3
Técnicas de agrupamento

Com o advento da internet a quantidade de informação disponível aumentou consideravelmente e com isso, tornou-se necessário uma forma automática de organizar e classificar esta informação.

O processo de agrupamento de objetos físicos ou abstratos em classes de objetos similares é conhecido também como clusterização. Um *cluster* é uma coleção de objetos de dados que são similares dentro de um mesmo grupo (*cluster*) e são dissimilares entre *clusters* distintos [4].

Ao longo deste capítulo apresentaremos os principais algoritmos e técnicas utilizadas no processo de agrupamento que serviu de base para o desenvolvimento desta dissertação.

3.1.
Tipos de agrupamento

De acordo com a necessidade da aplicação os dados a serem agrupados podem ser interpretados de diferentes maneiras. Deste modo, Tan et al. [5] apresentam alguns dos mais importantes e conhecidos tipos de agrupamento utilizados.

Hierárquico versus Particionado: No agrupamento particionado, os objetos são divididos em grupos no mesmo nível, ou seja, sem a sobreposição de *clusters* ou não-aninhada. No agrupamento hierárquico, os grupos de objetos estão aninhados, ou seja, estão organizados como em uma árvore.

Exclusivo versus Sobreposto versus *Fuzzy*: O agrupamento é dito exclusivo quando cada objeto de uma massa de dados está atribuído apenas a um *cluster*. Caso um objeto possa coexistir em diferentes clusters então dizemos que é não-exclusivo ou sobreposto. Já no agrupamento *fuzzy* cada objeto pertence a cada *cluster* de acordo com um grau de pertinência, onde um objeto

pertence a um *cluster* totalmente (100%), parcialmente (100% > x > 0%) ou não pertence (0%).

Completo versus parcial: No agrupamento completo, todos os objetos são atribuídos necessariamente a um *cluster*, ao contrário do agrupamento parcial, onde os objetos não são necessariamente atribuídos a um *cluster*. Alguns objetos podem não ser bem definidos e, então, não pertencerem a nenhum grupo.

3.2. Algoritmos de agrupamento

Algoritmos de agrupamento têm como objetivo particionar uma massa de objetos em grupos ou *clusters* de objetos similares. Esta tarefa pode ser dividida em agrupamento supervisionado, onde algum mecanismo externo (humano) provê informações de como classificar um determinado objeto corretamente, e agrupamento não-supervisionado, onde a classificação deve ser realizada sem referências externas.

3.2.1. Agrupamento supervisionado

De acordo com Sebastiani [6], o problema do agrupamento supervisionado é definido como:

Dado um grupo de D documentos e um grupo de C categorias pré-definidas, o objetivo é atribuir um valor booleano para cada par (d_i, c_j) ϵ D x C, onde d_i ϵ D e c_j ϵ C.

Muitos algoritmos de aprendizagem têm sido utilizados para ajudar nesse tipo de classificação, como por exemplo, k-Nearest Neighboor(k-NN) [7-9], Support Vector Machines(SVM) [10], Neural Networks(Nnet) [11, 12], Linear Least Squares Fit(LLSF) [13] e Naive Bayes(NB) [14, 15]. A seguir veremos o funcionamento de alguns dos algoritmos mais conhecidos.

3.2.1.1. K-Nearest Neighbor

Este algoritmo é considerado o mais simples dentre os algoritmos de machine learning (ML) e o seu propósito é classificar um novo objeto baseado nos exemplos de treinamento e em seus atributos.

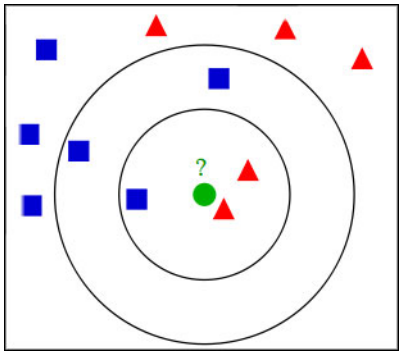

Figura 3 Exemplo k-NN

Dado um objeto x sem classificação, verifica-se os k vizinhos treinados mais próximos a ele. A categoria atribuída ao objeto x é a que possui o maior número de ocorrências (k-ocorrências) próximas a ele.

Como podemos ver na figura 3, o círculo representa o objeto sem classificação e os objetos restantes representam os objetos já treinados, ou seja, os vizinhos que ajudarão a classificar o novo objeto. Supondo que k seja definido como três (3-ocorrências), o círculo será classificado como triângulo, pois dentre os três vizinhos mais próximos dois deles são triângulos e apenas um é quadrado.

Existe uma variação do algoritmo k-NN chamada Weighted K-Nearest Neighbor (Wk-NN). Nele, a contribuição de cada vizinho é ponderada de acordo com a similaridade do objeto sem classificação, ou seja, para cada categoria a similaridade dos vizinhos pertencentes àquela categoria é somada obtendo um *score* para a categoria do objeto sem classificação, o objeto então é atribuído a categoria com o maior *score*. O cálculo para a categoria c_j com o objeto x é realizado através da form. (1).

$$score(cj, x) = \sum_{di\ \epsilon\ N(x)} \mathrm{y(di, cj)}.\cos(x, di) \qquad (1)$$

onde x é o objeto sem classificação; d_i é o objeto treinado, N(x) é o grupo de treinamento dos k objetos próximos a x; cos(x, d_i) é o cosseno da similaridade entre o objeto x e o objeto d_i; e y(d_i,c_j) é uma função cujo valor retornado é 1 se d_i pertence à categoria c_j e 0, caso contrário.

3.2.1.2. Classificador Naive Bayes

Este classificador, provavelmente, é o mais utilizado em *machine learning*. Ele é denominado ingênuo (*naive*) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa para nenhum outro evento. Mesmo considerado ingênuo, este algoritmo tem o melhor desempenho em várias tarefas de classificação, conforme pode ser visto em [16,17].

Existem dois modelos comuns para a classificação utilizando o método de Bayes, modelo multivariado de Bernoulli e modelo multinomial. Em ambos os modelos a classificação de teste é feita aplicando o teorema de Bayes vide form. (2).

$$P(cj|di) = \frac{P(cj).P(di|cj)}{P(di)} \quad (2)$$

onde d_i é o objeto a classificar e c_j é um objeto já classificado.

Se d_i pertence a uma categoria c_j, então devemos determinar $P(c_j|d_i)$, probabilidade posterior de c_j dado d_i, e atribuir a d_i a categoria com maior probabilidade posterior. Para conhecermos $P(c_j|d_i)$ é necessário conhecer as probabilidades prévias $P(c_j)$ e $P(d_i)$ que são baseadas nos objetos já treinados ou informações estatísticas conhecidas. E, por fim, calculamos $P(d_i|c_j)$ que, como já citado, pode ser estimado de acordo com o modelo multivariado de Bernoulli ou multinomial.

3.2.1.3. Support Vector Machine (SVM)

A SVM foi proposta inicialmente por Vapnik [18] para resolver problemas *two-class*, achar a superfície de decisão que separa maximamente as amostras de treinamento positiva e negativa de uma categoria. A figura 4 ilustra a idéia deste método para objetos linearmente separados que é a formulação mais simples deste tipo de problema.

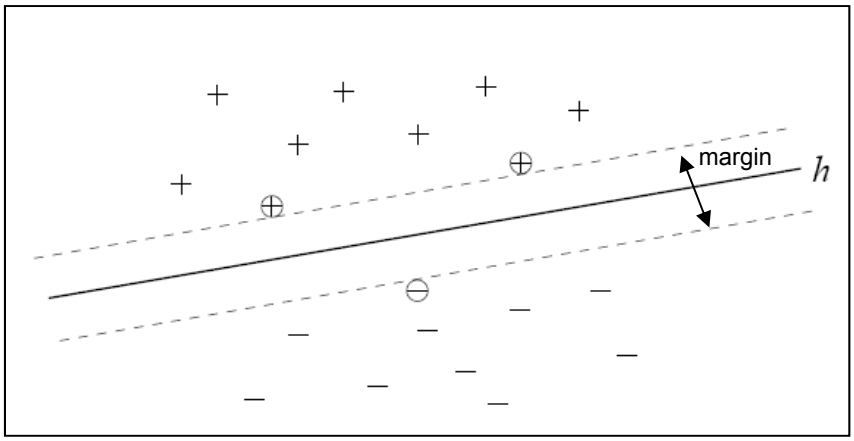

Figura 4 SVM localiza o hiperplano h, que separa as amostras de treinamento negativas e positivas com margem máxima. Os sinais circunscritos são chamados de Support Vectors.

A decisão de superfície é um hiperplano e quando linearmente separável pode ser escrita como a eq. (1).

$$w.d + b = 0 \tag{1}$$

onde d é um objeto a ser classificado e o vetor w e a constante b são conhecidas a partir do grupo de treinamento. Chama-se de hiperplano a superfície de decisão num espaço linearmente separado, o vão entre as linhas tracejadas mostra como a superfície de decisão pode ser movimentada sem conduzir a uma classificação errada e *margin* é a distância entre as linhas paralelas.

O problema SVM é justamente achar w e b de acordo com as restrições dadas por [10].

3.2.2. Agrupamento não-supervisionado

Ao contrário do agrupamento supervisionado, a utilização de categorias pré-definidas é desnecessária. O grande objetivo desta técnica é agrupar objetos com alto grau de semelhança, onde a similaridade é alguma função de distância, por exemplo, distância euclidiana.

De acordo com [19], a classificação não supervisionada pode ser dividida em dois tipos: algoritmos de agrupamento hierárquico e algoritmos de agrupamento particionado.

Algoritmos hierárquicos produzem partições aninhadas por divisão ou aglomeração e algoritmos particionados agrupam os dados em partições não aninhadas e não sobrepostas.

3.2.2.1. Algoritmos Particionados

3.2.2.1.1. K-Means

O algoritmo K-means [19] foi apresentado por J.B. MacQueen em 1967 e é um dos mais famosos algoritmos de agrupamento de dados, este algoritmo tenta fornecer uma classificação de acordo com os próprios dados, sendo a classificação feita por similaridade de grupos, onde o objeto é atribuído ao grupo (*cluster*) ao qual é mais semelhante.

A idéia por trás deste algoritmo é escolher k objetos (aleatoriamente ou com alguma heurística) que serão à base de cada grupo (denominados centróides), os demais objetos são associados ao centróide mais próximo. A cada passo os centróides são recalculados dentre os objetos de seu próprio grupo e os objetos são realocados para o centróide mais próximo, este procedimento é repetido até que o nível de convergência seja satisfatório de acordo com alguma heurística estabelecida, vejamos o exemplo da figura 5.

O pseudo-código do algoritmo K-Means é descrito como:

Algoritmo K-means

1 – Selecione K pontos como centróides iniciais
2 – repeat
3 – Atribua cada objeto ao *cluster* mais próximo
4 – Re-calcula cada centróide de cada *cluster*
5 – until (até que os centróides permaneçam estáveis)

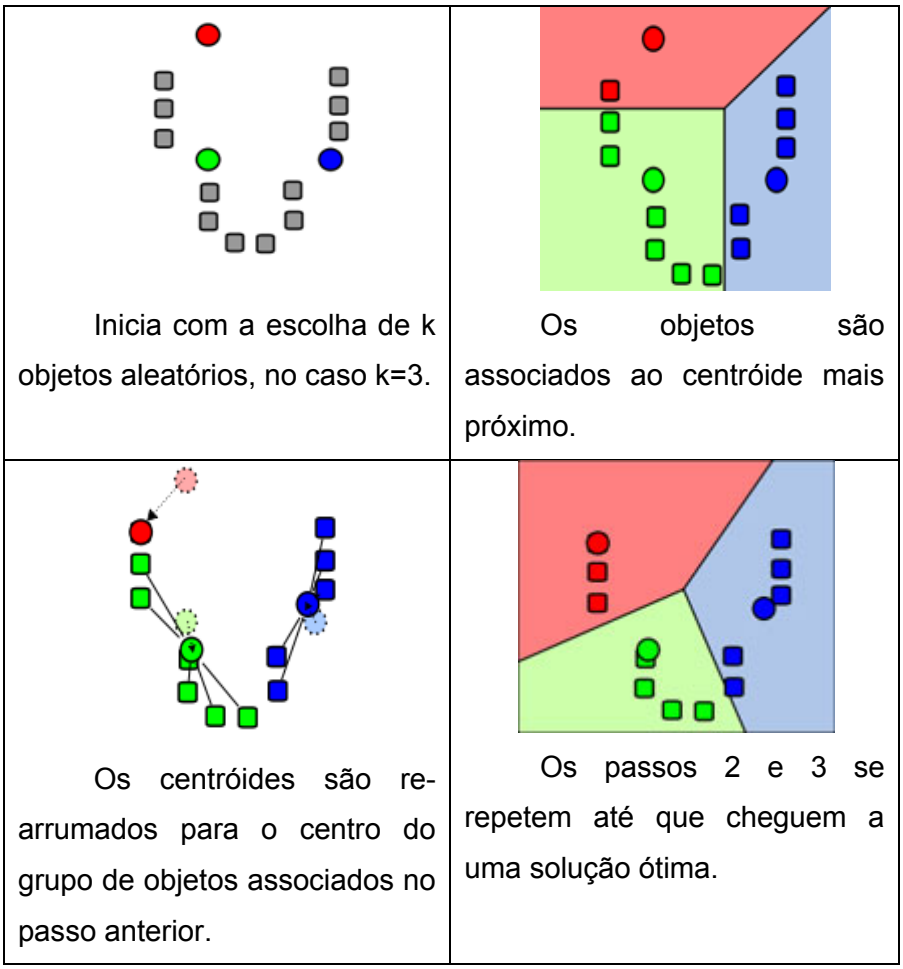

Figura 5 Exemplificação do algoritmo K-Means

3.2.2.1.2. K-Medóide

Esse algoritmo é uma variação do algoritmo K-Means e ao contrário deste o K-medóide escolhe objetos existentes como centróides. Ao final do agrupamento obteremos um objeto como o elemento central, normalmente classificado como o protótipo do agrupamento, o medóide.

Uma vantagem do algoritmo K-Medóide em relação ao K-Means é sobre os ruídos ou *outliers* (dados classificados erroneamente ou sem classificação) contidos no agrupamento, pois as estratégias na escolha do centróide e do medóide são diferentes. No K-Means, o centróide é dado pela média de todos os objetos dentro de um agrupamento. Desta maneira, se o agrupamento possuir um objeto muito distante dos outros, o centróide será influenciado erradamente. Já no K-Medóide, o algoritmo utiliza a média do erro quadrado para validar a escolha de um medóide então ao escolher um medóide que esteja mais próximo

a um ruído ou *outlier*, a média do erro quadrado irá aumentar e esta escolha será descartada, dado que o objetivo é minimizar a média do erro quadrado. Com isso, o algoritmo obtém o elemento mais representativo ou central do agrupamento.

3.2.2.2. Algoritmos hierárquicos

3.2.2.2.1. Divisão

O algoritmo de divisão [20] inicia-se com um *cluster* contendo todos os objetos disponíveis e a cada iteração o *cluster* mais apropriado no momento(*cluster* que possui a maior distância entre seus pares de objetos) é selecionado e dividido, o algoritmo pára quando algum critério pré-determinado é satisfeito, por exemplo, o número *k* de *clusters*.

Outro exemplo de algoritmo hierárquico de divisão é o Bisecting K-Means, a princípio ele é um algoritmo de partição, porém ele se torna hierárquico à medida que o algoritmo K-Means (k=2) é aplicado a cada partição selecionada a cada iteração. A partição selecionada pode ser aquela que possui mais objetos ou aquela em que a média do erro quadrado é alta.

Algoritmo Bisecting K-means
1 – Inicializa uma lista de *clusters* para conter o *cluster* que possui todos os objetos
2 – repeat
3 – Retira um *cluster* da lista de *clusters* (execute algumas vezes a bisecção do *cluster* escolhido)
4 – for 1 to número de tentativas
5 – Bi - seccione o *cluster* escolhido utilizando o K-means
6 – end for
7 – Seleciona os dois *clusters* com o menor erro (SSE)
8 – Adiciona estes dois *clusters* para a lista de *clusters*
9 – until (até que a lista de *clusters* possua k *clusters*)

3.2.2.2.2. Aglomeração

Ao contrário do algoritmo de divisão, no início, cada objeto corresponde a um *cluster*. A cada iteração os *clusters* com maior similaridade são agrupados até que algum critério de parada seja identificado. A representação clássica desse algoritmo é dada por uma árvore, também chamada de dendograma (figura 7). Vejamos o exemplo a seguir, onde temos vários objetos e a distância euclidiana é a medida de similaridade de acordo com a figura 6.

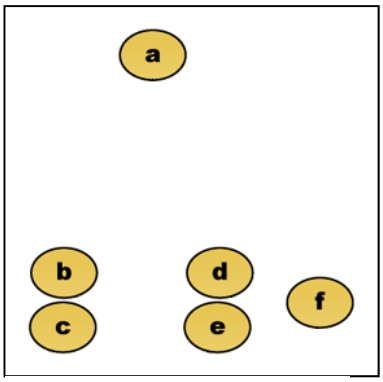

Figura 6 Objetos a serem aglomerados.

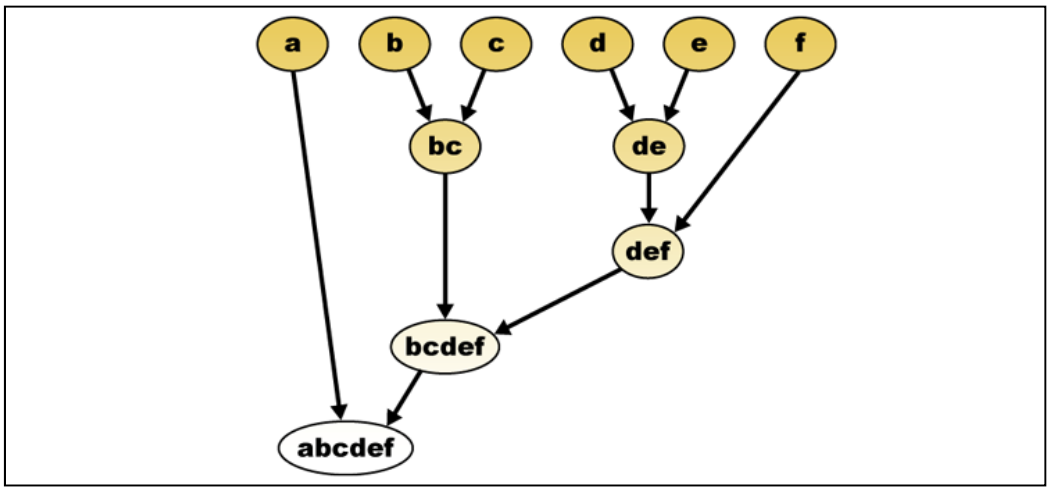

Figura 7 Dendograma do exemplo do algoritmo de aglomeração.

Na figura 7, temos seis elementos {a}, {b}, {c}, {d}, {e} e {f}. O primeiro passo foi determinar quais elementos deveriam ser aglomerados (pares de elementos com a menor distância). Este passo é iterado até o grau de generalização desejado.

3.3. Medidas de similaridade e dissimilaridade

As medidas de similaridade e dissimilaridade são fundamentais para a organização de objetos, seja para adicionar objetos a um determinado grupo ou retirá-los, respectivamente. As medidas mais comuns são:

- **Distância euclidiana**

- A distância euclidiana entre dois pontos x = (x_1, x_2,..., x_n) e y = (y_1, y_2,..., y_n) é dada pela form. (3).

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

onde x_i e y_i são as i coordenadas de cada objeto e x e y são objetos de uma massa de dados.

- **Distância de Manhattan**
  - A distância de Manhattan entre dois pontos x = (x_1, x_2,..., x_n) e y = (y_1, y_2,..., y_n) é dada pela form. (4).

$$d(x,y) = \sum_{i=1}^{n}\left|(x_i - y_i)\right| \quad (4)$$

onde x_i e y_i são as i coordenadas de cada objeto e x e y são objetos de uma massa de dados.

- **Similaridade do Cosseno**
  - É a medida do ângulo entre dois vetores de n dimensões. Cada objeto é representado por um vetor e a medida do ângulo entre eles representa o grau de similaridade dos dois objetos, caso o ângulo seja 0, significa uma relação total entre os objetos, caso o ângulo seja π, significa que não existe relação alguma entre os objetos e nos valores no intervalo (0, π), significa que existe alguma relação entre eles. A similaridade do cosseno é dada pela form. (5).

$$CosSim(A,B) = \frac{A \bullet B}{\|A\|\|B\|} \quad (5)$$

onde A e B são objetos representados como vetores.

3.4. Métodos de ligação sobre grupos

Os métodos de ligação são utilizados para determinar se a distância entre grupos é insuficientemente grande para que sejam reagrupados ou para agrupar os grupos mais próximos dentre todos os outros. Esses métodos são usualmente utilizados em algoritmos hierárquicos de aglomeração.

3.4.1. Single Linkage Clustering Method (SLINK)

A proximidade entre dois grupos é definida como a mínima distância entre dois objetos de dois diferentes grupos. Para isso, é computada a distância entre todos os pares de objetos, ver figura 8.

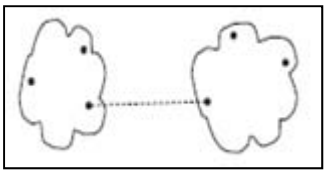

Figura 8 SLINK

3.4.2. Group Average Method ou Unweighted pair-group Method using Arithmetic Averages (UPGMA)

A proximidade entre dois grupos é definida como a média das distâncias entre cada objeto de um grupo e cada objeto do outro grupo, ver figura 9.

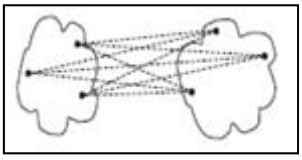

Figura 9 UPGMA

3.4.3. Complete Link Clustering Method (CLINK)

A medida de proximidade do CLINK é exatamente a oposta do SLINK, aqui a proximidade é definida como a máxima distância entre dois objetos de dois diferentes grupos, ver figura 10.

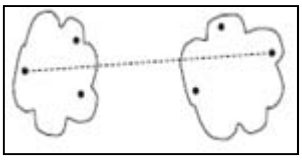

Figura 10 CLINK

3.4.4.
Ward's Method

A proximidade entre dois grupos é definida como o agrupamento de dois grupos no qual o aumento da perda seja mínimo. A perda é definida em termos da soma do erro quadrado. Calcula-se a soma do erro quadrado da junção de cada par de grupos e os que obtiverem o menor erro são agrupados.

3.5.
Métodos de validação de grupos

Para verificar a qualidade da estrutura de um grupo (*cluster*) é necessário informações sobre ele, estas informações podem ser obtidas através de métodos não-supervisionado, supervisionado ou relativo.

O método não-supervisionado mede a qualidade do grupo sem nenhuma informação externa, ou seja, usam-se apenas informações contidas no próprio grupo de dados, as medidas podem ser divididas em coesão e separação. A coesão valida a solidez dentro de um grupo e a separação valida o isolamento entre grupos, onde essas validações podem ser feitas através de medidas de proximidade de objetos. No caso dos sistemas baseados em protótipos a separação pode ser medida através da distância entre os protótipos e a coesão pode ser medida através da distância entre os objetos do grupo e seu protótipo, conforme ilustra a figura 11.

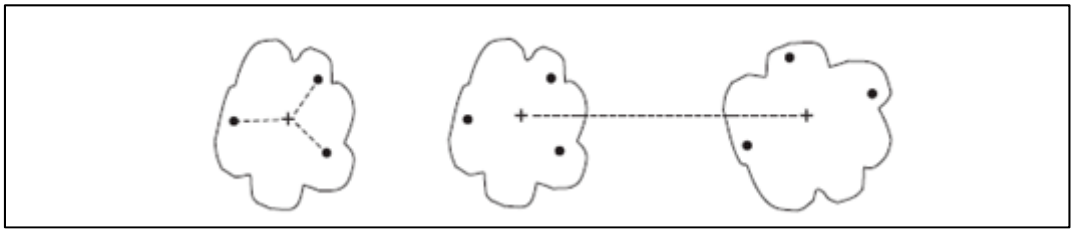

Figura 11 Ilustração da coesão de um grupo (à esquerda) e a separação de dois grupos (à direita) em relação a um elemento central (protótipo).

O método supervisionado, ao contrário do não-supervisionado, utiliza medidas externas para validar o agrupamento. Já o método relativo compara diferentes agrupamentos ou grupos, este método é tanto supervisionado como não-supervisionado e é utilizado como comparador entre as medidas com índices internos (diferentes agrupamentos) e índices externos (classificação já conhecida).

3.6. Tratamento de tipos de dados

Para que haja um agrupamento de dados é importante que sejam definidos os tipos de dados com os quais o algoritmo irá trabalhar. Para isso, Han e Kramber [21], especificaram em seu trabalho alguns tipos de variáveis para inferir sobre a similaridade de objetos no agrupamento de dados.

3.6.1. Variáveis escaladas em intervalos

Unidades de medida, como por exemplo: quilograma, litro, metro, entre outras. As medidas são escaladas para a unidade correta antes de serem aplicadas a medida de similaridade entre objetos.

3.6.2. Variáveis booleanas

Variáveis que possuem apenas dois tipos de valores (0,1), que representam se determinado objeto possui, ou não, determinada característica.

3.6.3. Variáveis nominais

Variáveis que possuem um conjunto finito de valores e não possuem uma ordem específica. Ex.: estado civil: (solteiro, casado, viúvo, divorciado).

3.6.4. Variáveis ordinais

Variáveis que possuem um conjunto finito de valores e uma ordem específica podendo assumir valores discretos ou contínuos. A avaliação deste

tipo de variável em sua medida de similaridade pode ser efetuada como as variáveis escaladas. Ex.: dias da semana (segunda-feira, terça-feira, ...)

3.6.5. Variáveis livres

Variável sem estrutura, texto livre.

3.7. Determinação do número de grupos

Embora muitos algoritmos de agrupamento sejam não-supervisionados, a maioria deles necessita de parâmetros de inicialização que estejam diretamente ou indiretamente ligados à determinação do número de grupos (*clusters).* Achar ou supor este número não é trivial, mesmo que tenhamos um conhecimento prévio sobre a massa de dados e dependendo da quantidade de informação essa tarefa se torna manualmente impossível. De acordo com Salvador e Chan [22], existem cinco abordagens normalmente utilizadas para tratar este problema.

3.7.1. Cross Validation

Método estatístico de particionamento de uma amostra de dados em subgrupos. Esses grupos servirão para treinar, validar e testar os grupos de dados formados.

3.7.2. Penalized likelihood estimation

Método que cria modelos para tentar ajustar os dados adequadamente, além de tentar diminuir sua complexidade.

3.7.3. Permutation tests

Método estatístico em que uma distribuição referencial é obtida através do cálculo de todas as possibilidades dos objetos de dados amostrados.

3.7.4.
Resampling

A técnica utiliza várias amostras do grupo de dados e tenta descobrir o número de grupos que é mais estável dentre essas amostras.

3.7.5.
Finding the knee of error curve

Método que tenta descobrir um número apropriado de grupos analisando a curva gerada a partir deste método; geralmente um teste é realizado para cada possível número de grupos e uma métrica para avaliação de cada grupo.

3.8.
Conclusão

O agrupamento de dados é uma técnica muito aplicada em diversas áreas como na biologia, estatística, psicologia, engenharia, medicina, marketing (análise de mercado), arquivologia, informática, negócios, entre muitas outras. Contudo, achar a melhor solução para este problema não é fácil, de acordo com [23], encontrar a configuração perfeita para os dados é um problema NP-Completo.

A utilização de algoritmos supervisionados tem um ponto diferencial conveniente, basear-se em uma massa de dados para treinamento já classificados, porém, em determinados casos, necessitamos uma forma de classificação automática, já que o ato de classificar dados para treinamento pode ser uma tarefa árdua ou, até mesmo, impossível. Deste modo, algoritmos de agrupamento não supervisionados têm um papel fundamental na classificação de dados similares.