UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Algoritmos de Agrupamento de Dados

Katti Faceli
André C P L F de Carvalho
Marcilio Carlos Pereira de Souto

Nº 249

RELATÓRIOS TÉCNICOS

São Carlos – SP
Jan./2005

Algoritmos de Agrupamento de Dados

Katti Faceli
André C.P.L.F. de Carvalho

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {katti, andre}@icmc.usp.br

Marcílio Carlos Pereira de Souto

Universidade Federal do Rio Grande do Norte
Departamento de Informática e Matemática Aplicada - DIMAp
Campus Universitario
59072-970 - Natal, RN, Brazil
e-mail: marcilio@dimap.ufrn.br

Resumo

A idéia básica dos algoritmos de agrupamento de dados é reunir uma série de objetos em grupos, ou clusters, de objetos semelhantes ou relacionados.

Algoritmos de agrupamento são ferramentas valiosas na análise exploratória de dados, mineração de dados e reconhecimento de padrões. Tais algoritmos fornecem um meio de explorar e verificar estruturas presentes nos dados, organizando-os em grupos ou clusters.

O processo de agrupamento envolve diversas etapas que vão desde a preparação dos dados, até a interpretação dos clusters obtidos, passando pela escolha da medida de similaridade, execução do algoritmo de agrupamento propriamente dito e validação dos resultados.

Palavras-Chave: Agrupamento, medidas de similaridade, validação.

Janeiro 2005
Draft impresso em 11 de janeiro de 2005

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Algoritmos de agrupamento são ferramentas valiosas na análise exploratória dos dados, mineração de dados e reconhecimento de padrões. Tais algoritmos fornecem um meio de explorar e verificar estruturas presentes nos dados, organizando-os em grupos, ou clusters (Fred 2001).

A Seção 2 contém uma discussão sobre agrupamentos e seus aspectos principais. A preparação dos dados, englobando pré-processamento e representação apropriada para utilização com um algoritmo de agrupamento, é descrita na Seção 3. Algumas das medidas de similaridade mais empregadas em agrupamento são descritas na Seção 4. Diversos algoritmos de agrupamento de potencial interesse para este trabalho são detalhados na Seção 5. A Seção 6 contém uma descrição dos critérios e de alguns dos índices de validação descritos na literatura. A Seção 7 resume os principais aspectos referentes à análise e comparação de algoritmos de agrupamento.

2 Definição e Aspectos Principais

O termo cluster não tem uma definição precisa. A variedade de definições desse termo resulta da visão e objetivos dos pesquisadores de diversas áreas e também das diversas aplicações de agrupamento. Algumas definições comuns são (Barbara 2000):

Cluster bem separado: um cluster é um conjunto de pontos tal que qualquer ponto em um cluster está mais próximo (ou é mais similar) a cada outro ponto no cluster do que a qualquer ponto que não pertence ao cluster.

Cluster baseado em centro: um cluster é um conjunto de pontos tal que qualquer ponto em um cluster está mais próximo (ou é mais similar) ao centro do cluster do que ao centro de qualquer outro cluster. O centro de um cluster pode ser um centróide, como a média dos pontos do cluster, ou um medóide, o ponto mais representativo do cluster.

Cluster contínuo (vizinho mais próximo ou agrupamento transitivo): um cluster é um conjunto de pontos tal que qualquer ponto em um cluster está mais próximo (ou é mais similar) a um ou mais pontos no cluster do que a qualquer ponto que não pertence ao cluster.

Cluster baseado em densidade: um cluster é uma região densa de pontos, separada de outras regiões de alta densidade por regiões de baixa densidade.

Cluster baseado em similaridade: um cluster é um conjunto de pontos que são similares, enquanto pontos de clusters diferentes não são similares.

Uma noção intuitiva do que é um cluster resulta em um princípio indutivo. A formulação matemática de um princípio indutivo, chamada critério de agrupamento ou função objetivo, consiste de uma forma de selecionar uma estrutura (ou modelo) para representar os clusters que melhor se ajuste a um determinado conjunto de dados (Estivill-Castro 2002). Colocado de outra maneira, o critério de agrupamento é uma forma de

expressar o objetivo do agrupamento. Esse critério, geralmente, é baseado na definição de cluster empregada e/ou em uma distribuição esperada dos dados em um domínio de aplicação específico (Jiang et al. 2004).

Um princípio indutivo associado a um conjunto de dados resulta em um problema de otimização. Em geral, esse problema de otimização é intratável ou tem uma complexidade muito alta para ser resolvido na prática para conjuntos de dados grandes. Por isso, o problema é resolvido de forma aproximada por algum algoritmo heurístico que seja adequado, fazendo um balanço entre a qualidade da otimização e o esforço computacional (Estivill-Castro 2002). Geralmente, esse algoritmo define uma medida de proximidade e um método de busca para encontrar uma partição ótima ou sub-ótima nos dados, de acordo com o critério de agrupamento (Jiang et al. 2004).

Independente da variedade de definições de um cluster, que resultam em diferentes critérios, a idéia básica de agrupamento é reunir uma série de objetos em grupos, ou clusters, de objetos semelhantes ou relacionados. O agrupamento geralmente está associado com a análise exploratória de um conjunto de dados, envolvendo problemas que possuem pouca informação prévia (como modelos estatísticos) disponível a respeito dos dados. Assim, agrupamento é particularmente apropriado para explorar as inter-relações entre os pontos de dados e fazer uma avaliação da sua estrutura.

O processo de agrupamento envolve diversas etapas que vão desde a preparação dos padrões, até a interpretação dos clusters obtidos. Dependendo do objetivo que se deseja atingir com o agrupamento, a etapa de interpretação dos clusters pode ser suprimida. A Figura 1 resume as etapas do processo de agrupamento com as informações utilizadas e geradas em cada etapa. As etapas e a figura apresentada são baseadas nas informações apresentadas por Jain et al. (1999) e Barbara (2000).

Preparação dos padrões:

Envolve a determinação da forma de representação dos padrões e a aplicação de transformações nos dados, como normalizações e seleção ou extração de características. Os principais aspectos relacionados a essa etapa são detalhados na Seção 3.

Medida de similaridade:

Esta etapa consiste da definição de uma medida de similaridade apropriada ao domínio da aplicação. Em geral, uma medida de similaridade é fornecida por uma função de distância definida entre pares de padrões. É possível incluir na medida de distância aspectos conceituais (qualitativos) ou numéricos (quantitativos). A Seção 4 contém uma descrição dos principais aspectos relacionados às medidas de similaridade.

Realização do agrupamento:

Esta etapa consiste da aplicação de um algoritmo de agrupamento apropriado para agrupar os dados de acordo com um objetivo específico. Existem inúmeros algoritmos que podem ser aplicados nesta etapa. As respostas desta fase podem ser *hard*

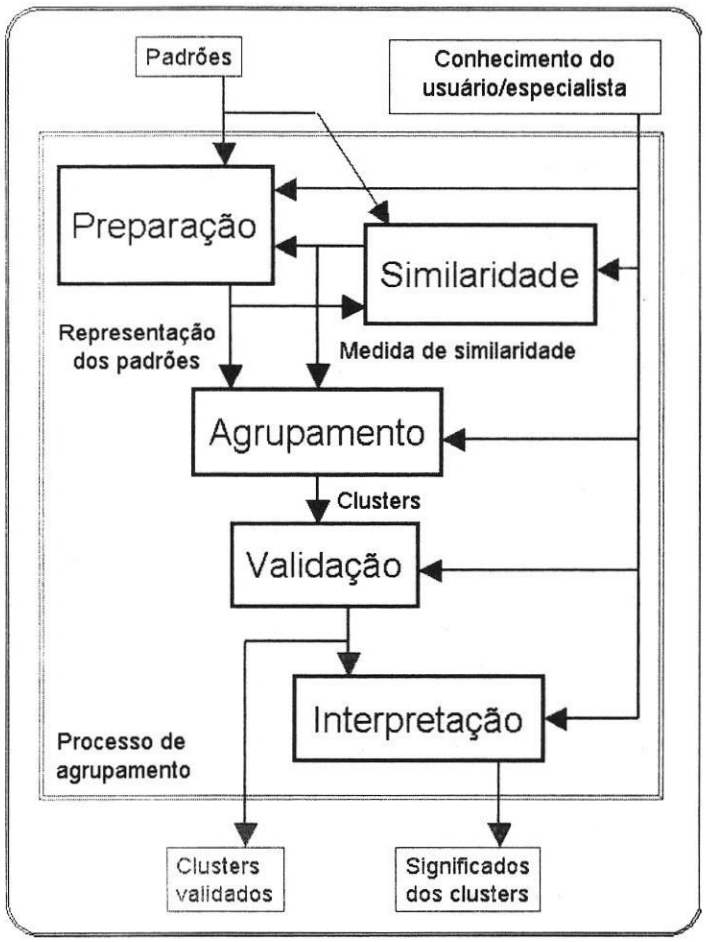

Figura 1: Etapas do processo de agrupamento.

(um exemplo pertence ou não pertence a um dado cluster) ou *fuzzy* (cada exemplo tem um grau de pertinência à cada um dos clusters). Diversos algoritmos de agrupamento são apresentados e comparados na Seção 5.

Validação:

Esta etapa se refere a avaliação da validade dos resultados obtidos. Esta avaliação é objetiva e visa determinar se o resultado é significativo. A estrutura resultante de um agrupamento é válida se não ocorreu por acaso ou como um artefato do algoritmo de agrupamento empregado. As principais formas de validação são resumidas na Seção 6.

Interpretação:

Refere-se ao processo de examinar cada cluster com relação a seus exemplos para rotulá-los descrevendo a natureza do cluster. A interpretação de clusters é mais que apenas uma descrição. Além de ser uma forma de avaliação entre os clusters encontrados e a teoria inicial, de um modo confirmatório, os clusters podem permitir avaliações subjetivas que tenham um significado prático. Ou seja, o especialista pode ter interesse em encontrar diferenças semânticas de acordo com os padrões e valores

de seus atributos em cada cluster.

O problema da maioria das abordagens de agrupamento é que elas podem produzir diferentes agrupamentos a partir de um único conjunto de dados (Zeng et al. 2002). Disso surgem algumas questões. Qual resultado é melhor? Quanto se pode confiar nesse resultado? Existe um resultado que seja melhor do que os outros? Se existe, como obtê-lo? Se não, é possível combinar todos os resultados disponíveis para ter um entendimento melhor dos dados?

É provado que a maioria dos problemas de agrupamento é NP (Não Determinístico Polinomial), o que significa que eles são intratáveis ou não computáveis em um tempo razoável. Como já foi dito, todas as abordagens disponíveis são heurísticas e podem fornecer apenas uma aproximação do resultado ótimo (Zeng et al. 2002). Além disso, apesar do grande número de algoritmos de agrupamento existentes, não existe uma técnica de agrupamento universal, capaz de revelar toda a variedade de estruturas que podem estar presentes em um conjunto de dados. Como lembra Hartigan (1985), "diferentes agrupamentos são corretos para diferentes propósitos, assim, não podemos dizer que um agrupamento é melhor". A definição da medida de similaridade e do critério de agrupamento dos algoritmos, geralmente dependem implicitamente da imposição de certas hipóteses a respeito da forma dos clusters ou da configuração dos múltiplos clusters. Outro aspecto importante é que os dados dificilmente estão estruturados "idealmente", ou seja, não formam configurações hiperesféricas, hiperelipsoidais, lineares, etc., de modo que cada algoritmo de agrupamento pode apresentar um comportamento superior aos demais para uma dada conformação específica dos dados no espaço de atributos.

3 Preparação dos Padrões

A preparação dos dados envolve vários aspectos relacionados ao seu pré-processamento e à forma de representação apropriada para sua utilização por um algoritmo de agrupamento. O pré-processamento pode envolver, por exemplo, normalizações, conversão de tipos e redução do número de atributos por meio de seleção ou extração de características (Jain et al. 1999). Para isso, o número de padrões e o número, tipo e escala das características do conjunto de dados são informações bastante úteis. Vários trabalhos discutem formas de padronização dos dados, seleção de atributos e outros aspectos relativos à preparação dos dados, como os de Jain & Dubes (1988), Gordon (1999), He (1999), Jain et al. (1999), Barbara (2000) e Berkhin (2002).

Os objetos a serem agrupados podem representar um objeto físico, como uma cadeira, ou uma noção abstrata, como um estilo de escrita. Tais objetos, também chamados padrões (exemplos, amostras ou pontos) são representados, geralmente, por um vetor de características ou atributos. Duas questões importantes no que se refere às características são a escolha das mais relevantes para o agrupamento e a definição do número desejável de atributos em aplicações dos algoritmos de agrupamento. Em resumo, o problema básico é encontrar um conjunto de características que melhor representa o conceito de similaridade com que se está lidando. Para resolver essa questão, podem ser aplicadas técnicas de seleção e/ou extração de características.

A seleção de características é o processo de identificação do subconjunto mais efetivo dos atributos disponíveis para descrever cada padrão. A extração de características se refere ao uso de uma ou mais transformações junto aos atributos de entrada de modo a salientar uma ou mais característica dentre aquelas que estão presentes nos dados.

O conhecimento do tipo e escala das características dos padrões é importante na escolha da medida de similaridade e também do algoritmo a serem empregados em um agrupamento, bem como na interpretação dos resultados. Existem medidas de similaridade apropriadas para cada tipo/escala de atributo. O tipo de um atributo diz respeito ao grau de quantização nos dados. A escala indica a significância relativa dos números. Em relação à escala, as características podem ser quantitativas ou qualitativas (Jain & Dubes 1988). Os tipos possíveis dos atributos são (Jain & Dubes 1988; Barbara 2000):

Binários: os atributos binários apresentam apenas dois valores, como sim/não e verdadeiro/falso.

Discretos: os atributos discretos apresentam um número finito, geralmente pequeno, de valores possíveis.

Contínuos: os atributos contínuos representam qualquer valor real.

As diferentes escalas são (Jain & Dubes 1988; Barbara 2000):

Qualitativa

Nominal: os valores são apenas nomes diferentes. Exemplos: CEP, cores, sexo.

Ordinal: os valores refletem somente uma ordem. Exemplos: hierarquia militar, avaliações qualitativas de temperatura como frio, morno e quente.

Quantitativa

Intervalo: a diferença entre os valores é significativa, isto é, existe uma unidade de medida. Exemplos: temperatura (90° Célsius é diferente de 90° Fahrenheit), a duração de um evento, em minutos.

Relação: os números tem um significado absoluto. Isto significa que existe um zero absoluto junto com uma unidade de medida, de forma que proporção tenha significado. Exemplos: altura, salário, distância.

Em muitos casos, é necessária a aplicação de algumas transformações antes da utilização dos dados. Muitas vezes, os diferentes atributos que representam os padrões se apresentam em escalas diferentes. Quando os intervalos de valores dos atributos diferem muito, pode ser que um atributo domine o resultado do agrupamento. Para solucionar esse problema, é comum a padronização dos dados de forma que os atributos estejam na mesma escala. Barbara (2000) descreve algumas formas de padronização dos dados.

Outro aspecto a ser considerado é a forma de representação dos dados a serem agrupados. Geralmente, assume-se que a representação adequada dos padrões está disponível

para a aplicação de um algoritmo de agrupamento. No entanto, uma investigação cuidadosa das característica disponíveis e das transformações que podem ser aplicadas aos dados pode auxiliar na obtenção de resultados significativamente melhores. Um exemplo simples é o agrupamento dos pontos da Figura 2. Os padrões representados na figura formam um cluster curvilíneo com distância da origem aproximadamente constante. Utilizando uma representação em coordenadas Cartesianas, muitos algoritmos de agrupamento produziriam dois ou mais clusters. Entretanto, se fossem utilizadas coordenadas polares para representar os padrões, uma solução de um único cluster poderia ser obtida mais facilmente (Jain et al. 1999).

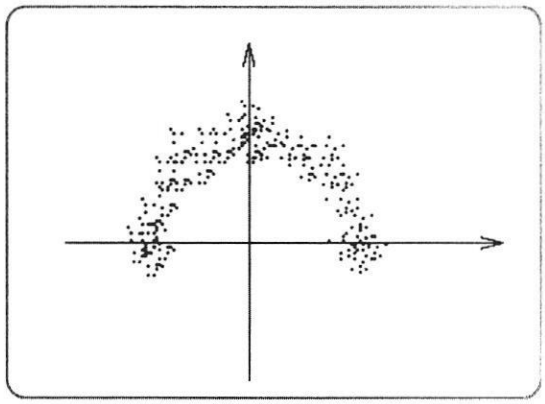

Figura 2: Cluster curvilíneo com pontos aproximadamente equidistantes da origem (Jain et al. 1999).

Na maioria dos casos, os dados brutos a serem submetidos a um algoritmo de agrupamento são representados por uma matriz de padrões $X_{n\times d}$, em que n é o número de padrões e d é o número de atributos que representam os padrões, isto é, é a dimensionalidade dos padrões ou do espaço de padrões. Cada elemento dessa matriz, X_{ij}, contém o valor da j-ésima característica para o i-ésimo padrão. Para muitos algoritmos, os dados são considerados como pontos no espaço de características. As d características podem ser vistas como um conjunto de eixos ortogonais. Os padrões são pontos no espaço de dimensão d, chamado espaço de padrões. Neste sentido, um cluster pode ser visto como uma coleção de padrões próximos ou que satisfazem alguma relação espacial. Neste trabalho, um padrão será denotado por x_i.

Algumas vezes, apenas a relação de proximidade entre os padrões é conhecida. Além disso, alguns algoritmos de agrupamento requerem uma forma de representação específica. Além da matriz de padrões, outras duas formas de representação dos padrões bastante comuns são a matriz e o grafo de similaridade ou proximidade.

Uma matriz de similaridade $S_{n\times n}$, contém os valores da similaridade/dissimilaridade entre dois padrões i e j, representados respectivamente na linha i e coluna j da matriz. Os valores de S_{ij} podem representar diretamente os dados brutos ou serem calculados pela aplicação de uma medida de similaridade aos dados representados na forma de matriz de padrões.

Uma matriz de proximidade define um grafo ponderado, em que os nós são os padrões

a serem agrupados e as arestas com pesos representam os valores de proximidade entre os padrões. Sob o ponto de vista de um grafo, realizar um agrupamento é equivalente a quebrar o grafo em componentes conectados, cada um representando um cluster. Muitos dos algoritmos de agrupamento são naturalmente descritos usando uma representação de grafo.

4 Medidas de Similaridade

Uma medida de similaridade ou proximidade é uma medida que indica o quão similares são dois padrões. A medida de similaridade a ser empregada com um algoritmo de agrupamento deve ser escolhida cuidadosamente devido à grande variedade de tipos e escalas das características. As medidas, em geral, consideram que todas as características contribuem igualmente para a proximidade. Gordon (1999) apresenta algumas questões relacionadas à escolha da medida de similaridade a ser empregada.

Segundo He (1999), existem pelo menos três conceitos de similaridade que precisam ser considerados: a similaridade entre entidades (padrões), a similaridade entre uma entidade e um grupo de entidades e similaridade entre dois grupos de entidades. Nesta seção são descritas medidas de similaridade entre padrões e entre grupos de padrões. Na realidade, as medidas podem se referir à similaridade ou dissimilaridade. As medidas mais comuns empregadas em agrupamento calculam a dissimilaridade. Essas medidas são as distâncias.

Normalmente, as medidas de proximidade devem satisfazer algumas propriedades. As medidas que satisfazem todas as propriedades são chamadas métricas. Porém, nem todas as medidas de similaridade empregadas são métricas. Quando medidas de similaridade não satisfazem as propriedades 4 e 5, elas não são consideradas métricas.

1. Para dissimilaridade: $S_{ii} = 0$ para todo i (Os pontos não são diferentes de si próprios)

   Para similaridade: $S_{ii} > \max S_{ij}$ (Os pontos são mais similares a si próprios)

2. $S_{ij} = S_{ji}$ (Simetria)

3. $S_{ij} \geq 0$ para todo i e j (Positividade)

4. $S_{ij} = 0$ somente se $i = j$

5. $S_{ik} \leq S_{ij} + S_{jk}$ para todo i, j e k (Desigualdade triangular)

Dependendo do tipo e da escala dos atributos, um conjunto de medidas de similaridade diferentes pode ser empregado. As medidas mais comuns para conjuntos de dados em que todos os atributos são contínuos e cuja escala é do tipo relação são as distâncias baseadas na métrica de Minkowski, como a distância Euclideana, a distância de Manhattan e a distância *supremum*. Quando todos os atributos são binários, é comum a utilização da distância de Manhattan, que neste contexto é chamada de distância de Hamming. Para dados binários e nominais existem diversos coeficientes de casamento (*matching*), como o coeficiente de casamento simples e o coeficiente de Jaccard. Existem ainda indices probabilísticos.

Gordon (1999) apresenta diversas medidas que são mais apropriadas para padrões cujos atributos são todos de um mesmo tipo. Ele classifica as métricas de acordo com o tipo das características para as quais a medida é apropriada. As medidas relacionadas aos outros tipos de atributos são descritas brevemente, porém o foco deste trabalho são as medidas para atributos quantitativos.

Atributos quantitativos: As medidas separação angular e correlação de Pearson, que são medidas de correlação, medem o coseno do ângulo entre dois vetores, sendo medidos, respectivamente da origem e da média dos dados.

Métricas de Minkowski: essas métricas são derivadas da Equação 1, de acordo com um valor escolhido para p, com $1 \leq p < \infty$. Chamadas de distâncias L_p, medem a dissimilaridade entre os padrões. Os menores valores de p correspondem a estimativas mais robustas (menos sensíveis a *outliers*). As métricas de Minkowski são sensíveis a variações de escala dos atributos, isto é, atributos representados em uma escala maior tendem a dominar os outros. Isso pode ser solucionado pela normalização dos atributos para um intervalo ou variância comum, ou pela aplicação de outros esquemas de ponderação (Jain et al. 1999).

$$S_{ij} = (\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p)^{1/p} \tag{1}$$

Alguns valores de p definem métricas bastante conhecidas.

- $p = 1$: **Distância de Manhattan** (ou distância bloco-cidade), dada pela Equação 2.

$$S_{ij} = \sum_{k=1}^{d} |x_{ik} - x_{jk}| \tag{2}$$

- $p = 2$: **Distância Euclideana**, dada pela Equação 3. Esta métrica tem um significado de variância total entre clusters. É a medida de distância mais comum. Ela é apropriada para conjuntos de dados que possuem clusters compactos ou isolados.

$$S_{ij} = (\sum_{k=1}^{d} (x_{ik} - x_{jk})^2)^{1/2} \tag{3}$$

- $p = \infty$: **Distância *supremum***, dada pela Equação 4, calcula o máximo da diferença absoluta em coordenadas. Em outras palavras, é a diferença máxima entre quaisquer componentes dos vetores.

$$S_{ij} = \max_{1 \leq k \leq d} |x_{ik} - x_{jk}| \tag{4}$$

Métrica de Canberra: é dada pela Equação 5. Esta métrica é muito sensível à pequenas mudanças próximas à $x_{ik} = 0 = x_{jk}$. Já possui uma padronização embutida.

$$S_{ij} = \sum_{k=1}^{d} |x_{ik} - x_{jk}|/(|x_{ik}| + |x_{jk}|) \tag{5}$$

Separação angular ou coseno: é dada pela Equação 6. Pode-se também obter e utilizar o ângulo como distância, a partir do coseno.

$$S_{ij} = \frac{\sum_{k=1}^{d} x_{ik}x_{jk}}{(\sum_{k=1}^{d} x_{ik}^2 \sum_{l=1}^{d} x_{jl}^2)^{1/2}} \tag{6}$$

Coeficiente de correlação de Pearson: é dado pela Equação 7, em que $\bar{x_i} = \sum_{k=1}^{d} x_{ik}/d$. Os valores dessa medida estão no intervalo $[-1, 1]$. O coeficiente de correlação é frequentemente descrito como uma medida da forma, no sentido de que é insensível a diferenças na magnitude dos atributos. É sensível à *outliers* e é menos intuitivo do que a distância Euclideana, por exemplo.

$$S_{ij} = \frac{\sum_{k=1}^{d} (x_{ik} - \bar{x_i})(x_{jk} - \bar{x_j})}{(\sum_{k=1}^{d} (x_{ik} - \bar{x_i})^2 \sum_{l=1}^{d} (x_{jl} - \bar{x_j})^2)^{1/2}} \tag{7}$$

Distância de Mahalanobis: é dada pela Equação 8, em que C_{kl} é o elemento da k-ésima linha e l-ésima coluna da inversa da matriz de covariância. Esta distância incorpora a correlação entre características e padroniza cada característica para média zero e variância um. A idéia básica desta medida é associar diferentes pesos à diferentes características com base em suas variâncias e a correlação linear entre pares de padrões (Jain et al. 1999). Neste caso, assume-se implicitamente que as densidades condicionais das classes são unimodais e caracterizadas por um espalhamento multidimensional (Jain et al. 1999). Outras formas de utilização da distância de Mahalanobis são a quadrada e a regularizada. As distorções nas medidas causadas por correlação linear entre características são melhoradas com a aplicação dessa medida.

$$S_{ij} = (\sum_{k=1}^{d} \sum_{l=1}^{d} (x_{ik} - x_{jk}) C_{kl} (x_{il} - x_{jl}))^{1/2} \tag{8}$$

Coeficiente de Dice: é dado pela Equação 9.

$$S_{ij} = \frac{2x_i^T x_j}{\|x_i\|^2 + \|x_j\|^2} = \frac{2\sum_{k=1}^{d} x_{ik}x_{jk}}{\sum_{k=1}^{d} x_{ik} \sum_{l=1}^{d} x_{jl}} \tag{9}$$

Distância expoente: é dada pela Equação 10.

$$S_{ij} = \exp(-\|x_i - x_j\|^{\alpha}) = \exp(-(\sum_{k=1}^{d} (x_{ik} - x_{jk})^2)^{\alpha/2}) \tag{10}$$

Distância produto interno: é dada pela Equação 11.

$$S_{ij} = \sum_{k=1}^{d} x_{ik}x_{jk} \tag{11}$$

Atributos binários: As medidas descritas para atributos binários são derivadas das seguintes informações a respeito de dois padrões i e j:

a_{11}: número de atributos com valor 1 para ambos os padrões,

a_{00}: número de atributos com valor 0 para ambos os padrões,

a_{01}: número de atributos com valor 0 para o padrão i e valor 1 para o padrão j.,

a_{10}: número de atributos com valor 1 para o padrão i e valor 0 para o padrão j.

Coeficiente de casamento simples:

$$S_{ij} = \frac{a_{00} + a_{11}}{a_{00} + a_{11} + a_{01} + a_{10}} = \frac{a_{00} + a_{11}}{d} \tag{12}$$

Coeficiente de Jaccard:

$$S_{ij} = \frac{a_{11}}{a_{11} + a_{01} + a_{10}} = \frac{a_{11}}{d - a_{00}} \tag{13}$$

Atributos nominais e ordinais: As medidas para esses tipos de atributo focalizam a atenção na determinação da contribuição de cada variável. As medidas de similaridade entre pares de padrões são obtidas pela soma das contribuições individuais de todas as variáveis.

Similaridade nominal/ordinal geral: é dada pela Equação 14, em que s_{ijk} é a contribuição de cada padrão baseada em índices de discordância entre pares de estados dos atributos categóricos.

$$S_{ij} = \sum_{k=1}^{d} s_{ijk} \tag{14}$$

Atributos mistos: Essa medida é adequada para obter a similaridade entre padrões descritos por características de diferentes tipos, por se adequar a qualquer um dos tipos individualmente.

Coeficiente geral de Similaridade: é dado pela Equação 15, em que s_{ijk} é a contribuição do k-ésimo atributo para a similaridade e w_{ijk} é 0 ou 1, dependendo se a comparação para a variável k é válida ou não. O valor de s_{ijk} pode ser definido para os diferentes tipos de atributos.

$$S_{ij} = \sum_{k=1}^{d} w_{ijk} s_{ijk} / \sum_{k=1}^{d} w_{ijk} \tag{15}$$

Inúmeras outras medidas de similaridade/dissimilaride empregadas em diversas aplicações que utilizam agrupamento são citadas por Jain et al. (1999). Jiang et al. (2004) citam algumas medidas de similaridade comuns em agrupamento de dados de expressão gênica, tais como a distância Euclideana, o coeficiente de correlação de Pearson e o coeficiente de correlação *Spearman´s rank-order*.

A seguir, são descritas algumas das medidas de distância entre grupos de objetos. Dados n_k pontos de dimensão d em um Cluster $C_i = \{x_i | i = 1, ...n_k\}$. Algumas medidas de dissimilaridade (distância) entre clusters se baseiam nos conceitos de centróide, x_0, raio, R e diâmetro, D, da terminologia de espaço vetorial, dados respectivamente pelas equações 16, 17 e 18. R é distância média dos pontos do cluster ao centróide e D é a distância média entre pares (*pairwise average distance*) em um cluster.

$$x_0 = \frac{\sum_{i=1}^{n_k} x_i}{n_k} \tag{16}$$

$$R = (\frac{\sum_{i=1}^{n_k} (x_i - x_0)^2}{n_k})^{1/2} \tag{17}$$

$$D = (\frac{\sum_{i=1}^{n_k} \sum_{j=1}^{n_k} (x_i - x_j)^2}{n_k(n_k - 1)})^{1/2} \tag{18}$$

Dados dois clusters $C_1 = \{x_i | i = 1, 2, ...n_{k1}\}$ e $C_2 = \{x_j | j = n_{k1} + 1, n_{k1} + 2, ...n_{k1} + n_{k2}\}$, com os respectivos centróides x_{01} e x_{02}, podem ser definidas as seguintes distâncias entre dois clusters (Zhang et al. 1996):

Distância Euclideana do centróide:

$$D_0 = ((x_{01} - x_{02})^2)^{1/2} \tag{19}$$

Distância Manhattan do centróide:

$$D_1 = |x_{01} - x_{02}| \tag{20}$$

Distância média *inter-cluster*:

$$D_2 = (\frac{\sum_{i=1}^{n_{k1}} \sum_{j=n_{k1}+1}^{n_{k1}+n_{k2}} (x_i - x_j)^2}{n_{k1} n_{k2}})^{1/2} \tag{21}$$

Distância média *intra-cluster*:

$$D_3 = (\frac{\sum_{i=1}^{n_{k1}+n_{k2}} \sum_{j=1}^{n_{k1}+n_{k2}} (x_i - x_j)^2}{(n_{k1} + n_{k2})(n_{k1} + n_{k2} - 1)})^{1/2} \tag{22}$$

Distância *variance increase*:

$$D_4 = \sum_{k=1}^{n_{k1}+n_{k2}} (x_k - \frac{\sum_{l=1}^{n_{k1}+n_{k2}} x_l}{n_{k1} + n_{k2}})^2 - \sum_{i=1}^{n_{k1}} (x_i - \frac{\sum_{l=1}^{n_{k1}} x_l}{n_{k1}})^2 - \sum_{j=n_{k1}+1}^{n_{k1}+n_{k2}} (x_j - \frac{\sum_{l=n_{k1}+1}^{n_{k1}+n_{k2}} x_l}{n_{k2}})^2 \tag{23}$$

5 Exemplos de Algoritmos de Agrupamento

Existe uma grande variedade de algoritmos de agrupamento. Cada um desses algoritmos emprega um critério de agrupamento, que impõe uma estrutura nos dados. Se por acaso os dados estão em conformidade com as exigências do critério empregado, então a estrutura verdadeira de clusters é encontrada. Porém, apenas um número pequeno de critérios de agrupamento independentes podem ser entendidos sob os pontos de vista matemático e intuitivo. Assim, muitos dos critérios propostos na literatura são relacionados. Muitas vezes, os mesmos critérios aparecem representados sob diferentes "disfarces" (Jain & Dubes 1988).

Alguns critérios citados por Jain & Dubes (1988) são o erro quadrático, o ajuste de um modelo de densidade misto (*mixture density model*) aos padrões, estimativa de densidade, conectividade de grafos e vizinhos mais próximos.

Esta seção contém uma descrição de várias técnicas de agrupamento. Os algoritmos de agrupamento descritos foram desenvolvidos por pesquisadores de diferentes áreas, englobando abordagens da análise de dados estatística, reconhecimento de padrões, teoria dos grafos, entre outras.

Os algoritmos de agrupamento podem ser classificados por meio de diferentes aspectos. Uma das classificações bastante comum é a dada por Jain et al. (1999) e utilizada por Halkidi et al. (2001), em que os algoritmos são classificados de acordo com o método adotado para definir os clusters. Neste caso, os algoritmos são divididos em algoritmos hierárquicos, particionais, baseados em *grid* e baseados em densidade. Muitos dos algoritmos se enquadram em mais de uma dessas categorias. Nesta seção, algumas outras categorias serão também consideradas, não necessariamente baseadas no método adotado para definir os clusters.

São detalhados neste relatório apenas alguns dos algoritmos de agrupamento de potencial interesse para utilização em análise de dados de expressão gênica. Existe uma variedade muito maior de algoritmos. Uma tabela resumindo as principais características dos algoritmos será apresentada para cada um dos algoritmos apresentados. As principais características desses algoritmos são: manipulação de atributos numéricos, adequação para dados de alta dimensionalidade e grande número de padrões. Outro aspecto a ser considerado é a disponibilidade de um software para a aplicação do algoritmo. A Tabela 1 contém um resumo dos algoritmos que serão apresentados.

As informações dos algoritmos foram obtidas dos artigos originais ou de artigos que revisam e comparam diversos algoritmos. Outras informações, não disponíveis nessas fontes foram consultadas diretamente dos autores por e-mail.

Duas das principais divisões dos algoritmos de agrupamento são exclusivo × não exclusivo e hierárquico × particional (Jain & Dubes 1988).

Um agrupamento exclusivo é uma partição de um conjunto de objetos. Cada objeto pertence exclusivamente a um único subconjunto (cluster). O resultado desse agrupamento pode ser dito *hard* (um exemplo pertence ou não pertence a um dado cluster). Um agrupamento não exclusivo pode associar um objeto a vários clusters. Os algoritmos de

Tabela 1: Comparação dos Algoritmos.

Nome	**Tipo de clusters formados**	**Número de atributos**	**Referência**
BIRCH	Esféricos de tamanho uniforme.	Elevado.	(Zhang et al. 1996)
k-means	Esféricos de tamanho uniforme ou grupos bem separados.	Não disponível.	(MacQueen 1967)
DENCLUE	De formas arbitrárias.	Elevado.	(Hinneburg & Keim 1998)
CLICK	Não é pré-determinado.	Elevado.	(Sharan & Shamir 2000)
CAST	Não disponível.	Não disponível.	(Ben-Dor et al. 1999)
SOM	Hiperesféricos (existe uma ordem topológica).	Elevado.	(Kohonen 1997)
GCS	Não disponível.	Elevado.	(Fritzke 1994)
SOTA	Esféricos.	Elevado.	(Dopazo & Carazo 1997)
CLIQUE	De formas arbitrárias.	Elevado.	(Agrawal et al. 1998)
MAFIA	De formas arbitrárias.	Elevado.	(Nagesh et al. 2001a) (Nagesh et al. 2001b)

agrupamento *fuzzy* (cada exemplo tem um grau de pertinência à cada um dos clusters) são uma forma de agrupamento não exclusivo. O foco deste trabalho é nos algoritmos exclusivos.

Esses algoritmos exclusivos podem ser subdivididos em hierárquicos e particionais, de acordo com o tipo de estrutura imposta aos dados. A estrutura resultante de um algoritmo particional é uma única partição dos dados, enquanto que um agrupamento hierárquico resulta em uma seqüência aninhada de partições.

Os algoritmos podem ser divididos ainda de acordo com outros aspectos. Algumas divisões são aglomerativos × divisivos, seriais × simultâneos e teoria dos grafos × algebra matricial.

A seguir são descritos alguns algoritmos, agrupados de acordo com a categoria em que se enquadram. As categorias em que foram divididos os algoritmos são: hierárquicos, particionais baseados em erro quadrático (*square-error*), baseados em densidade, baseados em grafo, baseados em redes neurais (auto-organizáveis) e baseados em *grid*, lembrando que os algoritmos podem se enquadrar em mais de uma das categorias. A maioria dos algoritmos nas categorias *grid*, densidade, grafo e redes neurais (auto-organizáveis) são algoritmos particionais, embora alguns casos sejam algoritmos hierárquicos.

Alguns algoritmos estudados, mas que não serão detalhados neste trabalho e que não se enquadram nessas categorias, são SVC (*Support Vector Clustering*) (Ben-Hur et al. 2001), MSVC (*Multiple sphere Support Vector Clustering*) (Chiang & Hao 2003), SNNC (*Shared Nearest Neighbor Clustering*) (Ertöz et al. 2002), *Biclustering* (Cheng & Church 2000), *Plaid Model* (Lazzeroni & Owen 2002) e CTWC (*Coupled Two-Way Clustering*) (Getz et al. 2003). Esses três últimos algoritmo realizam agrupamento conjunto dos padrões e das características.

5.1 Algoritmos Hierárquicos

Um método de agrupamento hierárquico é um procedimento para transformar uma matriz de proximidade em uma seqüência de partições aninhadas.

Seja uma seqüência de partições de n amostras em K clusters, em que o nível 1 corresponde a n clusters de um elemento e o nível n corresponde a um cluster com todos os elementos. Um agrupamento hierárquico (Duda et al. 2001) agrupa os dados de forma que se dois exemplos são agrupados em algum nível, nos níveis mais altos eles continuam fazendo parte do mesmo grupo, construindo uma hierarquia de clusters. O agrupamento hierárquico pode ser dividido em duas abordagens: a aglomerativa, que começa com n clusters com um único exemplo e forma a seqüência de partições agrupando os clusters sucessivamente e a divisiva, que começa com um cluster com todos os exemplos e forma a seqüência dividindo os clusters sucessivamente.

Neste tipo de agrupamento, as soluções são tipicamente representadas por um dendrograma, que consiste de um tipo especial de estrutura de árvore. Um dendrograma consiste de camadas de nós, cada um representando um cluster. Linhas conectam nós representando clusters aninhados. O corte de um dendrograma na horizontal representa uma partição.

Os aspectos positivos do agrupamento hierárquico são a flexibilidade em relação ao nível de granularidade, a facilidade na utilização de qualquer forma de similaridade ou distância e sua aplicação a qualquer tipo de atributo. Como aspectos negativos têm-se o critério de terminação vago e o fato de que a maioria dos algoritmos não melhora os clusters, uma vez construídos.

A maioria dos algoritmos hierárquicos usa métricas de integração (*linkage metrics*). Porém, existem várias outras implementações de algoritmos de agrupamento hierárquicos que visam melhorias, por exemplo, na manipulação de *outliers*, obtenção de clusters de diferentes formas e tamanhos e escalabilidade.

Os algoritmos hierárquicos mais comuns são os baseados em métricas de integração. Uma descrição geral desse tipo de algoritmo é apresentada a seguir, sem a especificação de um algoritmo particular, pois existe um grande número desse tipo de algoritmo. Outro algoritmo hierárquico de interesse para este trabalho é o BIRCH. Além desses algoritmo, foram investigados os algoritmos hierárquicos CURE (Guha et al. 1998), CHAMELEON (Karypis et al. 1999), OPTICS (Ankerst et al. 1999) e ROCK (Guha et al. 2000).

5.1.1 Algoritmos Hierárquicos Baseados em Métricas de Integração

As abordagens clássicas de agrupamento hierárquico se baseiam nas métricas de integração, que são medidas de proximidade entre subconjuntos de pontos. Esse tipo de agrupamento resulta em clusters de formas convexas próprias e possuem complexidade $O(n^2)$. A idéia de cluster por trás desses algoritmos é a de que um cluster é constituído de pontos similares.

Os algoritmos baseados nas métricas de integração funcionam da seguinte maneira: inicializam um sistema de cluster como um conjunto de clusters de um elemento (aglome-

rativo) ou um único cluster com todos os elementos (divisivo) e iterativamente unem ou dividem o cluster mais apropriado, até que seja atingido um critério de parada. Um cluster ser apropriado para ser unido ou dividido depende da similaridade/dissimilaridade dos elementos do cluster. Para agrupar/dividir subconjuntos de pontos a similaridade é dada pelas métricas de integração. A métrica utilizada afeta significativamente o algoritmo. As principais métricas são *single-link*, *average-link* e *complete-link*. Essas métricas são calculadas com base em uma medida de similaridade d, calculada para cada par de pontos, com um ponto no primeiro conjunto (cluster), C_1 e outro no segundo, C_2, seguida da aplicação de uma operação específica, op, como mínimo (*single-link*), média (*average-link*) e máximo (*complete-link*):

$$d(C_1, C_2) = op\{d(x, y) | x \in C_1, y \in C_2\} \tag{24}$$

A métrica *single-link* é boa para manipular formas não elípticas, mas é bastante sensível a ruídos e *outliers*, já a métrica *complete-link* é menos suscetível a ruídos e *outliers*, mas pode quebrar clusters grandes e tem problemas com formas convexas (Barbara 2000).

Esses algoritmos de agrupamento hierárquico não lidam bem com ruídos e *outliers* e dependem da ordem dos dados. Por outro lado, não requerem a especificação do número de clusters e seus resultados correspondem a taxonomias, muito comuns nas ciências biológicas. O resultado é geralmente apresentado na forma de um dendrograma, que consiste de uma árvore binária que representa a hierarquia dos clusters (Barbara 2000).

Esses algoritmos não tem uma função objetivo global. São baseados em decisões locais. Para as técnicas aglomerativas, o critério de agrupamento é tipicamente agrupar os pares de clusters mais próximos, de acordo com a métrica de integração utilizada (Barbara 2000). Para as técnicas divisivas o critério é, geralmente, dividir os grupos que geram partições mais diferentes.

5.1.2 BIRCH - Balanced Iterative Reducing and Clustering using Hierarchies

A principal idéia do algoritmo BIRCH (Zhang et al. 1996; Barbara 2000; Han et al. 2001) é comprimir os pontos de dados em sub-clusters e depois agrupar esses sub-clusters, na memória principal. Com isso, o algoritmo precisa de uma única varredura na base de dados. Sua principal contribuição é a habilidade de lidar com conjuntos de dados muito grandes.

Uma deficiência desse algoritmo é que ele apresenta um desempenho ruim quando os clusters não têm tamanho e forma uniformes. Além disso, ele é adequado para dados em espaços de vetor Euclideano, ou seja, os dados devem ser métricos (simplificadamente, dados para os quais médias fazem sentido).

Cada sub-cluster é representado por uma *clustering feature* (CF), que é uma tripla resumindo informações a respeito do grupo de pontos que ela representa. $CF = (N, \overrightarrow{LS}, SS)$, em que N é o número de pontos no sub-cluster, LS é a soma linear dos N pontos, dada pela Equação 25 e SS é a soma quadrática dos pontos, dada pela Equação 26. $\overrightarrow{LS}$ e SS são quantidades estatísticas comuns, a partir das quais pode-se derivar

diversas medidas de distância entre clusters.

$$\overrightarrow{LS} = \sum_{i=1}^{N} \overrightarrow{X}_i \tag{25}$$

$$SS = \sum_{i=1}^{N} \overrightarrow{X}_i^2 \tag{26}$$

As CFs são armazenadas em uma árvore CF, utilizada para resumir as representações dos clusters. Uma árvore CF é uma árvore balanceada na altura (*height-balanced*) com dois parâmetros. O fator de ramificação (*branching factor*), B, da árvore, especifica o número máximo de filhos por nó não folha. O *threshold*, T, especifica o diâmetro máximo dos sub-clusters armazenados nos nós folha. Esses parâmetros influenciam o tamanho da árvore resultante. Os nós que não são folhas armazenam as somas das CFs dos seus nós filhos, resumindo suas informações e representando um cluster composto de todos os sub-clusters representados por esses filhos.

A árvore CF é construída dinamicamente. Conforme os pontos de dados são lidos, a árvore é percorrida a partir da raiz, escolhendo o nó mais próximo em cada nível. Se a adição do ponto ao nó folha (cluster) mais próximo não resultar em um cluster com diâmetro maior do que T, o ponto é adicionado ao nó, atualizando a informação da sua CF e de todos os nós até a raiz. Caso contrário, se o nó folha não estiver cheio, uma nova entrada é criada. Se o nó estiver cheio, ele é dividido, propagando o resultado até a raiz. A cada divisão segue um passo de união. A construção da árvore também possui um procedimento para remoção de *outliers*.

O algoritmo BIRCH consiste de 4 fases. A primeira fase consiste em carregar os dados na memória, criando uma árvore CF que resume os dados. Em seguida, pode ser aplicada uma fase em que é construía uma árvore CF menor, se for necessário. A terceira fase consiste da aplicação de um algoritmo de agrupamento global para agrupar os nós folha. Nesta fase, vários algoritmos podem ser utilizados. No algoritmo original, foi adaptado para esta fase um algoritmo hierárquico aglomerativo, aplicado diretamente aos sub-clusters representados pelas CFs. Para isso, as métricas de distância entre dois clusters $D2$ ou $D4$ foram utilizadas. A quarta fase consiste da redistribuição dos pontos de dados utilizando os centróides dos clusters obtidos na terceira fase, gerando um novo conjunto de clusters. Esta fase pode ser utilizada também para rotular os pontos de acordo com o cluster a que eles pertencem.

Um resumo das principais características do algoritmo BIRCH pode ser observado na Tabela 2.

5.2 Algoritmos Particionais Baseados em Erro Quadrático

Esses algoritmos otimizam o critério de agrupamento utilizando uma técnica iterativa. O primeiro passo consiste da criação de uma partição inicial. Em seguida, os padrões são movidos de um cluster para outro com o objetivo de melhorar o valor da função objetivo.

Tabela 2: Resumo das características do algoritmo BIRCH.

Algoritmo	**BIRCH**
Categoria	Hierárquico.
Formas dos clusters	Esféricas de tamanho uniforme.
Tipo dos atributos	Numéricos métricos.
Alta dimensionalidade	Sim.
Escalabilidade	Sim.
Robustez contra ruídos/*outliers*	Possui um mecanismo para lidar com *outliers*, sendo mais eficiente que os algoritmos particionais. Porém, segundo Hinneburg & Keim (1999), é sensível a ruído para dimensões mais altas, para situações realísticas em que os dados são lidos em uma ordem aleatória.
Dependência da ordem dos dados	Sim.
Parâmetros de entrada	Parâmetros da árvore CF: raio dos clusters e fator de ramificação. Para uma discussão dos parâmetros específicos de cada fase ver Zhang et al. (1996).
Medidas de similaridade	Distâncias entre dois clusters: distância Euclideana do centróide, distância Manhattan do centróide, distância média *inter-cluster*, distância média *intra-cluster*, distância *variance increase*.
Critério de agrupamento	Encontrar uma partição do conjunto de dados em K clusters que minimiza a função de distância, levando em consideração a limitação na quantidade de memória disponível e a minimização do tempo de I/O.
Resultados	Uma árvore CF, com cada nó representando um sub-cluster e, dependendo da quarta fase, os pontos rotulados.
Complexidade	$O(n)$.
Referência	(Zhang et al. 1996)
Software	www.cs.wisc.edu/~vganti/birchcode

Esses algoritmos são computacionalmente eficientes, porém podem convergir para um mínimo local.

O critério de agrupamento utilizado por esses algoritmos é o erro quadrático. O objetivo é obter uma partição que minimiza o erro quadrático para um número fixo de clusters. Minimizar o erro quadrático, ou a variação dentro de um cluster é equivalente a maximizar a variação entre clusters.

Dado o centróide de um cluster, x_{0k}, como definido na Equação 16, o erro quadrático para o cluster C_k, dado pela Equação 27, é a soma das distâncias Euclideanas quadradas entre cada padrão no cluster C_k e seu centróide x_{0k}.

$$e_k{}^2 = \sum_{i=1}^{n_k} (x_{ik} - x_{0k})^T (x_{ik} - x_{0k}) \tag{27}$$

O critério para o agrupamento completo contendo K clusters é a soma da variação dentro dos clusters, dada pela Equação 28. A distância de Mahalanobis também pode ser utilizada para definir o erro quadrático.

$$E_K{}^2 = \sum_{k=1}^{K} e_k{}^2 \tag{28}$$

O objetivo desse tipo de agrupamento é encontrar uma partição contendo K cluster que minimiza $E_K{}^2$, para K fixo. A partição resultante também é chamada de partição de variância mínima.

O principal representante dessa categoria é o algoritmo *k-means*. Embora esse algoritmo não atenda a várias das necessidades deste trabalho, ele está incluído na lista dos algoritmos de interesse por ser um dos algoritmos mais simples e também bastante utilizado na literatura sobre agrupamento em análise de expressão gênica. Outros algoritmos brevemente estudados foram PAM (Kaufman & Rousseeuw 1990), CLARA (Kaufman & Rousseeuw 1990) e CLARANS (Ng & Han 1994).

5.2.1 K-means

O algoritmo *K-means* (Duda et al. 2001) é um método que particiona o conjunto de dados em K clusters. Esses clusters são formados com base em alguma medida de similaridade. Esse algoritmo utiliza uma técnica de realocação iterativa, que encontra um ótimo local. Existem várias versões do algoritmo, cada uma solucionando uma deficiência do algoritmo original. Por exemplo, a distância de Mahalanobis pode ser utilizada para encontrar clusters hiperelipsóides. Berkhin (2002) e Jain et al. (1999) discutem brevemente algumas dessas versões.

Basicamente, o algoritmo *k-means* começa inicializando um conjunto de K centróides para os clusters. Cada ponto do conjunto de dados é associado ao cluster com o centróide mais próximo. Em seguida, os centróides são re-calculados. O processo é repetido até que os centróides não sejam mais alterados.

Tabela 3: Resumo das características do algoritmo *k-means*.

Algoritmo	***K-means***
Categoria	Particional.
Formas dos clusters	Hiperesféricas de tamanho similar ou grupos bem separados.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Não disponível.
Escalabilidade	Não. Adequado para conjuntos pequenos e médios.
Robustez contra ruídos/*outliers*	Não.
Dependência da ordem dos dados	Sim, se os centróides forem atualizados incrementalmente.
Parâmetros de entrada	Número de clusters.
Medidas de similaridade	A mais comum é a distância Euclideana.
Critério de agrupamento	Minimizar o erro quadrático.
Resultados	Centros dos clusters.
Complexidade	$O(n)$.
Referência	(MacQueen 1967)
Software	Cluster 3.0 e Expander.

O critério de agrupamento do *k-means* pode ser descrito pela Equação 29, em que x_{0k} é o centróide do cluster C_k e $d(x_i, x_{0k})$ é a distância entre um ponto x_i e x_{0k}. O centróide pode ser a média ou a mediana de um grupo de pontos. Dito de outra maneira, o critério do *k-means* é minimizar a distância entre cada ponto e o centróide do cluster ao qual o ponto pertence (Halkidi et al. 2001). Essa função objetivo é minimizada por clusters de formato globular de tamanho igual ou clusters bem separados.

$$E = \sum_{k=1}^{K} \sum_{x_i \in C_k} d(x_i, x_{0k}) \tag{29}$$

A complexidade do algoritmo é $O(n)$, uma vez que o número de iterações é tipicamente pequeno e $K << n$ (Barbara 2000). Além disso, também é considerado que $d << n$.

O algoritmo é sensível à escolha inicial dos centróides e da sua forma de atualização. Dependendo da escolha dos centróides, o algoritmo pode convergir para um ótimo local. Além disso, é restrito a dados em espaços Euclideanos e os clusters encontrados são desbalanceados.

Um resumo das principais características do algoritmo *k-means* pode ser observado na Tabela 3.

5.3 Algoritmos Baseados em Densidade

Esses algoritmos assumem que os clusters são regiões de alta densidade de padrões separadas por regiões com baixa densidade, no espaço de padrões. Um cluster definido como um componente denso conectado cresce em qualquer direção dada pela densidade

Berkhin (2002). Portanto, os algoritmos baseados em densidade são capazes de obter clusters de formas arbitrárias.

Além do algoritmo DENCLUE descrito a seguir, e dos outros algoritmos também baseados em densidade, mas enquadrados em outras categorias, foram investigados os algoritmos DBSCAN (Ester et al. 1996) e Wave-cluster (também baseado em *grid*) (Sheikholeslami et al. 1998).

5.3.1 *DENCLUE - DENsity-based CLUstEring*

O algoritmo DENCLUE (Hinneburg & Keim 1998) modela a densidade global de um conjunto de pontos como a soma de funções "influência" associadas a cada cluster. A função de densidade global resultante tem picos locais que podem ser utilizados para definir clusters. Para cada ponto encontra-se o pico mais próximo associado a ele. O conjunto de todos os pontos associados a um pico particular (atrator de densidade local) se torna um cluster. Entretanto, se a densidade em um pico local é muito baixa, os pontos associados a esse cluster são considerados ruídos e descartados. Se dois picos locais são conectados por um caminho de pontos e a densidade de cada um desses pontos no caminho está acima de um *threshold* de densidade mínimo ξ, então os clusters associados a esses picos são unidos.

A função de densidade global exige a soma das funções influência de todos os pontos. Porém, a maioria dos pontos não contribui para a função de densidade global. Por isso, o algoritmo DENCLUE utiliza uma função de densidade local que considera apenas os pontos que de fato contribuem para a função de densidade global

O DENCLUE é baseado em estimação de densidade por *kernel* (*kernel density estimation*), que tem como objetivo descrever a distribuição dos dados por uma função. A contribuição de cada ponto para a função de densidade global é expressa por uma função influência ou *kernel*. A função global é a soma das funções associadas a cada ponto.

Para a definição da função influência é utilizada uma função de distância fd qualquer, que seja reflexiva e simétrica. Hinneburg & Keim (1998) utilizam a distância Euclideana. Tipicamente, a função influência é simétrica e decresce com o aumento da distância ao ponto. Uma função *kernel* utilizada frequentemente é a função Gaussiana $G(x) = \exp^{\frac{-fd(x,y)^2}{2\sigma^2}}$, em que σ é um parâmetro que governa o quão rapidamente a influência do ponto diminui.

Este algoritmo possui dois passos: pré-agrupamento e agrupamento. Na etapa de pré-agrupamento é construído um mapa da porção relevante do espaço de dados para acelerar o cálculo da função de densidade. Na etapa de agrupamento o algoritmo identifica os atratores de densidade e os pontos atraídos correspondentes.

O mapa é criado dividindo o (hiper-)retângulo de limite mínimo (*minimum bounding hyper-rectangle*) dos dados em hipercubos de dimensão d com aresta de tamanho 2σ. Os hipercubos que contêm pontos de dados são determinados. Em seguida, eles são numerados de acordo com sua posição em relação a uma origem particular. Dessa forma, os hipercubos são mapeados em chaves unidimensionais. As chaves dos cubos povoados são armazenadas em uma árvore de busca para permitir acesso eficiente posteriormente.

Tabela 4: Resumo das características do algoritmo DENCLUE.

Algoritmo	**DENCLUE**
Categoria	Baseado em densidade/*grid*.
Formas dos clusters	Arbitrárias.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Sim.
Escalabilidade	Sim.
Robustez contra ruídos/*outliers*	Sim.
Dependência da ordem dos dados	Não.
Parâmetros de entrada	Raio do cluster. Número mínimo de objetos.
Medidas de similaridade	Qualquer distância reflexiva e simétrica.
Critério de agrupamento	Cada cluster é formado pelos pontos atraídos por um atrator de densidade (máximo local de uma função de densidade global que é a soma das funções de "influência" de cada ponto).
Resultados	Atribuição de valores de dados aos clusters.
Complexidade	$O(n \log n)$.
Referência	(Hinneburg & Keim 1998)
Software	Não disponível.
Observações	Permite uma descrição matemática compacta dos clusters de formas arbitrárias em conjuntos de alta dimensionalidade. É muito sensível aos parâmetros, que são difíceis de determinar. Tem uma fundamentação matemática sólida. Dependendo dos parâmetros pode se comportar como DBSCAN, *k-means* ou hierárquico.

Para o agrupamento são considerados apenas os cubos mais povoados e os cubos conectados a eles. Para cada ponto x é calculada a função de densidade local considerando apenas os pontos de clusters conectados ao cluster que contém x e têm centróides a uma distância de $k\sigma$ de x. Cada ponto x' no caminho de x ao seu atrator de densidade é associado ao mesmo cluster de x se a distância entre x e x' for menor ou igual a $\sigma/2$. Os clusters associados com um atrator de densidade cuja densidade seja menor que ξ é descartado. Os atratores de densidade ligados por um caminho de pontos de densidade maior que ξ são unidos.

Este algoritmo tem uma fundamentação matemática sólida e apresenta uma descrição matemática compacta dos clusters. Ele pode também ser classificado como baseado em *grid*. Uma deficiência do DENCLUE é que ele é muito sensível aos parâmetros, que são difíceis de determinar. Dependendo da escolha dos parâmetros ele pode se comportar como os algoritmos DBSCAN, *k-means* ou hierárquico. Um resumo das principais características do algoritmo DENCLUE pode ser observado na Tabela 4.

5.4 Algoritmos Baseados em Grafo

Para a realização de um agrupamento baseado em teoria dos grafos, os dados são representados em um grafo de proximidade. No caso mais simples, cada nó é conectado com os $n-1$ nós restantes, resultando em um grafo completo.

Várias estruturas de grafo podem ser utilizadas para representar um subconjunto de arestas para refletir a estrutura dos clusters. Os métodos de agrupamento decompõem os grafos em componentes conectados pela remoção e inserção de arestas inconsistentes. Cada um desses componentes representa um cluster.

5.4.1 HSC - Highly Connected Subgraph e CLICK - Cluster Identification via Connectivity Kernels

Os algoritmos HSC (Hartuv & Shamir 2000) e CLICK (Sharan & Shamir 2000) utilizam uma abordagem da teoria dos grafos para agrupar os dados. Os dados de entrada são representados como um grafo de similaridade. O algoritmo particiona recursivamente o conjunto atual de elementos em dois subconjuntos. Antes de uma divisão, o algoritmo considera o subgrafo induzido pelo subconjunto atual de elementos. Se o subgrafo satisfaz um critério de parada, então ele é declarado um *kernel*. De outra forma, um corte de peso mínimo é computado naquele subgrafo, e o conjunto é dividido nos dois subconjuntos separados por aquele corte. A saída é uma lista de *kernels* que serve como base para os eventuais clusters. A diferença entre os dois algoritmos, HSC e CLICK, está no grafo de similaridade que eles constroem, no critério de parada e no pós-processamento dos *kernels* (Shamir & Sharan 2002).

O algoritmo CLICK é mais novo e mais utilizado atualmente. Assim, outros detalhes, resumidos na Tabela 5, são apresentados somente para o algoritmo CLICK.

5.4.2 CAST - Clustering Affinity Search Technique

Ben-Dor et al. (1999) apresentam um algoritmo teórico e um algoritmo heurístico CAST, baseado nas mesmas idéias do algoritmo teórico.

CAST (*Clustering Affinity Search Technique*) (Ben-Dor et al. 1999) é um algoritmo polinomial utilizado para encontrar agrupamentos verdadeiros com alta probabilidade, sob o seguinte modelo estocástico dos dados: A estrutura base de cluster correta é representada por um grafo que é uma união disjunta de grupos, e erros são subseqüentemente introduzidos no grafo, removendo e adicionando arestas entre pares de vértices com probabilidade α. Se todos os clusters são de tamanho pelo menos cn, para alguma constante $c > 0$ e n = número de padrões a serem agrupados, o algoritmo resolve o problema com a precisão desejada com alta probabilidade.

CAST emprega uma matriz de similaridade S, cuja entrada S_{ij} representa a similaridade dos padrões i e j de acordo com alguma medida de similaridade ($S_{ij} \in [0,1]$). Essa matriz é utilizada no cálculo da similaridade média (afinidade) entre vértices ainda não associados a clusters e o centro do cluster atual. A afinidade, utilizada para decidir se um elemento v fará parte de um cluster C ou não, é dada por $a(v) = \sum_{i \in C} S_{iv}$. O algoritmo utiliza o parâmetro t, limiar de afinidade (*affinity threshold*) para determinar qual nível

Tabela 5: Resumo das características do algoritmo CLICK.

Algoritmo	**CLICK**
Categoria	Baseado em grafo.
Formas dos clusters	Não é pré-determinada.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Sim.
Escalabilidade	Sim.
Robustez contra ruídos/*outliers*	Sim. Pode criar clusters de um único elemento.
Dependência da ordem dos dados	Não.
Parâmetros de entrada	Grafo de similaridade.
Medidas de similaridade	Produto interno.
Critério de agrupamento	Corte de peso mínimo em grafo.
Resultados	Não disponível.
Complexidade	$O(n)$.
Referência	(Sharan & Shamir 2000)
Software	EXPANDER.
Observações	Não se baseia em suposições sobre o número e a estrutura dos clusters (o algoritmo determina esse número).

de afinidade é considerado significante, influenciando o número e tamanho dos clusters produzidos. Os clusters são gerados um a um. O próximo cluster inicia com um único elemento. Elementos são adicionados ou removidos do cluster se sua afinidade é maior ou menor do que t (alta ou baixa afinidade ao cluster), respectivamente, até que o processo se estabilize e este cluster esteja pronto.

A Tabela 6 contém um resumo das características do algoritmo CAST.

5.5 *Algoritmos Baseados em Redes Neurais*

As redes neurais artificiais representam uma forma de computação não algorítmica, cujo funcionamento é inspirado na estrutura e funcionamento do cérebro humano. Tais redes podem ser definidas como sistemas paralelos distribuídos compostos de unidades de processamento simples, altamente interconectadas, que computam determinadas funções matemáticas. Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede (Braga et al. 2000).

Além dos algoritmos SOM, GCS e SOTA, foram encontrados na literatura os algoritmos HGSOT (Luo, Tang, & Khan 2003) e DGSOT (Luo, Khan, Bastani, Yen, & Zhou 2004).

Tabela 6: Resumo das características do algoritmo CAST.

Algoritmo	**CAST**
Categoria	Baseado em Grafo.
Formas dos clusters	Não disponível.
Tipo dos atributos	Vários.
Alta dimensionalidade	Não disponível.
Escalabilidade	Não disponível.
Robustez contra ruídos/*outliers*	Sim.
Dependência da ordem dos dados	Não disponível.
Parâmetros de entrada	Matriz de similaridade. Limiar de afinidade.
Medidas de similaridade	Qualquer medida em que $S_{ij} \in [0, 1]$.
Critério de agrupamento	Os clusters são construídos pela adição de padrões com alta afinidade pelo cluster e remoção de padrões com baixa afinidade.
Resultados	Grafo clique.
Complexidade	Não é determinada.
Referência	(Ben-Dor et al. 1999)
Software	Não disponível.
Observações	O algoritmo determina o número de clusters, não precisando de conhecimento prévio da estrutura de clusters. Não tem prova da complexidade de tempo e da convergência da heurística.

5.5.1 SOM - Self Organizing Map

SOM (*Self Organizing Map*) (Haykin 1999) é uma rede neural artificial não supervisionada, freqüentemente utilizada em tarefas de agrupamento e visualização. Nesse tipo de rede, os neurônios são organizados em um reticulado uni ou bidimensional. Cada neurônio no reticulado está conectado a todas as entradas da rede. Esta rede geralmente utiliza uma única camada computacional. A cada padrão de entrada apresentado à rede, os neurônios computam seus valores de ativação, ativando uma região diferente do reticulado. Para cada padrão de entrada, os neurônios de saída da rede competem entre si para serem ativados. O neurônio com maior valor de ativação é o vencedor da competição. Em seguida, é determinada a localização espacial de uma vizinhança topológica de neurônios excitados, centrada no neurônio vencedor. O próximo passo consiste de uma adaptação dos pesos. Os ajustes dos pesos são tais que a resposta do neurônio vencedor à aplicação subseqüente de um padrão de entrada similar é melhorada. Assim, durante a execução do algoritmo, os vetores de entrada direcionam o movimento dos vetores de peso, promovendo uma organização topológica dos neurônios da rede. Ainda durante o treinamento, a região de vizinhança dos neurônios é gradativamente reduzida.

O objetivo da rede SOM é encontrar um conjunto de vetores de referência e associar cada ponto do conjunto de dados ao vetor referência mais próximo. O algoritmo depende da inicialização dos vetores de referência. O resultado consiste de um conjunto de vetores de referência que definem implicitamente os clusters. Uma deficiência da rede SOM é que ela não detecta automaticamente a borda dos clusters.

Há um equivoco comum na comunidade de bioinformática de que cada unidade do mapa gerado pela rede SOM deve ser considerado como um cluster separado. Na realidade, várias unidades vizinhas podem modelar um único cluster.

As principais características do algoritmo SOM podem ser observadas na Tabela 7.

5.5.2 GCS - Growing Cell Structures

A rede neural auto-organizável G CS (*Growing Cell Structures*) (Fritzke 1994) é uma extensão da rede SOM. Esse modelo tem uma estrutura flexível e compacta, um número variável de elementos e é capaz de detectar clusters de padrões similares. Tais padrões são vetores de números reais de dimensão d que seguem uma distribuição de probabilidade desconhecida $P(x)$. A idéia da rede GCS é construir uma estrutura bi-dimensional de células com vetores de peso associados que modelem a distribuição $P(x)$. A remoção de células em regiões com baixa densidade de probabilidade faz com que a estrutura seja dividida em várias sub-estruturas desconectadas, cada uma identificando um cluster. A rede, desta forma, consegue determinar o número de clusters e também a distribuição de probabilidade dentro de cada cluster.

A topologia inicial da rede A é um simplex de dimensão k (se $k = 1$ a estrutura é uma linha, se $k = 2$ a estrutura é um triângulo). Os $k + 1$ vértices correspondem às células (neurônios). As $(k + 1)k/2$ arestas (conexões) denotam as relações de vizinhança topológica. Cada célula c tem um vetor de pesos associado w_c, com a mesma dimensão d dos dados de entrada. Esse vetor de pesos pode ser visto como a posição de c no espaço

Tabela 7: Resumo das características do algoritmo SOM.

Algoritmo	**SOM**
Categoria	Redes neurais.
Formas dos clusters	Hiperesféricas. Existe uma ordem topológica.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Sim (Nikkilä et al. 2002).
Escalabilidade	Sim (Tamayo et al. 1999).
Robustez contra ruídos/*outliers*	Sim (Herrero et al. 2001; Hautaniemi et al. 2003).
Dependência da ordem dos dados	Não disponível.
Parâmetros de entrada	Taxa de aprendizado, Topologia/número dos neurônios, Função de vizinhança.
Medidas de similaridade	Distância Euclideana, produto interno e, para dados de expressão gênica, correlação.
Critério de agrupamento	Associa pesos aos neurônios do mapa topológico de acordo com a similaridade dos padrões associados a cada neurônio, formando regiões de neurônios próximos, cada uma representando um cluster.
Resultados	Conjunto de vetores de referência que definem implicitamente os clusters (Barbara 2000).
Complexidade	$O(n)$.
Referência	(Kohonen 1997)
Software	Cluster 3.0, Expander e GeneCluster.

de entrada.

O aprendizado nessa rede é realizado através de adaptações nos vetores de peso e modificações na topologia da rede. A cada número fixo (λ) de passos de adaptação insere-se uma célula na estrutura da rede. Além disso, periodicamente, remove-se células pertencentes a regiões de baixa densidade de probabilidade. Todas essas inserções e remoções devem ser feitas de forma a manter a estrutura de simplex em cada região formada. A Figura 3 apresenta um exemplo da estrutura inicial de uma rede GCS. Um exemplo da estrutura final, após um certo número de ciclos de treinamento pode ser observado na Figura 4.

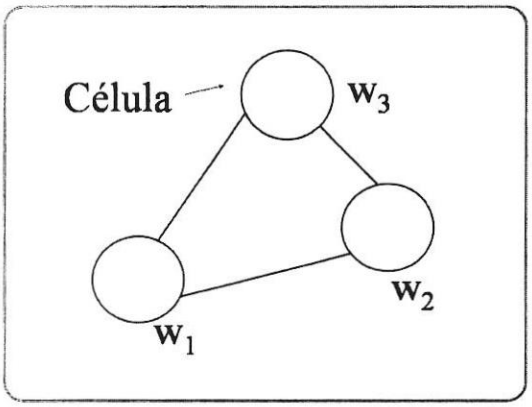

Figura 3: Exemplo de topologia inicial de uma rede GCS (Fritzke 1994).

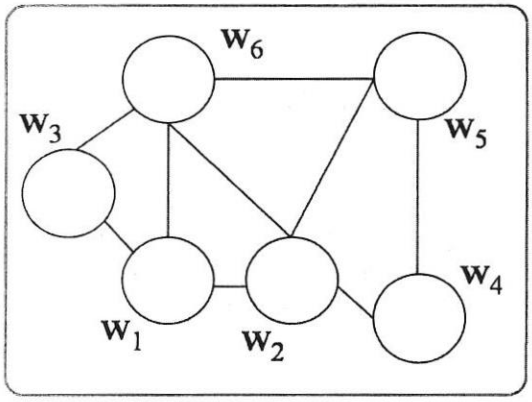

Figura 4: Exemplo da topologia de uma rede GCS depois de treinada (Fritzke 1994).

A adaptação dos pesos dos neurônios se dá da seguinte maneira. Para cada entrada x, determina-se a célula mais próxima, *bmu* (*best-matching unit*) satisfazendo $\|w_{bmu} - x\| \leq \|w_c - x\|$ para todo $c \in A$. Move-se a célula *bmu* e seus vizinhos topológicos diretos (N_{bmu}) em direção ao vetor de entrada, com base nas frações da distância total ε_{bmu} e $\varepsilon_{neighbor}$, de acordo com $\Delta w_{bmu} = \varepsilon_{bmu}(x - w_{bmu})$ e $\Delta w_c = \varepsilon_{neighbor}(x - w_c)$ para todo $c \in N_{bmu}$. Em seguida, incrementa-se um contador de sinal τ_{bmu}, que indica o número de vezes que essa célula já foi *bmu* e decrementa-se o contador de sinal de todas as células por $\Delta\tau_c = -\alpha\tau_c$ para todo $c \in A$.

Tabela 8: Resumo das características do algoritmo GCS.

Algoritmo	**GCS**
Categoria	Redes neurais.
Formas dos clusters	Não disponível.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Sim.
Escalabilidade	Não disponível.
Robustez contra ruídos/*outliers*	Não disponível.
Dependência da ordem dos dados	Não disponível.
Parâmetros de entrada	Não disponível.
Medidas de similaridade	Não disponível.
Critério de agrupamento	Os clusters são formados por adaptações nos vetores de peso e modificações na topologia da rede, formando estruturas desconectadas de acordo com a distribuição de probabilidade dos padrões (desconhecida).
Resultados	Uma estrutura simplex bi-dimensional para cada cluster.
Complexidade	Não disponível.
Referência	(Fritzke 1994)
Software	Não disponível.
Observações	O modelo encontra automaticamente uma estrutura e tamanho adequados para a rede (detecta automaticamente a borda dos clusters).

Para a inserção de células, considera-se que os vetores de peso distribuídos de acordo com $P(x)$ são obtidos quando cada célula tem a mesma probabilidade de ser *bmu* para o vetor de entrada atual. Assim, $P(x)$ é estimado pela freqüência relativa dos sinais de entrada recebidos por uma célula, $h_c = \tau_c / \sum_{j \in A} \tau_j$. Depois de λ passos de adaptação, determina-se a célula q com $h_q \geq h_c$ para todo $c \in A$. Em seguida, procura-se um vizinho direto de c, f (vizinho de maior distância), satisfazendo $\|w_f - w_q\| \geq \|w_c - w_q\|$ para todo $c \in A$ e insere-se uma célula r entre q e f, conectando-a às outras células de forma a manter a estrutura e fazendo $w_r = 0.5(w_q + w_f)$. Posteriormente, redistribui-se os contadores de sinal.

Para a remoção de células, leva-se em consideração que clusters de vetores similares são separados por regiões com baixa densidade de probabilidade. Para encontrar tais regiões, estima-se localmente a densidade de probabilidade próxima a w_c por $p_c = h_c / |F_c|$. A remoção de células é feita periodicamente, eliminando-se células com valores de p_c abaixo de um *threshold*, mantendo a estrutura de simplex.

Depois do aprendizado, cada padrão é categorizado em uma das células existentes. O resultado é uma rede interconectada de células que agrupam vários padrões de acordo com sua similaridade.

A Tabela 8 resume as principais características do algoritmo GCS.

5.5.3 SOTA - Self-Organizing Tree Algorithm

Self-Organizing Tree Algorithm (SOTA) (Herrero et al. 2001) é uma rede neural que cresce adotando a topologia de uma árvore binária. SOTA é um algoritmo hierárquico divisivo baseado nas redes neurais *Self Organizing Maps* (SOM) e *Growing Cell Structures* (GCS).

O sistema inicial é composto de 2 elementos externos, chamados células, conectados por um elemento interno chamado nó. Cada célula ou nó é um vetor com a mesma dimensão dos dados de entrada. Inicialmente, as entradas das células e nó são inicializadas com o valor médio das colunas correspondentes do conjunto de dados ou com valores aleatórios.

Nesta rede, apenas as células são comparadas com os padrões. O algoritmo funciona expandindo a topologia iniciando pela célula que tem a população de padrões associada mais heterogênea. A partir desta célula heterogênea, dois novos descendentes são gerados e a célula muda seu estado de célula para nó. Esse processo é chamado de ciclo. Durante um ciclo, células e nós são repetidamente adaptados de acordo com os padrões de entrada.

O processo de sucessivos ciclos de geração de células descendentes é repetido até que cada célula tenha um único padrão associado (ou vários padrões idênticos) ou até que um nível de heterogeneidade desejado nas células seja atingido.

O agrupamento obtido é proporcional à heterogeneidade dos dados, ao invés do número de itens. Assim, se um dado tipo de padrão é abundante, todos os itens similares permanecerão agrupados em um mesmo cluster, sem afetar as próximas etapas do processo de treinamento. Essa característica se deve ao fato de que o algoritmo SOTA preserva a distribuição, enquanto o SOM preserva a topologia.

A Tabela 9 resume as principais características do algoritmo SOTA.

5.6 Algoritmos Baseados em Grid

Este grupo de algoritmos define um *grid* para o espaço de dados e realiza todas as operações nesse espaço quantizado. Em termos gerais, essa abordagem é muito eficiente para conjuntos de dados grandes, é capaz de encontrar clusters de formas arbitrárias e lidam bem com *outliers*. Os dois algoritmos descritos nesta seção (CLIQUE e MAFIA) são técnicas projetadas especificamente para trabalhar com dados de alta dimensão.

Além dos algoritmos CLIQUE e MAFIA descritos foram investigados os algoritmos baseados em *grid* OptiGrid (Hinneburg & Keim 1999) e STING (Wang et al. 1997).

5.6.1 CLIQUE - Clustering In QUEst

O algoritmo CLIQUE (Agrawal et al. 1998) encontra clusters em sub-espaços dos dados. Este algoritmo é baseado em *grid* e densidade. Ele identifica clusters densos em sub-espaços de dimensionalidade máxima. Os clusters gerados são descritos na forma de expressões DNF (*Disjunctive Normal Form*) que são minimizadas para facilitar a compreensão. Os resultados do agrupamento não dependem da ordem de apresentação dos padrões. O algoritmo CLIQUE também não supõe nenhuma forma matemática específica

Tabela 9: Resumo das características do algoritmo SOTA.

Algoritmo	**SOTA**
Categoria	Redes neurais/hierárquico divisivo.
Formas dos clusters	Herrero et al. (2001) não têm uma idéia clara sobre a forma real dos clusters, mas devem ser de algum modo esféricos.
Tipo dos atributos	Qualquer dado que possa ser codificado como uma série de números e em que possa ser usada uma medida de similaridade computável entre os dados.
Alta dimensionalidade	Sim.
Escalabilidade	Sim.
Robustez contra ruídos/*outliers*	Sim.
Dependência da ordem dos dados	Não.
Parâmetros de entrada	*Threshold* de heterogeneidade.
Medidas de similaridade	Distância Euclideana e coeficiente de correlação de Pearson.
Critério de agrupamento	A hierarquia de clusters gerada minimiza a probabilidade de ter padrões associados de forma incorreta.
Resultados	Hierarquia de células com padrões semelhantes associados.
Complexidade	Aproximadamente O(n).
Referência	(Dopazo & Carazo 1997)
Software	Sotarray (roda on-line na web).
Observações	Baseado em SOM e GCS.

de distribuição dos dados. Além disso, o algoritmo é tolerante a valores ausentes nos dados de entrada.

Como algoritmo baseado em densidade, CLIQUE usa a definição de um cluster como uma região com densidade maior de pontos do que a região a sua volta. O problema tratado pelo algoritmo é identificar automaticamente projeções dos dados de entrada em um subconjunto dos atributos com essas projeções incluindo regiões de alta densidade.

Este algoritmo funciona encontrando regiões de alta densidade por meio do particionamento do espaço de dados em células (hiper-retângulos) e localizando as células densas. Um cluster corresponde à união de todas as células de alta densidade adjacentes. CLIQUE é baseado na seguinte propriedade dos clusters: uma vez que um cluster representa uma região densa em algum sub-espaço do espaço de características, haverá áreas densas correspondentes ao cluster em todos os sub-espaços de menor dimensão. Com base nisso, o algoritmo inicia encontrando todas as áreas densas em espaços unidimensionais correspondentes a cada atributo. Em seguida, o algoritmo gera um conjunto de células bi-dimensionais que podem ser densas. Isso é realizado a partir das células unidimensionais densas. Cada célula bi-dimensional deve ser associada com um par de células unidimensionais densas. Da mesma forma são construídas células densas para as demais dimensões. Os clusters são obtidos encontrando um conjunto maximal de unidades densas em k dimensões.

Efetivamente, o Algoritmo CLIQUE resulta em seleção de atributos (seleciona vários sub-espaços) e produz uma visão dos dados de diferentes perspectivas.

A Tabela 10 resume as principais características do algoritmo CLIQUE.

5.6.2 *MAFIA - Merging of Adaptative Finite Intervals*

O algoritmo MAFIA (Nagesh et al. 2001a; Nagesh et al. 2001b) é uma modificação do algoritmo CLIQUE que envolve a utilização de um *grid* adaptativo. Inicialmente, cada dimensão é particionada em um número fixo de células. Em seguida, é gerado um histograma mostrando o número de pontos em cada célula. Grupos de cinco células adjacentes são agrupadas em janelas. A cada janela é associado o valor máximo do número de pontos nas suas cinco células. Duas janelas adjacentes são agrupadas se seus valores são próximos. Se todas as janelas forem combinadas em um única janela, a dimensão é particionada em um número fixo de células e o *threshold* de densidade é aumentado para essa dimensão. Uma distribuição relativamente uniforme dos dados em uma dimensão em particular normalmente indica que não existe um agrupamento nessa dimensão.

A Tabela 11 resume as principais características do algoritmo MAFIA.

6 Validação

A avaliação do resultado de um agrupamento deve ser objetiva com o propósito de determinar se os clusters são significativos, ou seja, se a solução é representativa para o conjunto de dados analisado. Uma estrutura de agrupamento é válida se não ocorreu por acaso, ou se é "rara" em algum sentido, já que qualquer algoritmo de agrupamento encon-

Tabela 10: Resumo das características do algoritmo CLIQUE.

Algoritmo	**CLIQUE**
Categoria	Baseado em grid/densidade.
Formas dos clusters	Arbitrárias.
Tipo dos atributos	Vários.
Alta dimensionalidade	Sim. Acha clusters em sub-espaços dos dados, mas perde a efetividade com o aumento do número de dimensões.
Escalabilidade	Sim.
Robustez contra ruídos/*outliers*	Sim.
Dependência da ordem dos dados	Não.
Parâmetros de entrada	*Threshold* de densidade e Número de intervalos em que uma dimensão é particionada.
Medidas de similaridade	Não disponível.
Critério de agrupamento	Encontrar projeções dos dados de entrada em um subconjunto dos atributos, que contenham regiões de alta densidade.
Resultados	Gera um resumo compacto de cada cluster como uma expressão DNF.
Complexidade	Não disponível.
Referência	(Agrawal et al. 1998)
Software	Não disponível. Os direitos do código fonte pertencem à IBM.
Observações	Faz uso de um algoritmo de tempo exponencial para analisar a estrutura de sub-espaço. Os parâmetros parecem obscuros, difíceis de escolher e potencialmente limitantes.

Tabela 11: Resumo das características do algoritmo MAFIA.

Algoritmo	**MAFIA**
Categoria	Baseado em grid/densidade.
Formas dos clusters	Arbitrárias.
Tipo dos atributos	Numéricos.
Alta dimensionalidade	Escala linearmente com a dimensionalidade.
Escalabilidade	Escala linearmente com o tamanho do conjunto de dados.
Robustez contra ruídos/*outliers*	Sim.
Dependência da ordem dos dados	Não.
Parâmetros de entrada	*Cluster dominance factor*. Especifica a força dos clusters.
Medidas de similaridade	Não disponível.
Critério de agrupamento	Encontrar regiões de alta densidade em sub-espaços do conjunto de dados.
Resultados	Expressão DNF de tamanho mínimo.
Complexidade	$O(c^k + \frac{n}{B}m\gamma)$, em que m: maior dimensionalidade de qualquer unidade densa no conjunto de dados; B: número de registros que cabem no buffer de memória; γ: tempo de acesso de I/O para um bloco de B registros do disco; c: constante
Referência	(Nagesh et al. 2001a) (Nagesh et al. 2001b)
Software	Não disponível.
Observações	Detecta clusters em subespaços.

trará clusters, independentemente se existe ou não similaridade nos dados. Entretanto, se essa similaridade existe, alguns algoritmos podem encontrar clusters mais adequados que outros. Algumas técnicas para a validação dos resultados de técnicas de agrupamento têm sido discutidas na literatura, tais como (He 1999): testes de significância nas variáveis utilizadas para criar os clusters, replicação, testes de significância nas variáveis externas e procedimentos de Monte Carlo. Um estudo sobre processos de validação de agrupamentos pode ser encontrado em (Jain & Dubes 1988; Gordon 1999; Halkidi et al. 2001).

A validação do resultado de um agrupamento, em geral, é feita com base em índices estatísticos, que julgam, de uma maneira qualitativa, o mérito das estruturas encontradas. Um índice quantifica alguma informação a respeito da qualidade de um agrupamento. A maneira pela qual um índice é aplicado para validar um agrupamento é dada pelo critério de validação. Assim, um critério de validação expressa a estratégia utilizada para validar uma estrutura de agrupamento, enquanto que um índice é uma estatística pela qual a validade é testada. Existem três tipos de critérios para investigar a validade de um agrupamento:

Critérios internos: Medem a qualidade de um agrupamento com base apenas nos dados originais (matriz de padrões ou matriz de similaridade). Por exemplo, um critério interno pode medir o grau em que uma partição obtida por um algoritmo de agrupamento é justificado pela matriz de similaridade.

Critérios externos: Avaliam um agrupamento de acordo com uma estrutura pré-especificada, imposta ao conjunto de dados e que reflete a intuição do pesquisador sobre a estrutura presente nos dados. Essa estrutura pré-especificada pode ser uma partição que se sabe previamente existir nos dados, ou um agrupamento construído por um especialista da área com base em conhecimento prévio. Por exemplo, um critério externo pode medir o grau de correspondência entre o número clusters obtidos com o agrupamento e os rótulos dos dados conhecidos previamente.

Critérios relativos: Comparam diversos agrupamentos para decidir qual deles é o melhor em algum aspecto (qual é mais o estável ou qual é o mais adequado aos dados, por exemplo). Podem ser utilizados para comparar diversos algoritmos de agrupamento ou para determinar o valor mais apropriado de algum parâmetro do algoritmo aplicado, como o número de clusters. Por exemplo, pode-se medir quantitativamente qual dentre dois algoritmos melhor se ajusta aos dados ou determinar o número de clusters mais apropriado para um agrupamento feito com um determinado algoritmo.

Três tipos de estrutura podem ser avaliados: hierarquias, partições e clusters individuais. Os três critérios de validação podem ser empregados na validação de qualquer um dos tipos de estrutura possíveis.

Os critérios externos e internos são baseados em testes estatísticos e têm um alto custo computacional (Halkidi et al. 2001). Seu objetivo é medir o quanto o resultado obtido confirma uma hipótese pré-especificada. Neste caso, são utilizados testes de hipótese para determinar se uma estrutura obtida é apropriada para os dados. Isto é feito testando

se o valor do índice utilizado é extraordinariamente grande ou pequeno. Isto requer o estabelecimento de uma população base ou de referência. O mesmo índice pode ser utilizado em um critério externo e interno, embora as distribuições de referência do índice sejam diferentes (Jain & Dubes 1988).

A proposição de um índice para validação é fácil, porém é muito difícil estabelecer *thresholds* com base nos quais se possa afirmar que o valor do índice é grande ou pequeno o suficiente para se considerar o agrupamento "raro", ou válido.

Os índices de validação, ou estatísticas, são funções dos dados que contêm informações úteis, como o erro quadrático de um agrupamento ou a compactação de seus clusters. Um índice é uma variável aleatória. Sua distribuição descreve a frequência relativa com a qual seus valores são gerados sob alguma hipótese. Uma hipótese é uma afirmação sobre a frequência relativa de eventos no espaço amostral que expressa um conceito. Esse conceito pode ser a ausência de estrutura nos dados.

No caso de validação de agrupamentos, a hipótese nula, H_0, é uma afirmação de não estrutura ou aleatoriedade dos dados. Jain & Dubes (1988) e Gordon (1999) descrevem algumas hipóteses nulas comumente utilizadas para a validação de agrupamentos. Além disso, Jain & Dubes (1988) discutem aplicações de cada uma dessas hipóteses.

Conforme dito anteriormente, nesses critérios é utilizado um teste de hipótese que depende da distribuição do índice sob uma hipótese nula, H_0. A diferença entre esses critérios está nas informações utilizadas. Nos critérios externos pode-se observar a utilização de uma partição dos dados conhecida previamente, chamada aqui "partição real", no cálculo do índice. Já nos critérios internos, o cálculo do índice depende somente dos próprios dados, seja na forma de matriz de padrões (conjunto de dados original) ou de matriz de similaridade.

Um grande problema em validação externa ou interna de agrupamentos é o estabelecimento da distribuição dos índices (estatísticas) sob a hipótese nula e consequentemente a determinação dos *thresholds* que dizem se uma partição é adequada de acordo o índice. Os testes reais de validação são geralmente definidos utilizando ferramentas estatísticas, como análise de Monte Carlo e *bootstrapping*.

Os critérios relativos de validação têm como objetivo encontrar o melhor agrupamento que um algoritmo pode obter sob certas suposições e valores para seus parâmetros ou o algoritmo mais apropriado para os dados/estruturas analisados. Existem vários índices de validação relacionados aos critérios relativos. Os próprios índices comumente empregados em critérios internos podem ser utilizados em critérios relativos. O que os distingue é a maneira como o índice é aplicado. Na utilização de um índice em critério relativo, seu valor é calculado para vários agrupamentos que estão sendo comparados, obtendo-se uma sequência de valores. O melhor agrupamento é determinado pelo valor que se destaca nessa sequência, como o valor máximo ou mínimo. Nesse critério, vários algoritmos, ou um mesmo algoritmo com diferentes valores para seus parâmetros, são aplicados ao mesmo conjunto de dados. O índice é calculado para cada uma das partições obtidas. Esses valores do índice são comparados, em geral com o auxílio de um gráfico, para se determinar o melhor algoritmo, ou o melhor valor para um ou mais parâmetros de um algoritmo.

Jain & Dubes (1988) descrevem como os vários tipos de critérios podem ser empregados para validar cada tipo de estrutura (hierarquias, partições e clusters individuais). Já Halkidi et al. (2001) enfocam as análises mais comuns feitas com cada tipo de critério, principalmente utilizando técnicas de Monte Carlo.

Em seguida, são descritos alguns exemplos de índices utilizados em validação. Para essas descrições será utilizada a seguinte notação: P_e é uma partição encontrada com a aplicação de um algoritmo de agrupamento. C_i é o i-ésimo cluster de uma partição. A estrutura (partição) real, ou uma estrutura construída com base em uma intuição sobre a estrutura real dos dados é denotada por P_r.

A avaliação de uma hierarquia pode ser feita utilizando o índice interno coeficiente de correlação *Cophenetic* (CPCC). O diagrama de uma hierarquia produzida por um algoritmo hierárquico pode ser representado por uma matriz "cophenetic" SC. Cada elemento da matriz, SC_{ij}, representa o nível no dendrograma em que os padrões i e j foram encontrados no mesmo cluster pela primeira vez. O índice $CPCC$ mede o grau de similaridade entre SC e a matriz de proximidade dos padrões. Este índice é calculado pela Equação 30, em que $\mu_S = (1/M)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} S_{ij}$ e $\mu_{SC} = (1/M)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} SC_{ij}$. O valor de $CPCC$ está no intervalo $[-1, 1]$. Quanto mais próximo de 1, melhor a hierarquia se ajusta aos dados. Em geral, a validação com esse índice é realizada utilizando análise de Monte Carlo.

$$CPCC = \frac{(1/M)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(S_{ij}SC_{ij} - \mu_S\mu_{SC})}{[(1/M)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(S_{ij}^2 - \mu_S)]^{1/2}[(1/M)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(SC_{ij}^2 - \mu_{SC})]^{1/2}} \tag{30}$$

Alguns índices externos freqüentemente utilizados na validação de partições comparam uma partição resultante da aplicação de um algoritmo, P_e, com uma partição independente dos dados, construída com base na intuição ou conhecimento a priori sobre a estrutura real dos dados, P_r. São os casos da estatística *Rand* (R), dada pela Equação 31, do coeficiente de Jaccard (J), dado pela Equação 32, do índice de Folkes e Mallows (FM), dado pela Equação 33 e da estatística Huberts Γ normalizada, dada pela Equação 34. Um par de pontos, ou padrões, (x_v, x_u) é dito:

- SS: se pertencem ao mesmo cluster de P_e e ao mesmo cluster de P_r.
- SD: se pertencem ao mesmo cluster de P_e e a clusters diferentes de P_r.
- DS: se pertencem a clusters diferentes de P_e e ao mesmo cluster de P_r.
- DD: se pertencem a clusters diferentes de P_e e a clusters diferentes de P_r.

Sejam $a1$, $a2$, $a3$ e $a4$ os números de pares SS, SD, DS e DD, respectivamente. Define-se $M = a1 + a2 + a3 + a4$ como o número máximo de todos os pares no conjunto de dados ($M = n(n-1)/2$). Define-se também $m_1 = a1 + a2$ e $m_2 = a1 + a3$.

$$R = \frac{(a1 + a4)}{M} \tag{31}$$

$$J = \frac{a1}{(a1 + a2 + a3)} \tag{32}$$

$$FM = \frac{a1}{\sqrt{(m_1)(m_2)}} \tag{33}$$

$$\Gamma = \frac{Ma1 - m_1 m_2}{\sqrt{m_1 m_2 (M - m_1)(M - m_2)}} \tag{34}$$

Alguns índices medem a qualidade dos clusters gerados a partir dos conceitos de homogeneidade e separação, como os índices Dunn e Dunn-like, Davis Boulin e SD, resumidos por Halkidi et al. (2001). Esses índices medem em que grau os padrões são similares dentro de um cluster e diferentes em clusters separados.

Jiang et al. (2004) citam algumas abordagens comuns de validação empregadas em dados de expressão gênica. Segundo Jiang et al. (2004), Golub et al. (1999) avaliam a qualidade dos clusters com base na idéia de que se clusters supostos refletem a estrutura real, então um preditor de classes construído com base nesses clusters deve ter um bom desempenho. Jiang et al. (2004) resumem também as abordagens propostas por Yeung et al. (2000) e Tibshirani et al. (2001).

Yeung et al. (2000) apresentam a estatística figura de mérito (*Figure of merit*), FOM, para avaliação da qualidade do resultado de um agrupamento. Um algoritmo de agrupamento é aplicado a todos os padrões, exceto um padrão e. Esse padrão e é utilizado para estimar o poder preditivo, medindo sua similaridade intra-cluster (*within-cluster similarity*). A intuição por trás dessa medida é que a tendência de nível de expressão similar para genes no mesmo cluster indica um agrupamento significativo do ponto de vista biológico. Assim, quanto maior a similaridade intra-cluster de e mais forte é o poder preditivo e melhor o esquema de agrupamento. Seja $R(g, e)$ o nível de expressão do gene g sob a condição e na matriz de dados bruta. Seja $\mu_{C_i}(e)$ o nível de expressão médio na condição e dos genes no cluster C_i. $FOM(e, K)$ é dado pela Equação 35, para K clusters usando a condição e.

$$FOM(e, K) = \sqrt{\frac{1}{n} \sum_{i=1}^{K} \sum_{x \in C_i} (R(x, e) - \mu_{C_i}(e))^2} \tag{35}$$

Fazendo isso para cada uma das m amostras obtém-se a figura de mérito agregada, dada pela Equação 36, que é uma estimativa do poder preditivo total de um algoritmo sobre todas as amostras para K clusters.

$$FOM(K) = \sum_{e=1}^{m} FOM(e, K) \tag{36}$$

Tibshirani et al. (2001) dividem as amostras em um conjunto de treinamento X_{tr} e um conjunto de teste X_{te}. A idéia principal é usar o resultado do agrupamento gerado com a

aplicação de um algoritmo aos dados de treinamento para predizer a "co-pertinência" (se duas amostras pertencem ao mesmo cluster) no conjunto de teste.

Inicialmente, agrupa-se os dados de teste em K clusters A_{K1}, A_{K2}, ... A_{KK}. A operação de agrupamento é denotada por $C(X_{te}, K)$ e a "co-pertinência" é dada por $D[C(X_{te}, K), X_{te}]_{ii'} = 1$ se as observações i e i' estão no mesmo cluster e zero caso contrário. Em seguida, agrupa-se os dados de treinamento em K clusters e mede-se o quão bem os centros dos clusters do conjunto de treinamento predizem as "co-pertinências" no conjunto de teste. A força de predição (*prediction strength*), PS, de um agrupamento $C(., K)$ é dada pela Equação 37

$$PS(K) = \min_{1 \leq j \leq K} \frac{1}{n_{Kj}(n_{Kj} - 1)} \sum_{i \neq i' \in A_{Kj}} I(D[C(X_{tr}, K), X_{te}]_{ii'} = 1) \qquad (37)$$

Para cada cluster de teste é computada a proporção de pares observados nesse cluster que também foram associados ao mesmo cluster pelos centros dos clusters derivados do conjunto de treinamento. A força de predição é o mínimo dessa quantidade sobre os K clusters de teste. Quanto maior o valor de PS, melhor a qualidade do agrupamento.

7 Análise e Comparação de Algoritmos de Agrupamento

A análise e a comparação de algoritmos de agrupamento são tarefas bastante difíceis e que dependem muito de conhecimento, tanto do domínio da aplicação, como das técnicas de agrupamento empregadas.

Uma característica importante, inerente aos algoritmos de agrupamento, que torna difícil a análise do desempenho e a comparação de algoritmos de agrupamento é a ausência de uma estrutura ideal, que seja a resposta esperada para o agrupamento. Outras questões que tornam difícil a análise, escolha e comparação dos algoritmos são o grande número de algoritmos disponíveis e, segundo Estivill-Castro (2002), a falta de descrição explícita dos princípios indutivos e modelos descritos na literatura.

A análise do desempenho de algoritmos de agrupamento ainda é uma área em aberto. Atualmente, tal análise tem sido feita com base em conjuntos de dados que já têm uma estrutura conhecida. Segundo a literatura, alguns fatores parecem ter grande influência no desempenho das técnicas de agrupamento: a estrutura dos clusters (forma, tamanho, número de clusters), a presença de *outliers* e o grau de cobertura exigido, o grau de sobreposição dos clusters e a escolha da medida de similaridade (He 1999).

Segundo Jain & Dubes (1988) "Uma comparação teórica dos algoritmos de agrupamento não é factível porque os algoritmos de agrupamento são quase impossíveis de modelar de tal forma que os modelos possam ser comparáveis". Mas alguns critérios são úteis quando se deseja comparar diversos algoritmos de agrupamento. Em primeiro lugar deve-se ter uma idéia clara do princípio indutivo, ou, mais especificamente, do critério de agrupamento, no qual se baseia o algoritmo. Também é importante ter uma visão dos resultados gerados pelo algoritmo, ou seja, como ele representa os clusters gerados (modelo). Dado um contexto (princípio de indução e modelo) comum, pode-se observar

então características específicas dos algoritmos, relacionadas às habilidades do algoritmo, aos resultados que o algoritmo pode produzir, aos dados que ele suporta e à necessidade de interação com o usuário. Tais características são (Halkidi et al. 2001; Jiang et al. 2004):

Relacionadas ao algoritmo:

- Complexidade do algoritmo.
- Escalabilidade e eficiência para conjuntos de dados grandes.
- As medidas de similaridade que podem ser empregadas pelo algoritmo.
- Robustez relativa à ruídos e *outliers*.
- Se o algoritmo é capaz de lidar com dados de alta dimensionalidade ou encontra clusters em sub-espaços do espaço original.
- Estabilidade, ou seja, se para cada execução diferente do algoritmo, os dados são alocados aos mesmos clusters.
- Se o algoritmo é capaz de manipular incrementalmente a adição de novas entradas ou remoção de entradas antigas.

Relacionadas ao resultado:

- Forma dos clusters que o algoritmo é capaz de encontrar.
- Interpretabilidade dos resultados.

Relacionadas aos dados:

- Os tipos de dados que o algoritmo suporta (numéricos, categóricos, binários).
- Dependência da ordem dos dados.

Relacionadas à interação do usuário:

- Se o algoritmo encontra o número de clusters ou se o usuário deve fornecer esse número.
- Os parâmetros requeridos pelo algoritmo e o conhecimento do domínio requerido do usuário.

Estivill-Castro (2002) discute algumas questões referentes à falta de descrição explicita dos princípios indutivos e modelos em muitos dos algoritmos encontrados na literatura, o que pode gerar confusões sobre as propriedades dos algoritmos e tornar difícil a comparação entre eles. Algumas das observações e recomendações de Estivill-Castro são:

- Os algoritmos de agrupamento são categorizados mais com base nos modelos do que nos princípios de indução.
- Os pesquisadores devem tentar explicitar matematicamente os modelos e princípio indutivo dos algoritmos de agrupamento que estão propondo, facilitando com isso futuras investigações e comparações.

- Os índices de validade dos clusters são formulações matemáticas diretas de princípios de indução. Comparar algoritmos com bases nesses índices pode fornecer algumas dicas sobre os contextos nos quais um algoritmo funciona melhor do que outro, mas isso não implica que um algoritmo produz resultados mais válidos que outro. Dois algoritmos aplicados a um conjunto de dados que não possui estrutura irão ambos produzir resultados inválidos.

- Um algoritmo projetado para um universo de modelos não é adequado para conjuntos de dados que têm uma estrutura representável por uma família de modelos radicalmente diferente. Por exemplo, *k-means* não pode encontrar clusters não-convexos.

Em (Dubes & Jain 1976), um conjunto de critérios de admissibilidade é utilizado para comparar algoritmos de agrupamento. Estes critérios são baseados na maneira como os clusters são formados, na estrutura dos dados e na sensibilidade da técnica de agrupamento a mudanças que não afetem a estrutura dos dados. Além desses critérios de admissibilidade, existem algumas questões importantes a serem levadas em consideração, tais como: como os dados deveriam ser normalizados? qual medida de similaridade é apropriada para uma dada situação? como o conhecimento do domínio deve ser utilizado? e como um grande conjunto de dados pode ser agrupado eficientemente?

Assim, é essencial aos usuários dos algoritmos de agrupamento ter um bom entendimento da técnica particular que eles estão utilizando, conhecer detalhes do processo de obtenção dos dados, ter algum conhecimento do domínio e ter claramente definido o propósito do agrupamento que deseja obter, para que o agrupamento mais adequado para o problema em questão possa ser obtido.

O conhecimento a respeito dos dados é importante, por exemplo, para determinar as transformações necessárias aos dados antes do agrupamento e para escolher as medidas de similaridade que fazem sentido para esses dados. O conhecimento do domínio e o do propósito do agrupamento permitem determinar as características mais relevantes, os algoritmos de agrupamento mais apropriados e a forma de validação mais adequada.

Aldenderfer & Blashfield (1984) fornecem um guia para relatar estudos de agrupamento:

- Uma descrição não ambígua do método de agrupamento deve ser fornecida.

- A escolha da medida de similaridade (ou o critério estatístico, se um método iterativo for utilizado) deve ser informada claramente.

- O programa utilizado deve ser declarado.

- Os procedimentos utilizados para determinar o número de cluster devem ser explicados.

- Evidência adequada da validade da solução de agrupamento deve ser apresentada.

Jain & Dubes (1988) relatam brevemente alguns artigos sobre análise comparativa de algoritmos de agrupamento, que refletem algumas das abordagens presentes na literatura.

8 Conclusão

Os algoritmos de agrupamento existentes apresentam diferentes formas de explorar e verificar estruturas presentes em um conjunto de dados. Neste relatório, foram apresentadas as principais definições e os principais aspectos relacionados à agrupamento de dados. Foram destacadas as etapas necessárias para a realização do agrupamento em um conjunto de dados e detalhados alguns aspectos importantes dessas etapas.

Neste detalhamento, foram incluídas a preparação dos dados, a descrição de várias medidas de similaridade que podem ser empregadas em agrupamento, a descrição de vários algoritmos de agrupamento, a validação de agrupamentos e a análise e comparação de algoritmos de agrupamento.

Os algoritmos descritos foram: algoritmos hierárquicos baseados em métricas de integração, BIRCH, *k-means*, DENCLUE, HSC, CLICK, CAST, SOM, GCS, SOTA, CLIQUE e MAFIA. Existem vários outros algoritmos reportados na literatura. Neste relatório, os autores procuraram descrever aqueles mais relacionados à pesquisa que estão desenvolvendo.

Referências

Agrawal, R., J. Gehrke, D. Gunopulos, & P. Raghavan (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 94–105. ACM Press.

Aldenderfer, M., & R. K. Blashfield (1984). *Cluster Analysis*, Volume 44 of *Quantitative Applications in the Social Sciences*. London, Thousand Oaks, New Delhi: SAGE Publications.

Ankerst, M., M. M. Breunig, H.-P. Kriegel, & P. Sander (1999). Optics: Ordering points to identify the clustering structure. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pp. 49–60.

Barbara, D. (2000). An introduction to cluster analysis for data mining. `http://www-users.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf` [Acessado em 12/11/2003].

Ben-Dor, A., R. Shamir, & Z. Yakhini (1999). Clustering gene expression patterns. *Journal of Computational Biology 6*(3/4), 281–297.

Ben-Hur, A., D. Horn, H. Siegelmann, & V. Vapnik (2001). Support vector clustering. *Journal of Machine Learning Research 2*, 125–137.

Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA. `http://www.accrue.com/products/rp_cluster_review.pdf` [Acessado em 05/02/2004].

Braga, A. P., A. C. P. L. F. Carvalho, & T. B. Ludermir (2000). *Redes Neurais Artificiais: Teoria e Aplicações*. Livros Técnicos e Científicos (LTC).

Cheng, Y., & G. M. Church (2000). Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 93–103. AAAI Press.

Chiang, J.-H., & P.-Y. Hao (2003, Aug). A new kernel-based fuzzy clustering approach: support vector clustering with cell growing. *IEEE Transactions on Fuzzy Systems 11*(4), 518– 527.

Dopazo, J., & J. M. Carazo (1997). Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of Molecular Evolution 44*(2), 226–233.

Dubes, R., & A. Jain (1976). Clustering techniques: The user´s dilemma. *Pattern Recognition 8*, 247–260.

Duda, R., P. Hart, & D. Stork (2001). *Pattern Classification*. John Wiley & Sons.

Ertöz, L., M. Steinbach, & V. Kumar (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Proceedings of Workshop on Clustering High Dimensional Data and its Applications, 2nd SIAM International Conference on Data Mining (SDM'2002)*, pp. 105–115.

Ester, M., H.-P. Kriegel, J. Sander, & X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd*

International Conference on Knowledge Discovery and Data Mining (KDD'96), pp. 226–231.

Estivill-Castro, V. (2002). Why so many clustering algorithms - a position paper. *SIGKDD Explorations 4*(1), 65–75.

Fred, A. L. N. (2001, July). Finding consistent clusters in data partitions. In J. Kittler & F. Roli (Eds.), *MCS '01: Proceedings of the Second International Workshop on Multiple Classifier Systems*, Volume 2096 of *Lecture Notes in Computer Science*, Cambridge, UK, pp. 309–318.

Fritzke, B. (1994). Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks 7*(9), 1441–1460.

Getz, G., H. Gal, I. Kela, D. A. Notterman, & E. Domany (2003). Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics 19*, 1079–1089.

Golub, T., P. T. D.K. Slonim and, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, & E. Lander (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science 286*(5439), 531–537.

Gordon, A. (1999). *Classification*. Chapman & Hall/CRC.

Guha, S., R. Rastogi, & K. Shim (1998). CURE: an efficient clustering algorithm for large databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 73–84.

Guha, S., R. Rastogi, & K. Shim (2000). ROCK: A robust clustering algorithm for categorical attributes. *Information Systems 25*(5), 345–366.

Halkidi, M., Y. Batistakis, & M. Vazirgiannis (2001). On clustering validation techniques. *Intelligent Information Systems Journal 17*(2-3), 107–145.

Han, J., M. Kamber, & A. Tung (2001). *Geographic Data Mining and Knowledge Discovery*, Chapter Spatial Clustering Methods in Data Mining: A Survey. Taylor and Francis.

Hartigan, J. A. (1985). Statistical theory in clustering. *Journal of Classification 2*, 63–76.

Hartuv, E., & R. Shamir (2000). A clustering algorithm based on graph connectivity. *Information Processing Letters 76*(200), 175–181.

Hautaniemi, S., O. Yli-Harja, J. Astola, P. Kauraniemi, A. Kallioniemi, M. Wolf, J. Ruiz, S. Mousses, & O.-P. Kallioniemi (2003). Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps. *Machine Learning 52*(1-2), 45–66.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

He, Q. (1999). A review of clustering algorithms as applied in IR. Technical Report UIUCLIS–1999/6+IRG, Information Retrieval Group, University of Illinois.

Herrero, J., A. Valencia, & J. Dopazo (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics 17*(2), 126–136.

Hinneburg, A., & D. A. Keim (1998). An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pp. 58–65. AAAI Press.

Hinneburg, A., & D. A. Keim (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB'99: Proceedings of 25th International Conference on Very Large Data Bases*, pp. 506–517.

Jain, A., & R. Dubes (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jain, A., M. Murty, & P. Flynn (1999, September). Data clustering: A review. *ACM Computing Surveys 31*(3), 264–323.

Jiang, D., C. Tang, & A. Zhang (2004). Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering 16*(11), 1370–1386.

Karypis, G., E.-H. S. Han, & V. Kumar (1999). Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer: Special Issue on Data Analysis and Mining 32*(8), 68–75.

Kaufman, L., & P. J Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons.

Kohonen, T. (1997). *Self-organizing Maps*. Springer, Berlin.

Lazzeroni, L., & A. Owen (2002). Plaid models for gene expression data. *Statistica Sinica 12*(1), 61–86.

Luo, F., L. Khan, F. B. Bastani, I.-L. Yen, & J. Zhou (2004). A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles. *Bioinformatics 20*(16), 2605–2617.

Luo, F., K. Tang, & L. Khan (2003). Hierarchical clustering of gene expression data. In *3rd IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)*, pp. 328–335.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1: Statistics, pp. 281–297.

Nagesh, H., S. Goil, & A. Choudhary (2001a). Adaptive grids for clustering massive data sets. In *Proceedings of SIAM Conference on Data Mining (SDM'2001)*.

Nagesh, H., S. Goil, & A. Choudhary (2001b). *Data Mining for Scientific and Engineering Applications*, Chapter Parallel Algorithms for Clustering High-dimensional Large-Scale Datasets, pp. 335–356. Kluwer Academic Publishers.

Ng, R., & J. Han (1994). Efficient and effective clustering methods for spatial data mining. In *VLDB'94: Proceedings of 20th International Conference on Very Large Data Bases*, Santiago, Chile, pp. 144–155.

Nikkilä, J., P. Törönen, S. Kaski, J. Venna, E. Castrén, & G. Wong (2002). Analysis and visualization of gene expression data using self-organizing maps. *Neural Networks, Special issue on New Developments on Self-Organizing Maps 15*(8-9), 953–966.

Shamir, R., & R. Sharan (2002). *Current Topics in Computational Biology*, Chapter Algorithmic Approaches to Clustering Gene Expression Data, pp. 269–299. MIT Press.

Sharan, R., & R. Shamir (2000). CLICK: A clustering algorithm with applications to gene expression analysis. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 307–316.

Sheikholeslami, G., S. Chatterjee, & A. Zhang (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB'98: Proceedings of 24th International Conference on Very Large Data Bases*, New York City, pp. 428–439. Morgan Kaufmann.

Tamayo, P., P. Tamayo, D. Slonim, J. Mesirow, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, & T. R. Golub (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. In *Proc. Natl. Acad. Sci. USA*, Volume 96, pp. 2907–2912.

Tibshirani, R., G. Walther, D. Botstein, & P. Brown (2001). Cluster validation by prediction strength. Technical report, Department of Statistics, Stanford University.

Wang, W., J. Yang, & R. Muntz (1997). STING: A statistical information grid approach to spatial data mining. In *VLDB'97: Proceedings of 23rd International Conference on Very Large Data Bases*, Athens, Greece, pp. 186–195.

Yeung, K., D. Haynor, & W. Ruzzo (2000). Validating clustering for gene expression data. Technical Report UW-CSE-00-01-01, University of Washington, Department of Computer Science and Engeneering.

Zeng, Y., J. Tang, J. Garcia-Frias, & G. Gao (2002). An adaptive meta-clustering approach: Combining the information from different clustering results. In *IEEE Computer Society Bioinformatics Conference (CSB'02)*, Stanford, California, pp. 276.

Zhang, T., R. Ramakrishnan, & M. Livny (1996). BIRCH: an efficient data clustering method for very large databases. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, pp. 103–114.