestas (e_1,e_2) e (e_2,e_1) são idênticas, o que não acontece em um grafo direcionado. Na figura 11(b) vemos um grafo direcionado e pode-se ver como foi necessário adicionar uma aresta (4,1) além da aresta (1,4). Se além de direcionado o grafo for rotulado, ele é chamado de uma rede.

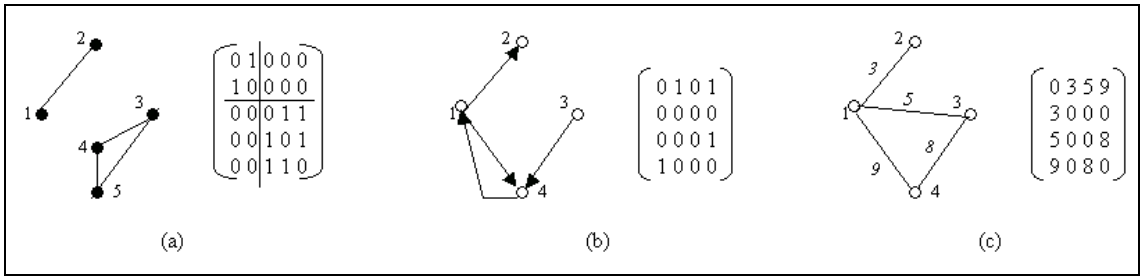

Figura 11. Exemplos de grafos. (a) Um grafo não direcionado, não conectado, não rotulado e de clique. (b) um grafo direcionado, não rotulado e não conectado. Este grafo não é uma rede, pois suas arestas não são rotuladas. (c) um grafo não direcionado, conectado e rotulado.

Um grafo é dito conectado se existe um caminho entre quaisquer dois nós do grafo, onde um caminho é uma seqüência de arestas { (e_1,e_2), (e_2,e_3), ..., (e_{k-1},e_k)}, onde e_1 é a origem e e_k é o destino. O exemplo da figura 11(c) é conectado, mas os exemplos (a) e (b) não o são. No primeiro caso, não é possível encontrar um caminho entre os nós 1 e 4, enquanto que no segundo não existe um caminho saindo do nó 2 e chegando no nó 1 (a aresta existente é direcionada e sai de 1 e chega em 2, e não vice-versa).

Um grafo é dito completo se todos os pares de vértices estão conectados por uma aresta. Um grafo de cliques é um grafo em que cada componente conectado é um grafo completo. O exemplo da figura (a) é um grafo de clique, pois o grafo pode ser separado em dois sub-grafos G_1 e G_2, como pode ser visto pela separação da matriz em sub-matrizes (G_1 é representado pela matriz no canto superior esquerdo, enquanto G_2 é representado pela matriz no canto inferior direito. As outras duas matrizes têm todos os elementos iguais a zero, indicando que os sub-grafos não são concetados). O primeiro contém os vértices 1 e 2 enquanto que o segundo contém os vértices 3,4,5. Note que cada um dois vértices em cada um dos grafos está ligado por uma aresta a todos seus companheiros de grafo.

Uma maneira de agrupar elementos consiste em definir um grafo baseado nos seus dados e supor que o grafo obtido representa uma clique corrompida. Desta maneira basta procurar o número mínimo de arestas a serem adicionadas ou retiradas de forma a obter um grafo de clique. Os sub-grafos completos obtidos representarão então os clusters mais adequados para os dados. Um exemplo de tal técnica pode ser visto na figura 12.

Infelizmente este problema, denominado problema das cliques corrompidas, é NP-difícil, o que implica não haver nenhum algoritmo prático para resolvê-lo. Felizmente existem heurísticas para nos ajudar. Vamos discutir uma delas, mas antes vamos entender como podemos criar o grafo no qual a heurística será aplicada.

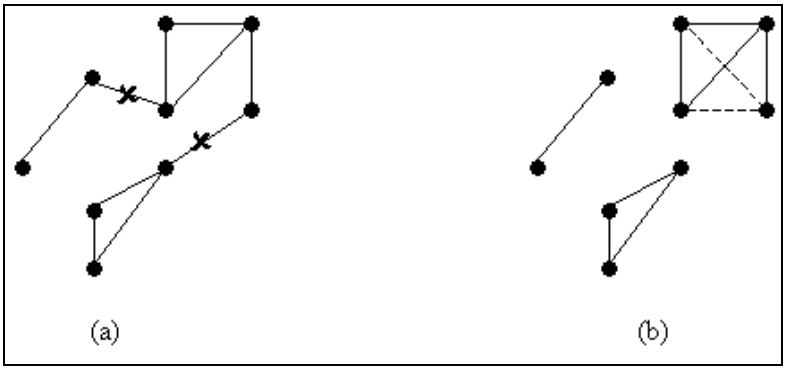

Figura 12. Exemplos de obtenção de cliques minimizando as alterações de um grafo. (a) O grafo original. As arestas marcadas por X serão apagadas. (b) O grafo de cliques obtido. As arestas pontilhadas foram inseridas. O número mínimo de alterações para obter este grafo de cliques é 4 (duas inserções e duas exclusões).

Se as distâncias entre dois elementos foram calculadas usando qualquer uma das métricas definidas na seção 1.2, pode-se montar uma matriz simétrica D em que cada elemento D_{ij} representa a distância entre os elementos i e j. Pode-se então escolher arbitrariamente um limite θ tal que existirá uma aresta entre os nós i e j se a distância d_{ij} for menor do que θ (maior, no caso de usarmos a correlação). Um exemplo da aplicação desta técnica pode ser visto na figura 13.

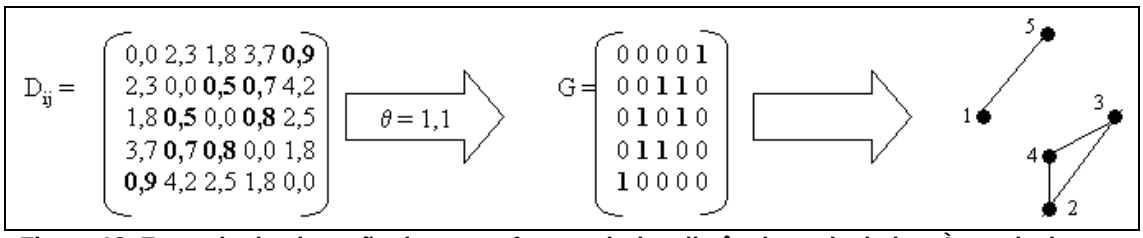

Figura 13. Exemplo de obtenção de um grafo a partir das distâncias calculadas. À matriz de distâncias obtida foi aplicado um threshold θ=1,1. Os elementos marcados em negrito estão abaixo desta distância. Gera-se então uma matriz de representação de um grafo com entradas iguais a um nos pontos marcados em negrito. O grafo representado por esta matriz pode ser visto na sua forma gráfica em seguida.

Um algoritmo extremamente prático para resolver o problema de cliques corrompidas é o CAST (Cluster Affinity Search Technique), definido inicialmente em (BEN-DOR *et al*, 1999). O algoritmo constrói iterativamente a partição P de um conjunto S encontrando um cluster C tal que nenhum elemento $i \in C$ seja distante de C e nenhum elemento $i \notin C$ seja próximo de C, dado um grafo formado a partir de um limite de distância θ. O algoritmo é o seguinte:

```
S=conjunto de vértices do grafo G
P=∅
Enquanto S≠∅
        C={v}, onde v = vértice de grau máximo no grafo G
        Acrescente a C todos os genes ligados a um elemento de C
        P=P+C
        S=S/C
        Remova vértices do cluster C do grafo G
Fim Enquanto
```

Existem vários outros algoritmos baseados em grafos, que podem ser considerados no lugar do CAST. Entre eles podemos citar o PCC, definido no mesmo artigo que o CAST (BEN-DOR *et al*, 1999), que se baseia em uma definição sobre as entradas de um modelo estocástico do erro e o CLICK (SHARAN *et al*, 2000), que também se baseia na definição de um modelo estatístico que designa um significado probabilístico para as arestas do grafo. Ambos os modelos também foram bastante usados e são bem mais complexos que o CAST, devendo, entretanto, ser considerados por todos aqueles que desejem expandir seus conhecimentos sobre métodos de agrupamento baseados em grafos.

3. Conclusão

Neste tutorial foram vistos os principais conceitos de agrupamento, várias heurísticas aplicadas a este problema e que podem ser aplicadas a diversas áreas de conhecimento.

Quando não existe nenhum tipo de informação pré-existente não existe uma classificação "ótima" para um conjunto de dados. Neste ponto, métodos de clustering servem como um ferramental

para exploração de dados que, se usados de forma adequada, podem particionar os dados sem o efeito de quaisquer preconceitos e noções arbitrárias. Dado o fato de que este problema é NP-difícil, é necessário aplicar uma heurística para resolver o problema.

O grande número de técnicas de agrupamento existentes pode ser confuso para o iniciante. Às vezes, em vez de escolher, o ideal pode ser usar todas de uma só vez. Existem trabalhos, tais como (HU *et al*, 2004) que usam simultaneamente várias técnicas de argupamento e combinam seus resultados para tomar uma decisão final.

Um ponto que deve ser realçado sempre quando se lida com dados de um campo de conhecimento, não se deve descartar o conhecimento desta área. Todos os dados e resultados devem ser analisados à luz do conhecimento existente: nenhum algoritmo deve ser usado em qualquer área de forma isolada, sem pensar nas ramificações das conclusões que o seu uso pode impor.

Todas estas questões são importantes, não só para técnicas de agrupamento como para outros algoritmos que se usem na área de aplicação que se deseja. Muitas vezes, as técnicas de agrupamento servirão como entrada para outros algoritmos ou como preparação para experimentos.

4. Referências

BEN-DOR, A., SHAMIR, R., YAKHINI, Z. (1999), "*Clustering Gene Expression Patterns*", Journal of Computational Biology, v. 6, n. 3, pp. 281-297.

BRAGA, A. P, CARVALHO, A. C. P. L. F., LUDERMIR, T. B., 2000, "Redes Neurais Artificiais – Teoria e Aplicações", 1ª. Edição, LTC Editora , Brasil.

CHIANG I.W-Y.; LIANG G-S.; YAHALOM S.Z (2003), "The fuzzy clustering method: Applications in the air transport market in Taiwan", The Journal of Database Marketing & Customer Strategy Management, v 11, n 2, pp. 149-158

DZWINEL, W., YUEN, D. A., BORYCZKO, K., *et al.* (2005), "Nonlinear multidimensional scaling and visualization of earthquake clusters over space, time and feature space", Nonlinear Processes in Geophysics n. 12 pp. 117–128

GORDON, A. D, 1981, Classification, Chapman and Hall Ed., 1981

HAIR, J F JR., ANDERSON, R. E. *et al*, Multivariate Data Analysis, Prentice Hall, 1995

HAMMOUDA, K. M. (2002), "Web Mining: Identifying Document Structure for Web Document Clustering", Tese de Mestrado, Department of Systems Design Engineering, University of Waterloo, Canada

HAYKIN, S., Redes Neurais: Princípios e Práticas, Ed. Bookman, 2001

HEBB, D. O. (1949). "The organization of behavior: A neuropsychological theory" New York: Wiley.

HU, X., ILLHOY, Y. (2004), **"**Cluster ensemble and its applications in gene expression analysis**",**ACM International Conference Proceeding Series; Vol. 55 Proceedings of the second conference on Asia-Pacific bioinformatics – v. 29, pp. 297-302 , Nova Zelândia.

JACKSON, C. R., DUGAS, S. L. (2003), "Phylogenetic analysis of bacterial and archaeal *arsC* gene sequences suggests an ancient, common origin for arsenate reductase", *BMC Evolutionary Biology* 2003, v. 3 n. 18

JIANG, D., ZHANG, A. (2002) "Cluster Analysis for Gene Expression Data: A Survey", Universidade de Nova Iórque.

JONES, N. C., PEZNER, P., "An Introduction to Bioinformatics Algorithms", MIT Press, EUA, 2004

KIM, J., WARNOW, T. (1999), "Tutorial on Phylogenetic Tree Estimation", Universidade de Yale.

LEVIA JR D. F., PAGE D. R. (2000), "The Use of Cluster Analysis in Distinguishing Farmland Prone to Residential Development: A Case Study of Sterling, Massachusetts", Environ Manage. V. 25 n. 5, pp. 541-548

LINDEN, R.,2005, "Um Algoritmo Híbrido Para Extração De Conhecimento Em Bioinformática", Tese de Doutorado, COPPE-UFRJ, Rio de Janeiro, Brasil

MARTÍNEZ-DÍAZ, R. A., ESCARIO, J. E., NOGAL-RUIZ, J. J., *et al* (2001), "Relationship between BiologicaI Behaviour and Randomly Amplified Polymorphic DNA Profiles of *Trypanosoma cruzi* Strains", Mem. Inst. Oswaldo Cruz vol.96 no.2, pp. 251-256, Brasil

MATIOLI, S. R. (ed.), Biologia Molecular e Evolução, Holos Editora, 2001

OJA, M., NIKKILA, J., TORONEN, P., *et al*, 2002, "Exploratory clustering of gene expression profiles of mutated yeast strains" in *Computational and Statistical Approaches to Genomics, Zhang, W. e Shmulevich, I.* (eds), pp. 65-78, Kluwer Press

SHARAN, R., SHAMIR R 2000. "CLICK: A clustering algorithm with applications to gene expression analysis", Proc. Int. Conf. Intell. Syst. Mol. Biol. **8:** 307–316.

YEOH, E., ROSS, M. E., SHURTLEFF, S. A. *et al.*, 2002, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling", Cancer Cell, v. 1, n. 1, pp. 133-143.