Parte I

Capítulo

3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fundamentação Teórica

**3. O PROCESSO DE KDD – Knowledge Discovery in Databases**

**3.1 Introdução**

Os constantes avanços na área da tecnologia da informação têm viabilizado o armazenamento de grandes e múltiplas bases de dados. Tecnologias como a Internet, sistemas gerenciadores de banco de dados, dispositivos de armazenamento de dados de maior capacidade e de menor custo e sistemas de informação em geral são alguns dos exemplos que têm viabilizado a proliferação de inúmeras bases de dados de natureza comercial, administrativa, governamental e cientifica (Han & Kamber & Pei, 2011).

Atualmente, dados científicos em projetos de pesquisa, tais como missões espaciais da NASA, monitoramento temporal, em redes sociais, tais como Google, Facebook, etc., de um modo geral, o universo digital, têm alcançado proporções gigantescas, na ordem de zetabytes e ou terabytes de informações.

Diante desse cenário, a análise de grandes quantidades de dados pelo homem é inviável sem o auxílio de ferramentas computacionais apropriadas. Portanto, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação (Han; Kamber & Pei, 2011).

Então, para atender a este novo contexto, surge segundo Han; Kamber & Pei (2011), uma nova área denominada ***Data Mining***, que vem despertando grande interesse junto às comunidades cientificas e industrial.

Conforme Han; Kember & Pei (2011), o *Data mining (DM),* é um campo que tem conduzido a substanciais pesquisas em relação a mineração em vários tipos de dados, incluindo dados relacionais, dados de data *warehouse*, dados transacionais, dados de séries temporais, dados espaciais, dados textos, multimídia, dados da Web e de arquivos planos (*flat files*). A mineração de dados multidimensionais (também conhecida como *exploratory multidimensional data mining*, **online analytical mining**, ou **OLAM**) integra OLAP com mineração de dados para descobrir conhecimento em bases de dados multidimensionais. Dentre os muitos paradigmas e arquiteturas de sistemas de mineração de dados, a mineração de dados multidimensional, é particularmente importante, pelas seguintes razões:

* **Alta qualidade dos dados no data warehouse**. O data warehouse é uma valiosa fonte de dados de alta qualidade para o OLAP bem como para a mineração de dados.
* **Infraestrutura disponíveis em torno da mineração de dados**. A infraestrutura construída sistematicamente em torno do *data warehouse*, incluindo acesso, integração, consolidação e transformação de múltiplos bancos de dados heterogêneos, conexões ODBC/OLEDB, acesso Web, relatórios e ferramentas OLAP de análise.
* **Exploração de dados multidimensionais baseado em OLAP**. Mineração de dados multidimensionais fornecem facilidades para minerar diferentes subconjuntos de dados em vários níveis de abstração – por cortar e girar o cubo de dados em vários níveis (*drilling*, *dicing*, *slicing*) além de filtrar e girar o cubo os dados em vários ângulos.
* **Seleção online de funções de mineração de dados.** Nem sempre o usuário sabe o que está procurando. Então, por integrar OLAP com várias funções de mineração de dados, a mineração de dados multidimensional fornece usuário flexibilidade para selecionar as funções e tarefas de mineração de dados desejadas dinamicamente.

Não deve-se esquecer que, a finalidade da análise de dados é descobrir previamente características, relacionamentos, dependências ou tendências desconhecidas dos dados. Então, essas descobertas, segundo Rob (2011), tornam-se parte do modelo de informação a partir do qual as decisões são construídas. Portanto, uma típica ferramenta de análise de dados depende de os usuários finais definirem o problema, selecionarem os dados e iniciarem as análises adequadas, de modo a gerar informações que auxiliem na modelagem e resolução dos problemas descobertos por esses usuários.

Embora as ferramentas ***OLAP*** suportem a análise multidimensional e tomada de decisão, ainda segundo Han; Kember & Pei (2011), são necessárias as ferramentas de análise de dados para análises em profundidade, tais como: classificação de dados, associação, sumarização, árvores de decisão, caracterização, e agrupamento de dados (*clusting*), dentre outras.

Portanto, a grande quantidade de dados gerados, coletados ou armazenados, obtidos por operações diárias ou pesquisas cientificas, requer um processo automatizado para descobrir padrões, exceções, tendências ou correlações entre eles. Daí, surge a necessidade da Mineração de dados (Data Mining). Para Han; Kamber & Pei (2011), os desafios nessa área não param de crescer, e que ainda teremos a necessidade de um desenvolvimento ainda maior e que a Mineração de Dados será apenas um capítulo em uma área mais amplas: Ciência dos Dados (*Data Science* – como se registra na literatura Internacional).

O processo de KDD não se resume apenas a mineração de dados, mas esta é considerado, segundo alguns autores, a apenas uma das fases no processo para geração do conhecimento a partir de Bases de Dados. Na sessão seguinte será apresentado todas as fases do processo de KDD.

**3.2 Caracterização do Processo de KDD**

Basicamente, uma aplicação de KDD é composta por três tipos de componentes: o problema em que será aplicado o processo de KDD, os recursos disponíveis para a solução do problema e os resultados obtidos a partir da aplicação dos recursos disponíveis em busca da solução do problema (Passos & Goldschmidt, 2005).

1. **O problema a ser submetido ao processo de KDD** - Este componente pode ser caracterizado por três elementos: conjunto de dados, o especialista do domínio da aplicação e objetivos da aplicação.
2. **Os recursos disponíveis para solução do problema em questão** - Entre eles podem ser destacados: o especialista em KDD, as ferramentas de KDD e plataforma computacional disponível (Passos & Goldschmidt, 2005).
3. **Os resultados obtidos a partir da aplicação dos recursos no problema** - Compreende, fundamentalmente, os modelos de conhecimento descobertos ao longo da aplicação de KDD e o histórico das ações realizadas (Passos & Goldschmidt, 2005).

Segundo Han et al. (2011); Witten & Frank (2005); Elmasri (2005), o processo de KDD é mostrado na Figura 3.1 e consiste da sequência iterativa dos seguintes passos: **Seleção de dados, Limpeza dos dados, Enriquecimento, Transformação de dados, Data Mining** e **Avaliação dos padrões e representação do conhecimento**.

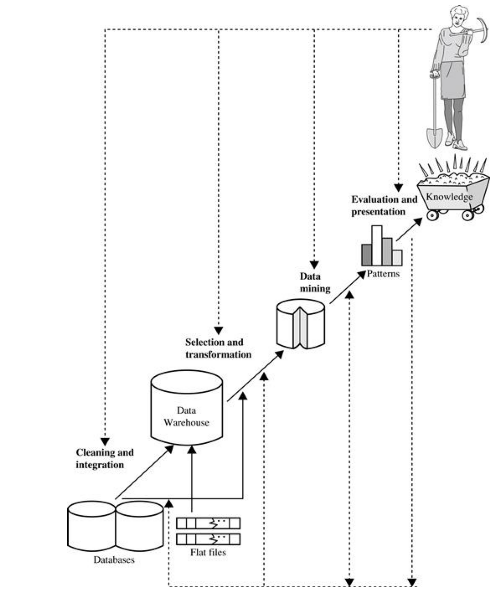
Do passo 1 até o passo 4, são as diferentes maneiras de pré-processamento dos dados, onde os dados são preparados para mineração. A etapa de *data mining*, pode interagir com o usuário ou com a base de conhecimento. O conhecimento encontrado, são apresentado ao usuário e pode ser armazenado como um novo conhecimento na base de conhecimento. A **Figura** 3.1, mostra o data mining, como uma etapa no processo de descoberta de conhecimento (*Knowledge Discovery Process*).

Estas etapas ainda podem, segundo alguns autores, serem subdividas em Pré-processamento, Mineração de Dados e Pós-processamento.

**3.3 Pré-processamento**

A etapa de pré-processamento compreende todas as funções relacionadas à captação, à organização e ao tratamento dos dados. Essa etapa tem como objetivo a preparação dos dados para o algoritmo da etapa da Mineração de Dados. As principais funções de pré-processamento são: **Seleção de dados**, **Limpeza dos dados**, **Transformação dos dados**, **Enriquecimento dos dados**, **Partição do Conjunto de Dados** e a **Integração dos Dados**.

**Figura 3.1** Mineração de Dados como um passo no processo de Descoberta de Conhecimento. **Fonte**: Han; Kamber & Pei (2011).



Segundo Witten & Frank (2005), associados a essa etapa, tem-se o conceito de variáveis do problema, também denominado de atributos, que podem ser classificados sob dois aspectos distintos: quanto à representação de seus valores (tipo de dados) e quanto à natureza da informação (tipo de variável). Os tipos de dados indicam a forma em que eles estão armazenados. Os tipos de variáveis expressam a natureza com que a informação deve ser interpretada. Dessa forma, as variáveis podem ser classificadas em (Witten, 2005):

* **Nominais ou Categóricas** – são variáveis utilizadas para nomear ou atribuir rótulos a objetos. Podem assumir valores pertencentes a um conjunto finito e pequeno de estado possíveis. Por exemplo, o gênero de uma pessoa: masculino e feminino. Nesse tipo de variável não há uma ordenação de seus valores ((Passos & Goldschmidt, 2005; Witten, 2005).
* **Discretas** – são similares às variáveis nominais, no entanto, os valores (estados) que elas assumem possuem um ordenamento, e este possui algum significado. O dia da semana é um bom exemplo deste tipo de variável, onde a “segunda-feira” vem após o “domingo” e antes da “terça-feira”, e assim sucessivamente. Podem também, ser intervalos em um espectro contínuo de valores. Por exemplo, faixa de renda e faixa etária, entre outros. De forma similar às variáveis nominais, os valores de variáveis discretas podem ser representadas por tipos de dados alfanuméricos (Passos & Goldschmidt, 2005; Witten, 2005).
* **Contínuas** – são variáveis quantitativas cujos valores possuem uma relação de ordem entre eles. O conjunto de valores de uma variável contínua pode ser finito ou infinito. Como exemplo, pode-se citar renda e idade como variáveis contínuas. Por sua natureza, normalmente os valores das variáveis contínuas são representadas por um tipo de dados numérico (Passos & Goldschmidt, 2005; Witten, 2005).
  + 1. **Seleção de dados**

Essa função, também denominada Redução de Dados, compreende, em essência, a identificação de quais informações, dentre as bases de dados existentes, devem ser efetivamente consideradas durante o processo de KDD. A seleção dos dados pode ter dois enfoques distintos: a escolha de atributos ou a escolha de registros que devem ser considerados no processo de KDD (Passos & Goldschmidt, 2005).

Deve-se mencionar que, em algumas situações, no início do processo de KDD, o especialista de KDD, recebe o conjunto de dados a ser analisado, não estruturados adequadamente para o processo de descoberta de conhecimento. Portanto, é função do especialista de KDD fazer esta estruturação nos dados. Para isto existem algumas técnicas, as quais serão discutas a seguir:

* + - 1. **Redução de Dados Horizontal**

A seleção por redução horizontal é caracterizada pela escolha de casos. Entre as operações de redução de dados horizontal podem ser citadas: amostragem aleatória, eliminação direta de casos, segmentação do banco de dados e agregação de informações (Passos & Goldschmidt, 2005).

* + - 1. **Segmentação do Banco de Dados**

Na segmentação deve-se escolher um ou mais atributos para nortear o processo de segmentação. Suponha que deseje-se apenas analisar os clientes que moram em residência própria. Tal operação poderia ser implementada por uma instrução de seleção em SQL do tipo: “Select \* FROM Cliente WHERE TP\_RES = “P””.

Dessa forma, o conjunto de dados resultantes desta consulta se tornaria o conjunto a ser efetivamente considerado deste ponto em diante no processo de KDD (Passos & Goldschmidt, 2005; Witten, 2005).

* + - 1. **Eliminação Direta de Casos**

Esta operação pode ser interpretada como uma variação da segmentação, e nela são especificados os casos a serem eliminados e não os casos que devem permanecer na análise. Considerando o mesmo exemplo anterior, acima citado, tal operação poderia ser implementada por uma instrução de exclusão em SQL do tipo “DELETE FROM Cliente WHERE TP\_RES <> “P””.

* + - 1. **Amostragem Aleatória**

Esta operação consiste de sortear da base de dados um número preestabelecido de registros de forma que o conjunto resultante possua menos registros do que o conjunto original. Existem várias estratégias de amostragem aleatória. A seguir será ilustrada algumas delas. Suponha que tem-se uma grande massa de dados, contendo N *tuplas*, e que n seja o número de amostras desejadas (n < N) (Passos & Goldschmidt, 2005).

* **Amostragem Aleatória Simples sem Reposição** – neste caso, todas as *tuplas* possuem a mesma probabilidade de seleção (1/N). Cada *tupla* selecionada é excluída do conjunto de dados original durante o processo de forma a evitar uma nova seleção.
* **Amostragem Aleatória Simples com Reposição** - também neste caso, todas as *tuplas* possuem a mesma probabilidade de seleção (1/N). No entanto, cada tupla selecionada é mantida no conjunto de dados original durante o processo de forma que a mesma *tupla* possa ser selecionada novamente.
* **Amostragem de Clusters (Agrupamentos)** – neste caso, as *tuplas* devem estar agrupadas em M clusters de tal forma que possa ser realizada uma amostragem aleatória dentre os clusters. Por exemplo, as *tuplas* de um banco de dados normalmente são recuperadas uma página por vez. Cada página pode ser considerada um cluster de tal forma que a Amostragem de Cluster possa ser aplicada.
* **Amostragem Estratificada** – neste caso, o conjunto de dados deve ser divido em grupos disjuntos. A amostragem estratificada consiste em selecionar aleatoriamente um subconjunto de amostras de cada grupo. Essa abordagem auxilia na obtenção de amostras representativas, especialmente em situações que apresentam dados tendenciosos. Por exemplo, imagine os clientes do exemplo anterior, separados em grupos por faixa etária. Aplicando a Amostra Estratificada, assegura-se que mesmos os clientes da faixa etária com menor número de elementos sejam representados na amostra final.
  + - 1. **Agregação de Informações**

Esta operação consiste em reunir (agregar) os dados de forma a reduzir o conjunto de dados original. Na agregação de informações, dados em um nível maior de detalhamento são consolidados em novas informações com menor detalhe. Por exemplo, somar os valores de todas as compras de cada cliente, obtendo o total de despesas por ele realizadas durante um determinado período (Passos & Goldschmidt, 2005).

* + - 1. **Redução de Dados Vertical**

Segundo Passos & Goldschmidt (2005); Han; Kamber & Pei (2011), esta é uma operação de pré-processamento muito importante no processo de KDD. Formalmente, sendo **S** um conjunto de dados com atributos **A1**, **A2**, **A3**, ..., **An**, o problema da redução de dados vertical consiste em identificar qual das 2n combinações entre tais atributos deve ser considerada no processo de descoberta de conhecimento. Ou seja, a redução de dados vertical, também conhecida como redução de dimensão, é implementada pela eliminação ou pela substituição dos atributos de um conjunto de dados. Tem como objetivo procurar encontrar um conjunto mínimo de atributos de tal forma que, a informação original seja preservada. Obviamente, quanto maior for o valor de n, maior será o desafio na escolha dos atributos: o número de possibilidades de subconjuntos de atributos cresce exponencialmente na medida em que n aumenta (Passos & Goldschmidt, 2005).

Ainda segundo Passos & Goldschmidt (2005); Han & Kamber (2011), entre as principais motivações para a aplicação da redução de dados verticais podem ser citadas: (1) um conjunto de atributos bem selecionados pode conduzir a modelos de conhecimento mais conciso e com maior precisão; (2) se o método de seleção dos atributos for rápido, o tempo de processamento necessário para utilizá-lo e, em seguida, aplicar o algoritmo de mineração de dados em um subconjunto dos atributos, pode ser inferior ao tempo de processamento para aplicar o algoritmo de mineração de dados sobre todo o conjunto de atributos; (3) a eliminação de atributo é muito mais significativa em termos de redução do tamanho de um conjunto de dados, do que a exclusão de registros.

Segundo Freitas (2002), existem duas abordagens para a redução de dados vertical, usualmente utilizadas em problemas de Classificação:

* **Abordagem Independente de Modelos** (*Filter*) – nessa abordagem, a seleção de atributos é realizada sem considerar o algoritmo de Mineração de Dados que será aplicado aos atributos selecionados.
* **Abordagem Dependente de Modelo** (*Wrapper*) – essa abordagem consiste em experimentar o algoritmo de Mineração de Dados para cada conjunto de atributos e avaliar os resultados obtidos.

Segundo Han; Kamber & Pei (1999), em quaisquer das abordagens mencionadas acima, há três estratégias clássicas e simples para a escolha do conjunto de atributos, que podem ser utilizadas:

* **Seleção Sequencial para Frente** (*Forward Selection*) – Essa seleção começa com o subconjunto de atributos candidatos vazio. O processo é iterativo. Cada atributo por vez é adicionado ao subconjunto de atributos candidatos, que é avaliado segundo alguma medida de qualidade. Ao final de cada iteração, é incluído no subconjunto de atributos candidatos, aquele atributo que tenha maximizado a medida de qualidade considerada.
* **Seleção Sequencial para Trás** (*Backward Selection*) – É o processo contrário da Seleção Sequencial para Frente: o subconjunto de atributos candidatos começa completo, com todos os atributos do problema. Então, cada atributo é retirado subconjunto, que é avaliado segundo alguma medida de qualidade. Ao final de cada iteração, é excluído do subconjunto de atributos candidatos, aquele atributo que tenha minimizado a medida de qualidade considerada.
* **Combinação das Estratégias Anteriores** – A seleção para frente e a seleção para trás são combinadas de tal forma que, a cada passo do algoritmo, o algoritmo seleciona o melhor atributo (incluindo-o o subconjunto de atributos candidatos) e remove o pior atributo dentre os remanescentes do conjunto de atributos.

De acordo com Han; Kamber & Pei (2011); Freitas (2002), existem estratégias mais interessantes para a escolha do conjunto de atributos a ser utilizados. Como por exemplo, os algoritmos Genéticos, podem ser utilizados no processo de otimização do conjunto de atributos. Outro exemplo, seria os Algoritmos para indução de Árvores de Decisão, tais como **ID3** e **C4.5**, que também podem ser aplicados para selecionar atributos em problemas de classificação. Ainda segundo os autores citados, uma árvore de decisão é construída a partir de dados, de tal forma que todos os atributos que não aparecem na árvore são considerados irrelevantes para o problema.

* + - 1. **Eliminação Direta de Atributos**

Segundo Passos & Goldschmidt (2005), essa tarefa se refere à eliminação de atributos cujo conteúdo não seja relevante ao processo de KDD. A sua utilização depende do conhecimento prévio existente acerca do problema. Essa operação pode ser implementada por uma instrução SQL.

Ainda segundo Passos & Goldschmidt (2005), duas heurísticas podem ser utilizadas para indicar que essa operação deve ser usada:

1. Eliminar todos os atributos que representem valores constantes em todos os registros da base de dados. Uma vez que, atributos nesta situação não contribuem para distinguir os registros uns dos outros, sendo, portanto, dispensáveis do processo de KDD.
2. Eliminar os atributos que sejam identificadores na base de dados em análise. Tal fato, justifica-se, uma vez que essas informações são usadas para identificar unicamente cada registro da base de dados.
   * + 1. **Redução de Valores**

A operação de redução de valores, segundo Passos & Goldschmidt (2005), é uma alternativa interessante à operação de corte de atributos oferecida pela redução de dados vertical. Essa operação consiste em reduzir o número de valores distintos em determinados atributos. Com menos valores, menos comparações são feitas, reduzindo o tempo de processamento de diversos algoritmos de Mineração de Dados. Existem métodos de redução de valores contínuos e métodos de redução de valores nominais.

* + - * 1. **Redução de Valores Nominais**

Essa operação é aplicável somente a variáveis nominais, onde o número de valores é finito, sem ordenação entre os mesmo. A seguir serão apresentados dois métodos (Passos & Goldschmidt, 2005):

* **Identificação de Hierarquia entre Atributos** – Neste caso, o especialista no domínio da aplicação deve apresentar a hierarquia (abstração) existente entre os atributos. Como exemplo, considere um endereço composto pelas seguintes informações: logradouro, bairro, cidade, estado. O especialista da aplicação por meio de uma ordenação, pode definir a hierarquia entre esses atributos como: logradouro ⊂ bairro ⊂ cidade ⊂ estado. A partir desta especificação, pode-se estabelecer um nível de corte do detalhamento da informação.
* **Identificação de Hierarquia entre Valores** – Aqui, o especialista no domínio da aplicação deve apresentar possíveis generalizações para os valores de cada atributo. Como exemplo, considere um conjunto de dados que contenham informações sobre os produtos vendidos: tênis, sapato, sandália, bermudas, calça, camisa, paletó. Então, o especialista pode definir hierarquias para indicar generalizações de conceito com relação aos valores existentes: tênis ⊂ calçado, sapato ⊂ calçado, sandália ⊂ calçado, bermuda ⊂ roupa, calca ⊂ roupa e paletó ⊂ roupa. Em outras palavras, as três primeiras generalizações indicam que tênis, sapato e sandália são casos particulares de calçado. De forma análoga, bermuda, calça e paletó são casos particulares de roupa. A partir desta especialização, os valores originais podem ser substituídos pelas respectivas generalizações, reduzindo um domínio de 7 valores distintos para apenas 2.
  + - * 1. **Redução de Valores Contínuos (ou Discretos)**

De acordo com Passos & Goldschmidt (2005), esse tipo de operação pressupõe aplicação somente a variáveis contínuas ou discretas. Variáveis contínuas ou discretas possuem uma relação de ordenação entre seus valores. A seguir serão apresentados alguns métodos para esse fim.

* **Particionamento em Células (*Bins*) de mesma Cardinalidade (“*Equidepth Bins*”)** – Nesse método, os valores são agrupados em células com o mesmo número de elementos em cada uma delas.
* **Redução de Valores pelas Medianas das Células (“Bin Medians”)** – O mesmo procedimento descrito para o método anterior é realizado neste método. Em seguida, é calculada a mediana de cada uma das células. Os valores originais são substituídos pela mediana associada a cada célula, gerando um novo conjunto de dados.
* **Redução de Valores pelas Médias das Células (“Bin Means”)** - O procedimento deste método é análogo ao descrito para o método anterior. Nesse caso, é calculada a média dos valores em cada uma das células. Os valores originais são substituídos pela média associada a cada célula, gerando um novo conjunto de dados.
* **Redução de Valores pelos Limites das Células (“Bin Boundaries”)** – O mesmo procedimento descrito para o primeiro método é realizado neste método. Em seguida, os valores nos extemos das células são considerados. O procedimento calcula a distância de cada valor em relação aos extremos de cada célula. O valor original é substituído pelo valor do extremo mais próximo, gerando um novo conjunto de dados.
* **Arredondamento de Valores** – O arredondamento de valores, também chamado de aproximação de valores, é uma função comum em nosso dia-a-dia. A seguir é apresentada uma alternativa para arredondamento de valores em números inteiros. A variável x indica o valor original a ser arredondado. A variável y recebe o resultado intermediário do procedimento de arredondamento. O parâmetro k é o número de casas decimais à direita a ser considerado no arredondamento.

**(3.1)**

* **Agrupamento de Valores (“Agrupamento”)** – Consiste em agrupara os valores de um atributo em clusters (grupos) levando em consideração a similaridade existente entre tais valores. O processo de Agrupamento, que será detalhado em sessões seguintes, consiste em procurar minimizar a similaridade dos dados pertencentes ao mesmo cluster e maximizar a similaridade dos dados em clusters distintos. Uma vez concluído o processo de agrupamento, cada cluster pode passar a ser representado pelo média dos valores a ele atribuído. Deve-se ressaltar que o problema da redução de valores pode ser interpretado como um problema de otimização.

De acordo com Passos & Goldschmidt (2005), duas heurísticas podem ser utilizadas para auxiliar na escolha de qual, dentre os métodos acima, deve ser utilizado:

* Quando o número de células for de moderado a grande, recomenda-se a utilização dos métodos de redução de valores pela mediana ou pela média.
* Quando o número de células for pequeno, recomenda-se a utilização do método de redução de valores pelos limites das células.
  + 1. **Limpeza dos Dados**

Abrange qualquer tratamento realizado sobre os dados selecionados de forma a assegurar a qualidade (completude, veracidade e integridade) dos fatos por eles representados. Informações ausentes ou inconsistentes nas bases de dados devem ser corrigidas de forma a não comprometer a qualidade dos modelos de conhecimento a serem extraídos ao final do processo de KDD (Passos & Goldschmidt, 2005).

Em aplicações reais, é comum que os dados sobre os quais se deseja extrair algum conhecimento estejam incompletos, ruidosos ou inconsistentes. Os dados são considerados incompletos se há informações ausente para determinados atributos ou ainda se há dados pouco detalhados. Por outro lado, dados ruidosos são dados errados ou que contenham valores considerados divergentes, denominados de *outliers*, do padrão normal esperado. Dados inconsistentes são aqueles que contêm algum tipo de discrepância semântica entre si (Passos & Goldschmidt, 2005).

É importante perceber que a qualidade dos dados possui grande influência na qualidade dos modelos de conhecimento a serem abstraídos a partir destes dados. Quanto pior for a qualidade dos dados informados ao processo de KDD, pior será a qualidade dos modelos de conhecimento gerados (GIGO, Garbage in, Garbage out) (Passos & Goldschmidt, 2005).

Para se fazer a limpeza dos dados, existem algumas funções. A seguir serão discutidas as algumas delas:

* + - 1. **Limpeza de Informações Ausentes**

Esta função compreende a eliminação de valores ausentes em conjunto de dados. Alguns métodos para preenchimento de valores ausentes estão descritos a seguir:

* **Exclusão de Casos** – Consiste de excluir do conjunto de dados as *tuplas* que possuam pelos menos um atributo não preenchido.
* **Preenchimento Manual de Valores** – em geral, essa abordagem demanda alto consumo de tempo e recursos, sendo muitas vezes inviável na prática. Esse método pode ser implementado por meio de pesquisas junto às fontes de dados originais que procurem captar as informações ausentes.
* **Preenchimento com Valores Globais Constantes** – esse método consiste em substituir todos os valores ausentes de um atributo por um valor padrão tal como “desconhecido” ou “null”. Esse valor padrão pode e deve ser especificado pelo especialista no domínio da aplicação.
* **Preenchimento com Medidas Estatísticas** – medidas estatísticas podem ser empregadas como alternativa à utilização de constantes padrão no processo de preenchimento de valores ausentes. Como exemplo de medidas estatísticas para preenchimento de informações ausentes pode ser citados: média para atributos numéricos e moda para atributos categóricos.
* **Preenchimento com Métodos de Mineração de Dados** – nesse caso, modelos preditivos podem ser construídos de forma a sugerir os valores mais prováveis a serem utilizados no preenchimento de valores ausentes. Algoritmos de Mineração de Dados tais como Redes Neurais, Estatística (modelos Bayesiano) e Árvores de Decisão são alternativas na construção destes modelos.
  + - 1. **Limpeza de Inconsistências**

De acordo com Passos & Goldschmidt (2005), essa função compreende a identificação e a eliminação de valores inconsistentes em conjunto de dados. Uma inconsistência pode envolver uma única tupla, ou um conjunto de *tuplas*. Inconsistência em uma única *tupla* ocorre quando houver divergência entre os valores desta *tupla*. Nesta situação, a participação do especialista no domínio da aplicação é fundamental na identificação de inconsistências. Além do mais, esse processo demanda conhecimento prévio acerca do problema. Existem alguns métodos para limpeza de inconsistências, como descrito a seguir:

* **Exclusão de Casos** – Consiste em excluir do conjunto de dados original, as *tuplas* que possuem pelo menos uma inconsistência. A identificação dos casos com inconsistência pode ser obtida por meio de instruções SQL, cujas restrições especifiquem o tipo de inconsistência a ser verificada.
* **Correção de Erros** – Esse método consiste em substituir valores errôneos ou inconsistentes identificados no conjunto de dados. Pode envolver desde a correção manual até a atualização desses valores em um lote predeterminado de registros.
  + - 1. **Limpeza de Valores não Pertencentes ao Domínio**

Segundo Passos & Goldschmidt (2005), essa função compreende a identificação e a eliminação de valores que não pertençam ao domínio dos atributos do problema. Ela pode ser considerada um caso particular da operação anterior e demanda um conhecimento prévio do domínio de cada atributo. A seguir temos alguns dos métodos utilizados para limpeza de valores não pertencentes ao domínio:

* **Exclusão de Casos** – Consiste em excluir do conjunto de dados original, as tuplas que possuam pelo menos um valor fora do conjunto de valores válidos de cada atributo.
* **Correção de Erros** – Consiste em substituir os valores inválidos identificados no conjunto de valores. Pode envolver desde a correção manual até a atualização destes valores em um lote predeterminado de registros utilizando instruções SQL.
  + 1. **Transformação dos Dados**

Segundo Passos & Goldschmidt (2005), codificação de dados é a operação de pré-processamento responsável pela forma como os dados serão representados durante o processo de KDD.

Deve-se salientar que os dados devem ser codificados de forma a atender às necessidades específicas dos algoritmos de Mineração de Dados. Por exemplo, uma rede neural requer que os dados estejam em uma representação numérica. Dessa forma, se a base de dados a ser processada apresente valores nominais, estes devem ser codificados antes de serem submetidos a rede (Passos & Goldschmidt, 2005).

A maneira como a informação é codificada tem forte influência sobre o tipo de conhecimento a ser encontrado. Em essência, a codificação pode ser: Numérica – Categórica, que transforma valores reais em categorias ou intervalos; ou Categórica – Numérica, que representa numericamente valores de atributos categóricos (Passos & Goldschmidt, 2005).

* + - 1. **Codificação: Numérica – Categórica**
* **Mapeamento Direto** – Consiste na simples substituição dos valores numéricos por valores categóricos. Por exemplo, sexo: 1 🡪 M, 0 🡪 F.
* **Mapeamento em Intervalos** – também chamado Discretização, a representação em intervalos pode ser obtida a partir de métodos que dividam o domínio de uma variável numérica em intervalos. Existem diversos métodos para ser fazer sito, como por exemplo, dividir em intervalos com comprimentos definidos pelos usuário, dividir em intervalos de comprimentos iguais ou até mesmo dividir em intervalos por meio de agrupamentos (clusters) (Passos & Goldschmidt, 2005).
  + - 1. **Codificação: Categórica – Numérica**
* **Representação Binária Padrão** – Nesta representação, cada valor categórico é associado a um valor de 1 até N e é representado por uma cadeia de dígitos binários. Por exemplo, se temos 5 possíveis valores, pode-se representa-los com cadeias binárias de comprimento 3, como mostrado na Tabela 3.1:

**Tabela 3.1** Representação Binária Padrão

|  |  |
| --- | --- |
| Valores originais | Representação binária padrão |
| Casado | 001 |
| Solteiro | 010 |
| Viúvo | 100 |
| Divorciado | 011 |
| Outro | 110 |

* **Representação Binária 1-de-N** – Nessa representação, o código 1-N tem um comprimento igual ao número de categorias discretas permitidas para a variável, onde cada elemento na cadeia de bits é 0, exceto para um único elemento: aquele que representa o valor da categoria em questão, como ilustrado na Tabela 3.2.

**Tabela 3.2** Representação Binária 1-de-N

|  |  |
| --- | --- |
| Valores originais | Representação binária 1-de-N |
| Casado | 00001 |
| Solteiro | 00010 |
| Viúvo | 00100 |
| Divorciado | 01000 |
| Outro | 10000 |

* **Representação Binária por Temperatura** – Essa representação é utilizada mais frequentemente quando os valores discretos estão relacionados de algum modo. Por exemplo, uma variável discreta pode ter um dos seguintes valores: fraco, regular, bom e ótimo. Nesse caso, existe uma graduação entre os valores. Onde o valor ótimo é o melhor valor e fraco o pior valor. Dessa forma, a diferença entre os conceitos fraco e ótimo deve ser a maior possível entre os valores. Pode-se considerar que cada valor corresponde a um acréscimo de bit igual a 1 na representação, conforme mostra a tabela abaixo, como mostra a Tabela 3.3.

**Tabela 3.3** Representação Binária por Temperatura

|  |  |
| --- | --- |
| Valores originais | Representação binária por temperatura |
| Fraco | 0001 |
| Regular | 0011 |
| Bom | 0111 |
| Ótimo | 1111 |

De acordo com Passos & Goldschmidt (2005), uma medida de distância normalmente utilizada conjuntamente a esta representação é a distância de ***Hamming*** (**DH**), também conhecida como *City-Block*. Essa distância expressa a diferença entre duas cadeias de bits, adicionando uma unidade sempre que bits de mesma posição possuem valores distintos. A tabela a seguir mostra a distância de *Hamming* entre os conceitos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DH | Fraco | Regular | Bom | Ótimo |
| Fraco | 0 | 1 | 2 | 3 |
| Regular | 1 | 0 | 1 | 2 |
| Bom | 2 | 1 | 0 | 1 |
| Ótimo | 3 | 2 | 1 | 0 |

* + 1. **Enriquecimento dos dados**

De acordo com Passos & Goldschmidt (2005), a função de enriquecimento consiste em conseguir de alguma forma mais informações que possa ser empregada aos registros existentes, enriquecendo os dados, para que esses forneçam mais informações para o processo de descoberta de conhecimento. A seguir será comentada algumas das operações mais usualmente utilizadas no processo de enriquecimento das bases de dados.

* **Pesquisas** – essa operação inclui todas as iniciativas de enriquecimento que envolve a captação de novas informações junto às fontes originais. Normalmente requerem a inclusão de novos atributos ou mesmo de novas tabelas. Busca-se, no enriquecimento, agregar novas informações.
* **Consultas a Bases de Dados Externas** – o processo de enriquecimento pode ser realizado mediante a incorporação de informações fornecidas por outros sistemas. É muito comum a importação de informações advindas de outras bases de dados.
  + 1. **Partição do Conjunto de Dados**

A Mineração de Dados, conforme será detalhada na próxima seção, é a etapa do processo de KDD responsável pela abstração de modelos de conhecimentos a partir dos dados existentes (Passos & Goldschmidt, 2005). A qualidade desses modelos precisam ser avaliada. A avaliação de um modelo de conhecimento requer a confrontação deste com dados visando à mensuração de algumas medidas que expressem a qualidade deste modelo. Para que essa avaliação seja isenta, os dados utilizados na construção do modelo não devem ser os mesmos utilizados na avaliação desse modelo. Portanto, pelo menos dois conjuntos de dados devem ser utilizados no processo de KDD: um conjunto de treinamento e um conjunto de teste. Como, em geral, tem-se um conjunto de dados, a operação de Partição do conjunto de dados em treinamento e teste assume grande importância. A seguir estão indicados alguns métodos utilizados na partição do conjunto de dados, com vistas à posterior avaliação dos modelos de conhecimento gerados.

* **Holdout** – esse método divide aleatoriamente os registros em uma percentagem fixa p para treinamento e (1 – p) para teste, considerando normalmente p > ½. Segundo Rezende (2002), embora não existam fundamentação teórica sobre esta percentagem, valores tipicamente utilizados são: p = 2/3 e (1 – p) = 1/3. Utilizado quando deseja-se produzir um único modelo de conhecimento.
* **Validação Cruzada com K Conjuntos** (*K-Fold CrossValidation*) – Esse método consiste em dividir aleatoriamente o conjunto de dados com N elementos em K subconjuntos disjuntos (folds), com aproximadamente o mesmo número de elementos (N/K). Neste processo, cada um dos K subconjuntos é utilizado como conjunto de teste e os (K – 1) demais subconjuntos são utilizados como treinamento. Assim, o processo é repetido K vezes, sendo gerados e avaliados K modelos de conhecimento.
* **Validação Cruzada com K Conjunto Estratificada** (*Stratifield K-Fold CrossValidation*) – Aplicável em problemas de classificação, este método é similar à Validação Cruzada com K Conjuntos, sendo que ao gerar os subconjuntos mutuamente exclusivos, a proporção de exemplos em cada uma das classes é considerada durante a amostragem. Isso significa que, se o conjunto de dados original possui duas classes com distribuição de 20% e 80%, cada subconjunto também deverá conter aproximadamente esta mesma proporção de classes.
* **Leave-One-Out** – Esse é método é caso particular da Validação Cruzada com K conjuntos, em que cada um dos K subconjuntos possui um único registro. É computacionalmente dispendioso e frequentemente usado em pequenas amostras.
* **Bootstrap** – O conjunto de treinamento é gerado a partir de N sorteios aleatórios e com reposição a partir do conjunto de dados original (contendo N registros). O conjunto de teste é composto pelos registros do conjunto de dados original não sorteados para o conjunto de treinamento. Esse método consiste em gerar os conjuntos, abstrair e avaliar o modelo de conhecimento um número repetido de vezes, a fim de estimar uma média de desempenho do algoritmo de Mineração de dados.
  + 1. **Integração dos Dados**

De acordo com Han; Kamber & Pei (2011); Witten (2005), os dados para o processo de KDD, são selecionados de diversas fontes distintas, como já foi mencionado anteriormente. Então nesta fase, faz-se a integração de várias fontes de dados, mantendo a consistência e a coerência dos dados integrados.

* 1. **Mineração de Dados**

**3.4.1 Definição**

Existem, segundo Han; Kamber & Pei (2011), várias definições para Data Mining, por ser um assunto verdadeiramente interdisciplinar, ela envolve um extenso campo de pesquisa, que associa técnicas e conceitos de diversas áreas como sistemas de banco de dados, sistemas baseados em conhecimento, inteligência artificial, aprendizado de máquina, aquisição do conhecimento, estatística, bancos de dados espaciais e visualização de dados. As várias tarefas desenvolvidas em Mineração de Dados têm como objetivos primário a predição e/ou a descrição. A predição usa atributos para predizer valores futuros de uma ou mais variáveis (atributos) de interesse. A descrição contempla o que foi descoberto nos dados de vista da interpretação humana.

Durante a etapa de Mineração de Dados é realizada a busca efetiva por conhecimentos úteis no contexto da aplicação de KDD. É por tanto, na Mineração de Dados onde são definidas as técnicas e os algoritmos a serem utilizados no problema em questão. A escolha da técnica depende, muitas vezes, do tipo de tarefa de KDD a ser realizada. São muitas as tarefas de Mineração de dados, mas nesse trabalho será dados atenção especial as tarefas **Associação**, **Classificação**, **Árvores de Decisão** e **Agrupamento**, pois essas as pretendidas a serem usadas nessa pesquisa para desenvolvimento do Caso de Uso.

De acordo com Han; Kamber & Pei (2011); Passos & Goldschmidt (2005), a execução da etapa de Mineração de Dados (Data Mining) compreende a aplicação de algoritmos sobre os dados procurando abstrair conhecimento. Estes algoritmos são fundamentados em técnicas que procuram, segundo determinados paradigmas, explorar os dados de forma a produzir modelos de conhecimento.

Mas antes de apresentar as tarefas de KDD, será apresentado um conceito muito importante no processo de Mineração de dados, que é a noção de **similaridade**. Suponha que o conjunto de dados possa ser interpretado como um conjunto de pontos em um espaço k-dimensional, o conceito de similaridade entre dois pontos pode ser traduzido como a distância entre estes pontos. Quanto maior a similaridade, menor a distância entre os pontos (Passos & Goldschmidt, 2005).

O conceito de distância é formalizado como uma função D : E x E R (a cada par de pontos associa um valor real ) que atende às seguintes restrições:

* D(x,x) = 0
* D(x,y) = D(y,x)
* D(x,y) ≤ D(x,z) + D(z,y)

Alguns exemplos de distância mais geralmente usados:

* **Distância Euclidiana**:

**(3.2)**

* **Distância de Hamming (City-Block)**:

**(3.3)**

* **Distância de Minkowski**:

**(3.4)**

Diversos algoritmos de Mineração de Dados, com destaque para o K-NN (K-Nearest Neighbors – K vizinhos mais próximos) utilizam o conceito de distância entre os dados no conjunto de dados.

* + 1. **Tarefas de Mineração de Dados**

As metas primárias que podem ser alcançadas através da Mineração de Dados, são as seguintes (Fayyad, et al, 1996):

* **Previsão** - Nesse caso busca-se um modelo de conhecimento que permita, a partir de um histórico de casos anteriores, prever os valores de determinados atributos em novas situações.
* **Descrição** - Nesse caso busca-se por um modelo que descreva, de forma compreensível pelo homem, o conhecimento existente em um conjunto de dados.

Segundo Fayyad, et al (1996), a mineração preditiva consiste na generalização de exemplos ou experiências passadas com respostas conhecidas ou regras de negócios estabelecidas por especialistas. Enquanto, a mineração descritiva, consiste na identificação de comportamentos intrínsecos do conjunto de dados, sendo que estes dados não possuem uma classe especifica.

* + - 1. **Descoberta de Associações**

De uma forma geral, a tarefa clássica de busca por regras de associação, também denominada regras associativas, foi introduzida em (Agrawal et al., 1993). Intuitivamente essa tarefa consiste em encontrar conjunto de registros de itens que ocorram simultaneamente e de forma frequente em um banco de dados.

Uma **regra de associação** é uma implicação da forma X => Y, onde X = {x1, x2, ..., xn} e Y = {y1, y2, ..., ym} são conjunto de itens, com xi e yj sendo itens distintos para todo i e todo j. Essa associação estabelece que se um cliente comprar X, ele também estará propenso a comprar Y. Para que uma regra de associação seja de interesse para a mineração de dados, a regra precisa satisfazer algumas medidas. Duas medidas de interesse comuns fornecem suporte e confiança. Segue a seguir, as formulas para cálculo do fator de suporte e de confiança (Agrawal, et al., 1996; Elmasri, 2005).

**(3.5)**

**, onde N é o número total de tuplas.**

**(3.6)**

O fator de **suporte** pode ser descrito como a probabilidade de uma transação qualquer satisfazer tanto X como Y, ao passo que o fator de **confiança** é a probabilidade de que uma transação satisfaça Y, dado que ela satisfaça X. a tarefa de descobrir regras de associação consiste em extrair do banco de dados todas as regras com “Fs” e “Fc” maiores ou iguais a um “Fs” e “Fc” especificado pelo analista de dados (Agrawal, et al., 1996).

Uma associação é considerada frequente se o número de vezes em que a união de conjuntos de itens (x ocorrer em relação ao número total de transações do banco de dados for superior a uma frequência mínima (denominada suporte mínimo) que é estabelecida em cada aplicação. Busca-se por meio do suporte, identificar que associações surgem em uma quantidade expressiva a ponto de ser destacada das demais existentes (Agrawal, et al., 1996).

Ainda segundo Agrawal et al. (1996), uma associação é considerada válida se o número de vezes em que X Y ocorrer em relação ao número de vezes que X ocorrer for superior a um valor denominado confiança mínima, que também é estabelecida em cada aplicação. A medida de confiança procura expressar a qualidade de uma regra, indicando o quanto a ocorrência do antecedente da regra pode assegurar a ocorrência do consequente desta regra.

Desta forma, a tarefa de Descoberta de Associações ou Descoberta de Regras de Associações, pode ser definida formalmente como a busca por **regras de associação frequentes e válidas** em um banco de dados, a partir da especificação dos parâmetros de suporte e confiança mínima (Agrawal et al., 1996).

Os valores dos parâmetros de suporte e confiança mínima devem ser especificados pelo especialista em KDD em conjunto com o especialista no domínio da aplicação (Agrawal et al., 1996).

Segundo Agrawal et al. (1996), existem diversos algoritmos desenvolvidos especificamente para aplicação na tarefa de descoberta de associações, dentre eles: Apriori, DHP (Direct Hashing and Pruning), Parttion, DIC (Dynamic ItemSet Counting), Eclat, MaxEclat, Clique, MaxClique, etc. além dos mais, existem versões destes algoritmos para sistemas distribuídos.

* + - 1. **Classificação**

Essa tarefa pode ser compreendida como a busca por uma função que permita associar corretamente cada entrada **X**i de um conjuntos de dados de entrada, a um único rótulo **Y**i, denominado classe. Uma vez identificada, essa função pode ser aplicada a novas entradas de forma a prever a classe em que tais dados se enquadram. Dados podem ser associados a classes ou a conceitos através de um processo de discriminação ou caracterização (Han; Kamber & Pei, 2011).

De acordo com Han; Kamber & Pei (2011), a tarefa de classificação é um processo de dois passos. No primeiro passo, constrói-se um modelo com base nos dados. No segundo passo, determina se a acurácia desse modelo é aceitável, se assim for, usa-se esse modelo para classificar novos dados.

Para formalizar a tarefa de classificação, consideremos um par ordenado da forma **(x, ƒ(x))**, onde **x** é um vetor de entradas n-dimensional e **ƒ(x)** a saída de uma função **ƒ**, desconhecida, aplicada a **x**. A tarefa de inferência indutiva consiste em, dada uma coleção de exemplos de **ƒ*,*** obter uma função ***h*** que se aproxime de **ƒ**. A função ***h*** é chamada de hipótese ou modelo de ***ƒ*** (Passos, Goldschmidt, 2005).

Nos casos em que a imagem de **ƒ** é formada por rótulos de classes, a tarefa de inferência indutiva é denominada classificação e toda hipótese ***h*** chamada de classificador. A identificação da função ***h*** consiste em um processo de busca no espaço de hipóteses **H**, pela função que mais se aproxime da função original **ƒ**. Esse processo é denominado **aprendizado** (Russel e Norving, 1955, citado por Passos & Goldschmidth, 2005). Todo algoritmo que possa ser utilizado nesse processo é denominado de **algoritmo de aprendizado**. O conjunto de todas as hipótese que podem ser obtidas a partir de um algoritmo de aprendizado **L** é representado por **H**L. Cada hipótese pertencente ao **H**L é representado por ***h***L.

A acurácia de um classificador em um dado conjunto de teste é a percentagem do conjunto de tuplas de testes que são corretamente classificadas pelo classificador Han; Kamber & Pei (2011), ou seja, a acurácia da hipótese ***h*** em mapear corretamente cada vetor de entradas x em **ƒ(x)**. O conjunto de pares **(x, ƒ(x))** utilizado na identificação da função ***h*** é denominada conjunto de treinamento. Por outro lado, o conjunto de pares **(x, ƒ(x))** utilizados para avaliar a acurácia de ***h*** é denominado conjunto de testes. Dessa forma, o algoritmo **L** pode ser interpretado como uma função tal que:

, onde T é o espaço composto por todos os conjuntos de treinamento possíveis para **L**.

Segundo Utgolff (1996), cada algoritmo possui *bias* indutivo que direciona o processo de construção dos classificadores. O *bias* indutivo de um algoritmo pode ser definido como o conjunto de fatores que coletivamente influenciam na seleção de hipótese.

Em termos práticos, o *bias* de um algoritmo de aprendizado **L** afeta o processo de aprendizado de duas formas: restringem o tamanho de espaço de hipóteses **HL**, e impõem uma ordem de preferência sobres as hipóteses em **HL** (Bensusan, 1999, citado por Utgolff, 1996).

Conforme mencionado anteriormente, uma medida de desempenho de um classificador comumente utilizada é a acurácia (Acc(*h*)), também conhecida como precisão do classificador.

**(3.7)**

Acc(*h*) = 1 – Err(*h*), onde:

Err(h) é denominada taxa de erro ou taxa de classificação incorreta:

**(3.8)**

Onde:

O operador ||E|| retorna 1 se a expressão E for verdadeira e 0, caso contrário;

**n** é o número de exemplos (registros da base de dados, por exemplo);

**yi** é a classe real associada ai i-ésimo exemplo;

***h*(i)** é a classe indicada pelo classificador para o i-ésimo exemplo.

O modelo derivado pode ser apresentado de várias formas, tais como regras de classificação (***IF-THEN***), árvores de decisão, fórmulas matemáticas, ou redes neurais. Existem muitos outros métodos de construção de modelos de classificação, tais como classificação ***naïve Bayesian***, máquina de vetor de suporte (***support vector machines***), e classificação do vizinho mais próximo (***k-nearest neighbor***), Backpropagation, classificação usando Padrões frequentes, etc (Han; Kamber & Pei, 2011).

**3.4.2.3 Agrupamento (Clustering)**

Técnicas de agrupamento e classificação objetivam realizar uma separação ótima entre objetos de uma coleção, permitindo a descoberta de novos padrões, previamente desconhecidos. O resultado da segmentação, independentemente da ferramenta utilizada, pode ser interpretado eficientemente por um especialista na área de origem dos dados sob análise. A facilidade de visualização resultante do agrupamento, favorece a análise (Han; Kamber & Pei, 2011).

A tarefa de agrupamento, é usada para segmentar os dados de entrada em subconjuntos ou clusters, de tal forma que elementos de um cluster compartilhem um conjunto de propriedades comuns que os distingam dos elementos de outros clusters. O objetivo desta tarefa é maximizar similaridades intracluster e minimizar similaridades intercluster. Diferente da classificação que tem rótulos predefinidos, o agrupamento precisa automaticamente identificar os rótulos. Por essa razão, o agrupamento também é chamada de indução não supervisionada (Passos & Goldschmidt, 2005).

Segundo Han; Kamber & Pei (2011), em geral, a classe não é representada nos dados de treinamento, simplesmente porque elas não são conhecidas no começo. Entretanto, os agrupamentos (*clusters*) podem ser usados para gerar tais rótulos. Os objetos são agrupados baseados no princípio de máxima similaridade intra-classe e mínima similaridade inter-classe. Isto é, o agrupamento de objetos são formados de modo que os objetos dentro do agrupamento tenham alta similaridade em comparação com um outro objeto, mas são muitos diferentes dos objetos em outro agrupamento. Cada agrupamento formado pode ser visto como uma classe de objetos, da qual pode derivar regras. Agrupamento também facilita a formação de taxonomia (**taxonomy formation**), isto é, a organização das observações em uma hierarquia de classes que agrupem eventos similares juntos.

Formalmente para o processo de agrupamento, supõe-se a existência de **n** pontos de dados **x1**, **x2**, ..., **xn** tais que cada ponto pertença a um espaço d dimensional **Rd**. A tarefa de agrupamento desses pontos de dados, separando-os em **k** clusters consiste em encontrar **k** pontos **mj** em **Rd** de tal forma que a expressão (Passos & Goldschmidt, 2005):

**(3.13)**

Seja minimizada, onde **d2(xi, mj)** denota uma distância entre **xi** e **mj**. Os pontos **mj** são denominados centroides ou médias dos clusters.

De forma resumida, o problema descrito acima consiste em encontrar k centroides de clusters de tal maneira que a distância entre cada ponto de dado e o centroide do cluster mais próximo seja minimizada (Passos & Goldschmidt, 2005).

De acordo com (Passos & Goldschmidt, 2005), para que os algoritmos de agrupamento possam efetuar sua tarefa é necessário que sejam utilizadas estruturas de dados capazes de armazenar os objetos a serem processados. Algoritmos de agrupamento que trabalham com dados armazenados na memória principal, normalmente, utilizam uma das seguintes estruturas de dados no seu processamento:

* **Matrizes de dados** – As linhas representam cada um dos objetos a serem agrupados e as colunas, os atributos ou características de cada objeto. Considerando **n** objetos, cada qual com p atributos, obtém-se uma matriz **n x p** como ilustrado abaixo:
* **Matriz de similaridade** – Cada elemento da matriz representa a distância entre pares de objetos. Visto que a distância entre o objeto **i** e o objeto **j** é igual a distância entre o objeto **j** e o objeto **i**, não é necessário armazenar as distâncias entre os objetos. Considerando **n** objetos a serem agrupados, obtém-se uma matriz quadrada de tamanho **n x n** como a ilustrada a seguir:

O ponto **d(i,j)** representa a distância ou similaridade entre o objeto **i** e o **j**. Como as medidas de similaridade expressam o conceito de distância, estas são sempre números positivos. Quanto mais próximo de zero for **d(i,j)**, mais similares serão os objetos.

Segundo Han; Kamber & Pei (2011), quando um algoritmo que trabalha com matrizes de similaridade recebe um matriz de dados, ele primeiro a transforma em uma matriz de similaridade antes de iniciar o processo de agrupamento.

Agrupamento ou Clustering é também chamado de ***data segmentation*** em algumas aplicações, porque segmenta grande conjunto de dados em grupos de acordo com suas similaridades. O Agrupamento também pode ser usado em detecção de desvio (outlier detection), onde desvios (valores “distantes” de qualquer cluster) pode ser mais interessante do que os casos comuns. Aplicações de detecção de desvio inclui a detecção de fraudes em cartões de crédito e o monitoramento de atividades criminais em comercio eletrônico (Han; Kamber & Pei, 2011).

Segundo Han; Kamber & Pei (2011), o agrupamento é um campo de pesquisa desafiador e que, para sua utilização no Data Mining são necessários alguns requisitos, tais como:

* **Escalabilidade**: muitos algoritmos de agrupamento trabalham bem com pequenos data sets, contendo pouco mais de 100 objetos de dados; embora, um grande banco de dados, possa conter milhões ou bilhões de objetos, particularmente no cenário da Web. Fazer agrupamento em uma amostra de um grande *DataSet* pode levar a resultado tendencioso. Portanto, os algoritmos de agrupamento necessitam de alta escalabilidade.
* **Habilidade para lidar com diferentes tipos de atributos**: muitos algoritmos são projetados para cluster de dado numérico (baseado em intervalo). Embora, as aplicações possam requerer outros tipos de dados, tais como binário, nominal (categórico), e dados ordinal, ou uma mistura desses tipos. Recentemente, mais e mais aplicações necessitam de técnicas de agrupamento para tipos de dados complexos tais como gráficos, imagens, sequencias e documentos.
* **Descobrir clusters de forma arbitrária:** muitos algoritmos de agrupamento determinam os clusters baseados em medidas de distância Euclidiana ou Manhattan. Algoritmos baseados em tais medidas de distância tendem a encontrar clusters esféricos com tamanho e densidade similares. É importante desenvolver algoritmos que possa detectar clusters de forma arbitrária.
* **Requisitos para determinar os parâmetros de entrada para o domínio do conhecimento**: muitos algoritmos de agrupamento exigem que os usuários forneçam o conhecimento na forma de parâmetros de entrada, como o número desejado de clusters. Consequentemente, o resultado pode ser que os clusters sejam sensíveis aos parâmetros. Parâmetros são difíceis de serem determinados.
* **Habilidade para lidar com dados ruidoso**: muitos conjuntos de dados do mundo real contêm dados errados, ausência de dados, os dados desconhecidos. Como por exemplo, um sensor de leitura de dados, onde em algumas leituras pode ser imprecisas.
* **Clustering incremental e sensitivo a ordem de entrada dos dados**: em muitas aplicações, pode haver atualizações a qualquer momento. Alguns algoritmos de agrupamento não incorporam a capacidade de atualização incremental de clusters, tendo de recalcular tudo do zero. Algoritmos de agrupamento também podem ser sensíveis a ordem de entrada de dados. Isto é, dado um conjunto de objetos, o algoritmo de agrupamento pode retornar diferentes clusters dependendo da ordem em que os objetos são apresentados. Portanto, são necessários algoritmos que sejam incrementais e sensíveis a ordem de entrada dos dados.
* **Capacidade de clustering de alta dimensionalidade:** um conjunto de dados pode conter muitas dimensões ou atributos. Quando agrupamos documentos, por exemplo, cada palavra-chave pode ser considerada uma dimensão, e frequentemente existem milhares de palavras-chave. Muitos algoritmos de agrupamento são bons em manipular conjunto de dados de poucas dimensões, tais como conjuntos envolvendo duas ou três dimensões.
* **Clustering com base em restrições:** aplicações do mundo real pode necessitar de realizar cluster em vários tipos de restrições. Suponha que em seu trabalho, você necessite determinar a localização para novas *automatic teller machine* (**ATM**) em uma cidade. Para decidir sobre isso, você pode agrupar considerando restrições como grupos familiares, redes rodoviárias e número de pessoa por cluster.
* **Interpretabilidade e usabilidade:**  os usuários esperam que o resultado do agrupamento seja interpretável, compreensível, e usável. Isto é, o agrupamento, pode necessitar ser atrelado uma interpretação e semântica especifica e aplicação. Isto é importante para estudar como uma meta da aplicação pode influenciar a seleção das características de agrupamento e métodos de agrupamento.

De acordo com Han; Kamber & Pei (2011), existem várias técnicas e métodos de agrupamento. Entre os principais algoritmos de agrupamento podem ser citados: K-Means, Fuzzy K-Means, K-Modes e K-Medoid. Entretanto, os métodos de agrupamento mais conhecidos e utilizados são os métodos particionamento, os métodos hierárquicos, os métodos baseados em densidade e os métodos baseados em Grade (*Grid*). Na sessão seguinte, serão discutidos alguns desses métodos.

**3.4.3 Métodos de Mineração de Dados**

Nesta sessão trata-se dos métodos de Mineração de Dados, mais é importante destacar que muitos deles são utilizados em outros tipos aplicações.

De acordo com Passos & Goldschmidt (2005), cada método de Mineração de Dados requer diferentes necessidades de pré-processamento. Tais necessidades variam em função do aspecto extensional da base de dados em que o método será utilizado. Em decorrência da diversidade de métodos de pré-processamento de dados, são muitas as alternativas possíveis de combinações entre métodos. A escolha dentre estas alternativas pode influenciar na qualidade do resultado do processo de KDD (Morik, 2000; Engels, 1996; Engels et al., 1997).

De acordo com Passos & Goldschmidt (2005), os métodos de KDD, sendo os métodos de Mineração de Dados um caso particular, podem ser considerados operadores definidos a partir de precondições e efeitos. Uma precondição de um método de KDD é um predicado que estabelece um requisito que deve ser cumprido antes da execução do método. Um efeito de um método de KDD também é um predicado que descreve uma situação gerada após a aplicação do método. Um plano de ação de KDD válido é toda sequência de métodos de KDD onde as precondições para execução de cada um dos métodos da sequência sejam devidamente atendidas.

**3.4.3.1 Métodos Baseados em Redes Neurais**

Segundo Han; Kember & Pei (2011); Passos & Goldschmidt (2005); Rezende (2003), diversos modelos de Redes Neurais podem ser utilizados na implementação de métodos de Mineração de Dados. Classificação, Regressão, Previsão de Séries Temporais e Agrupamento (Clusters) são exemplos de tarefas de Mineração de Dados que podem ser implementadas por métodos de Redes Neurais. Além do mais, alguns modelos de Redes Neurais podem ser aplicados em mais de um tipo de tarefa de Mineração de dados.

A topologia da rede neural varia em função do problema e da representação adotada para os dados. De uma forma geral, em aplicações de Mineração de Dados, a camada de entrada do modelo neural recebe os dados pré-processados de cada registro de um banco de dados. A rede processa esses dados produzindo uma saída cuja natureza também varia em função da aplicação (Passos & Goldschmidt, 2005).

Ainda de acordo com Passos & Goldschmidt (2005), em redes neurais com aprendizado supervisionado, a entrada corresponde aos atributos preditivos enquanto a saída do modelo corresponde ao atributo objetivo do problema. Então, o algoritmo de aprendizado pode estimar o erro, ou distância, entre a saída produzida pela rede e a saída desejada. Dessa forma, o algoritmo, em função do erro calculado, ajusta os pesos das conexões da rede a fim de tornar a saída real tão próxima quanto seja possível da saída desejada. Estes tipos de modelos neurais são muito úteis em geral para reconhecimento de padrões e para tarefas de Mineração de Dados que envolvam predição.

As Redes Neurais com aprendizado não supervisionado são adequados para tarefas que envolvam descrição de dados, como por exemplo, a tarefa de Agrupamento. Nestes casos, não há saída desejada (Passos & Goldschmidt, 2005).

Alguns algoritmos de aprendizado e indicadas a tarefa de Mineração de dados, são:

* **Back-Propagation** – O algoritmo Back-Propagation, também conhecido como algoritmo de retro-propagação do erro, é um algoritmo de aprendizado supervisionado, cuja aplicação é adequada a tarefa de Mineração de Dados tais como Classificação, Regressão ou Previsão. Esse algoritmo tem como objetivo minimizar a função de erro entre a saída gerada pela rede neural e a saída real desejada, utilizando o método do gradiente descendente.
* **Kohonen** – O mapa Kohonen pertence à classe das Redes Neurais Auto organizáveis. Em uma Rede Neural Auto organizável o treinamento é não supervisionado, geralmente baseado em uma forma de competição entre os elementos processados. Entre as principais aplicações das Redes Auto organizáveis estão:
  + **Tarefas de Clusterização –** Tarefa na qual os dados de entrada devem ser agrupados em conjuntos que agregam padrões semelhantes.
  + **Detecção de Regualaridades** – Modelo em que o sistema deve extrair as características relevantes dos padrões de entrada.

Segundo Passos & Goldschmidt (2005), o método de aprendizagem mais comum nas Redes Neurais Auto Organizáveis é denominado aprendizado por competição (*Competitive Learning*). O qual, consiste em uma forma de aprendizado que divide o conjunto de padrões de entrada em grupos inerentes aos dados. Para isto, esse método considera, em sua abordagem mais simples, que os neurônios de saída da rede competem entre si, resultando em apenas um neurônio vencedor.

**3.4.3.2 Métodos Estatísticos**

De acordo com Passos & Goldschmidt (2005); Han; Kamber & Pei (2011), diversos algoritmos de Mineração de Dados são fundamentados em princípios e teorias da Estatística. A seguir serão apresentados alguns deles:

* **Classificador Bayeasiano Ingênuo** – O classificador Bayeasiano Ingênuo baseia-se no Teorema de Bayes, estando relacionado ao cálculo de probabilidades condicionais. É aplicável, conforme o próprio nome sugere, em tarefas de classificação.

Formalmente, sejam **X(A1, A2, ..., An, C)** um conjunto de dados, **C1**, **C2**, ..., **Cn**, as classes do problema (valores possíveis do atributo **C**) e R um novo registro que deve ser classificado. Sejam ainda **a1**, **a2**, ..., **an** na os valores que **R** assume em **X**. O calculador Bayesiano possui dois passos:

1. Calcular a probabilidade **P(C=Ci/R)**, **i**=1,2, ..., k.
2. Indicar com saída do algoritmo a classe **Cj** tal que **P(C=Cj/R)** seja máxima.

* **K-Means** – O algoritmo k-means é um método popular da tarefa de agrupamento. Toma-se, randomicamente, k pontos de dados (dados numéricos) como sendo os centroides (elementos centrais) dos clusters. Em seguida, cada ponto (ou registro da base de dados) é atribuído ao cluster cuja distância deste ponto em relação ao centroide de cada cluster é a menor dentre todas as distâncias calculadas. Um novo centroide para cada cluster é computado pela média dos pontos do cluster, caracterizando a configuração dos clusters para a iteração seguinte. O processo termina quando os centroides dos clusters param de se modificar, ou após um número limitado de iterações que tenham sido especificado pelo usuário (Han, Kamber & Pei, 2011; Passos & Goldschmidt, 2005).

Formalmente, segundo Han; Kamber & Pei (2011), suponha um conjunto de dados **D**, contendo **n** objetos em um espaço Euclidiano. O método de particionamento, distribui os objetos em **D** entre k clusters, **C1**, ..., **Ck**, isto é, e para (. A execução do algoritmo ***k-means*** toma um parâmetro de entrada k, e divide um conjunto de n objetos em k clusters tal que a similaridade intracluster resultante seja alta, mas a similaridade intercluster seja baixa. A similaridade em um cluster é medida em respeito ao valor médio dos objetos neste cluster (centro de gravidade do cluster). O método *k-means* é inicializado com os centros (médias) colocados em posições aleatórias. A busca pelo centro comum se faz de forma iterativa. Após essa inicialização, os objetos restantes são agrupados conforme a distância em que se encontram das médias. A diferença entre um objeto , e **ci,** a representatividade do cluster, é medida por *dist* (**p**, **c**), onde *dist* (**x**, **y**) é a distância Euclidiana entre dois pontos **x** e **y**. A qualidade do cluster Ci pode ser medido pela variação intra-cluster, a qual é definida pela soma

**(3.14)**

Onde **E** é a soma do quadrado dos erros para todos os objetos no conjunto de dados; **p** é o ponto no espaço Euclidiano representando um dado objeto; e **ci** é o centroide do cluster (ambos **p** e **ci** são multidimensionais). Em outras palavras, para cada objeto em cada cluster, a distância do objeto ao centro do cluster é quadrada, e as distâncias são somadas (Han; Kamber & Pei, 2011).

* **K-Modes** – O algoritmo k-modes é uma variação do método k-means, só que utilizado para agrupamento de dados categóricos (nominais). Em geral, no lugar do cálculo da média, calcula-se a moda dos objetos, usando medidas de similaridades para tratar objetos categóricos, e usando métodos baseados em frequência para atualizar as modas dos clusters.
* **K-Prototypes** – O método *k-prototypes* é a integração dos métodos *k-means* e *k-modes*. Esse método pode ser aplicado a bases de dados que contenham tanto atributos numéricos quanto atributos categóricos.
* **K-Medoids** - O algoritmo k-medoids baseia-se, primeiramente, em encontrar os medoids (objetos mais centralmente localizado em um cluster). Os objetos restantes são então agrupados com o medoid ao qual ele é mais similar. Há então uma troca iterativa, de um medoid por um não medoid, visando à melhoria do agrupamento. O método então, é realizado baseado no princípio de minimizar a soma das dissimilaridades entre cada objeto **p** e seu correspondente objeto representativo. Isto é, um critério absoluto de erro (***absolute-error criterion***), definido como

**(3.15)**

Onde **E** é a soma dos erros absolutos para todos objetos **p** no conjunto de dados; e **Oi** é o objeto representativo de **Ci**. Isto é o básico para o método ***k-medoids***, os quais os grupos de **n** objetos em **k** clusters, minimizando o erro absoluto.

Os algoritmos **PAM**, **CLARA** e **CLARANS** são os principais algoritmos da classe *k-medoids*. O primeiro (**PAM**) é bem ineficiente, o último (**CLARANS**) é bem mais eficiente, mais ainda tem o problema da dificuldade de detectar clusters de formatos arbitrários.

**3.4.3.3 Método Específico - Apriori**

O *Apriori* é um algoritmo clássico de Mineração de Regras de Associação (Agrawal, 1993). Diversos algoritmos tais como GSP, DHP, Partition, DIC, etc., foram inspirados no funcionamento do *Apriori* e se baseiam no princípio da antimonotonicidade do suporte. Segundo este princípio, “Um *k-itemset* somente pode ser frequente se todos os seus (k-1)-*itemsets* forem frequentes”. Assim sendo, a combinação de *itemsets* para gerar um novo *itemset* somente ocorre quando estes são frequentes.

Segundo Passos & Goldschmidt (2005), tais algoritmos podem ser decompostos basicamente em duas etapas:

1. Encontrar todos os conjuntos de itens frequentes (que satisfaçam à condição suporte mínimo).
2. A partir do conjunto de itens frequentes, gerar as regras de associação (que satisfazem à condição de confiança mínima).

**3.4.3.4 Métodos Baseados em Indução de Árvores de Decisão**

Segundo Passos & Goldschmidt (2005), alguns dos principais métodos de Mineração de Dados são baseados na construção de árvores de decisão a partir da base de dados. Em geral a construção de uma árvore de decisão é realizada segundo alguma abordagem recursiva de particionamento da base de dados. Um exemplo clássico de método baseado na indução de árvores de decisão é o algoritmo C4.5.

* **C4.5** – O C4.5 procura abstrair árvores de decisão a partir de uma abordagem recursiva de particionamento das bases de dados. Utiliza, para tanto, conceitos e medidas da Teoria da Informação.

Uma árvore de decisão é um modelo de conhecimento em que cada nó interno da árvore representa uma decisão sobre um atributo que determina como os dados estão particionados pelos seus nós filhos. Inicialmente, a raiz da árvore contém toda a base de dados com exemplos misturados das várias classes. Um predicado, denominado ponto de separação, é escolhido como sendo a condição que melhor separa ou discrimina a classe. Tal predicado envolve exatamente um dos atributos do problema e divide a base de dados em dois ou mais conjuntos, que são associados cada um a um nó filho. Cada novo nó abrange, portanto, uma partição da base de dados que é recursivamente separada até que conjunto associado a cada nó folha consista inteiramente ou predominantemente de registros de uma mesma classe Passos & Goldschmidt (2005).

De acordo com Passos & Goldschmidt (2005), a maioria dos algoritmos baseados na abstração de árvores de decisão consiste de duas fases: Construção da Árvore de Decisão e Simplificação da Árvore de Decisão.

**3.4.3.5 Métodos Baseados em Lógica Nebulosa**

Diversos métodos de Mineração de dados foram adaptados de forma a incorporar a flexibilidade proporcionada pela Lógica Nebulosa. Entre eles pode-se citar as versões nebulosas K-Means e do C4.5. Nessas versões, os registros da base de dados podem pertencer a diversos clusters e classe simultaneamente, com diferentes graus de pertinência. Um exemplo de algoritmo nebuloso é o algoritmo de Wang-Mendel, concebido para aplicação na tarefa de Previsão de Séries Temporais (Passos & Goldschmidt, 2005).

**3.4.3.6 Métodos Hierárquicos**

Agrupamento hierárquico produz uma fusão sequencial aninhada das variáveis em estudo, baseado em algumas métricas de correlação ou semelhança, como a correlação de Pearson ou a distância Euclidiana. A fusão aninhada é representada por um “dendrograma". No nível mais baixo do dendrograma, cada variável é um membro de um cluster (grupo) individual. No primeiro passo, as variáveis mais semelhantes são fundidas em um cluster. Em seguida, as próximas duas variáveis mais similares são unidas em outro cluster. Em cada passo, os dois grupos mais semelhantes (incluindo conjuntos unitários) são unidos para formar um grande grupo. No nível mais alto do dendrograma, existe um conjunto contendo todas as variáveis (Han; Kamber & Pei, 2011).

A correlação de Pearson entre dois experimentos X e Y é dada por:

**(3.16)**

Onde N é o número de variáveis, Xavg é a média das variáveis no experimento X, e Yavg é a média das variáveis no experimento Y. A correlação de Pearson é uma medida de similaridade comumente usada entre dois vetores, portanto, um menos a correlação é usada como distância métrica.

A distância Euclidiana entre dois vetores é dada por:

**(3.17)**

Onde xi são as variáveis do experimento X e yi são as variáveis do experimento Y.

Além destas métricas, podemos citar ainda a métrica de Minkowski, que tem a distância Euclidiana como um caso particular. Outras versões populares da métrica de Minkowski são as distâncias de Manhattan e de Chebyshev. Outra métrica de distância cujo cálculo é extremamente simples é a distância Camberra que calcula a soma das diferenças fracionárias entre as coordenadas de pares de objetos (Linden,2009).

Uma outra técnica usual de agrupamento é o mapa auto-organizável (*self-organized map* - **SOM**) ou mapa de Kohonen, que é um algoritmo de agrupamento em redes neurais (Amaratunga; Cabreira, 2004). Este permite uma melhor visualização e identificação de agrupamentos similares como também da correlação entre as amostras.

O **SOM** transforma dados de alta dimensão em imagens de uma ou duas dimensões, onde o agrupamento pode ser identificado. É um algoritmo versátil e a visualização dos resultados pode ser feita de diferentes maneiras.

O K-means é também uma heurística de agrupamento não hierárquico que busca minimizar a distância dos elementos a um conjunto de forma iterativa. Podemos citar ainda os Grafos que são modelos matemáticos que representam relações entre objetos. Um grafo é um conjunto dado por G=(V, E), onde V é um conjunto finito de pontos, normalmente denominados de nós ou vértices e E é uma relação entre vértices, ou seja, um conjunto de pares em VxV (Linden, 2009).

De acordo com Han; Kamber & Pei (2011), seja qual for o método a ser usado, a necessidade principal é medir a distância entre dois clusters, onde cada cluster é geralmente, um conjunto de objetos.

Existem outros algoritmos hierárquicos tais como **BIRCH**: ***Multiphase Hierarchical Clustering*** usando recurso de clusters em árvores, o **Chameleon**: ***Multiphase Hierarchical Clustering*** usando modelagem dinâmica e ***Probabilistic Hierarchical Clustersing***. Esses algoritmos não serão discutidos nesse trabalho.

**3.4.3.7 Métodos Baseados em Densidade**

Os métodos baseados em densidade permitem descobrir grupos de formatos arbitrários. Estes métodos consideram grupos como sendo regiões densas de objetos no espaço de dados separados por regiões de baixa densidade, que geralmente representam ruídos, onde esses ruídos estão incluídos nos clusters (Han; Kamber & Pei, 2011). Pode-se citar como exemplos de algoritmos baseados em densidade **DBSCAN**, **OPTICS** e **DENCLUE**, que serão discutidos em seguida.

* **DBASCAN** - O algoritmo **DBSCAN** (*Density-based Spatial Clustering of Applications with Noise*) cria grupos de objetos em regiões de alta densidade e é capaz de descobrir grupos de formatos arbitrários em bases de dados espaciais e com ruído. Para o **DBSCAN** um cluster é definido como um conjunto máximo de pontos densamente conectados. Ele encontra clusters com formatos (***shape***) arbitrários em bancos de dados espaciais, contendo ruídos (*outliers*) (Han; Kamber & Pei, 2011).

O algoritmo DBSCAN distingue três tipos de objetos em um conjunto de dados:

1. **Pontos centrais:** são pontos que estão no interior de uma região densa, onde existe pelos menos ***η*** pontos no raio ***ε*** desse objeto. A cardinalidade desses pontos em relação ao parâmetro de ***ε*-vizinhança** deve ser de no mínimo ***η*** pontos;
2. **Pontos de borda (border point):** estão na fronteira de uma região densa, ou seja, sã pontos que estão na **ε-vizinhança** de algum ponto central, porém não são pontos centrais, pois a cardinalidade desses pontos em relação ao raio *ε* não excede *η*;
3. **Outliers (noise point):** esses pontos não são centrais e nem de borda e assim não são conectados por densidade a nenhum outro ponto, não pertencendo a nenhum *cluster*.

Caso um ponto for classificado como ***outlier***pelo algoritmo, posteriormente ele pode estar na *ε*-vizinhança de outro ponto não visitado ainda pelo DBSCAN. Sendo assim, essa classificação pode ser removida caso o objeto seja diretamente alcançável por densidade a partir de um ponto central ainda não visitado (Han; Kamber & Pei, 2011).

Segundo Han; Kamber & Pei, se for usado um índice espacial, a complexidade computacional do **DBSAC** é O(***n.* log *n***), onde ***n*** é o número de objetos de dados. Caso contrário, a complexidade é O(**n**2). Para configurações dos parâmetros apropriados pelo usuário, o algoritmo é efetivo para encontrar clusters de forma arbitrária.

* **OPTICS** - Embora o **DBSACN** possa encontrar clusters dado os parâmetros de entrada tais como **ε-vizinhança** (o raio máximo de uma vizinhança) e MintPts (o número de pontos requeridos na vizinhança de um *core objects*), isso transfere para os usuários a responsabilidade de selecionar os valores dos parâmetros que irão levar a descoberta de clusters aceitáveis. Isto é um problema, que também estar associados a muitos outros algoritmos de clusters. A configuração de tais parâmetros são usualmente definidos empiricamente, especialmente para parâmetros do mundo real, conjunto de dados de alta dimensionalidade. Além do mais, muitos algoritmos são sensíveis para esses valores de parâmetros: configurações levemente diferentes pode levar a muitos diferentes agrupamento de dados (Han; Kamber & Pei, 2011). Para superar a dificuldade em usar os parâmetros na análise de clusters, foi proposto um método chamado **OPTICS**.

O método **OPTICS** não produz explicitamente um conjunto de clusters de dados. Em vez disso, ele exibe ***cluster ordering***. Isto é, uma lista linear de todos os objetos que estão sobre análise e representa a estrutura de clusters baseado na densidade (*density-based clustering structure*) dos dados. Isto equivale a agrupamento obtida a partir de uma ampla e variada configuração de parâmetros. A ordenação de clusters pode ser usada para extrair informações básicas sobre os clusters (por exemplo, o centro do cluster, ou a forma arbitrária dos clusters), derivar a estrutura intrínseca do cluster, bem como proporcionar a visualização do cluster (Han; Kamber & Pei, 2011).

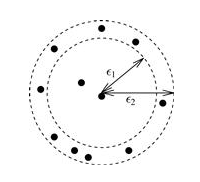
De acordo com Han; Kamber & pei (2011), para construir diferentes clusters simultaneamente, os objetos são processados em uma ordem especifica. Os objetos selecionados são aqueles que estão em uma distância menor do que **ε,** assim os clusters com alta densidade irão ser construídos primeiro.

A estrutura do algoritmo OPTICS é muito similar ao DBSCAN. Consequentemente, os dois algoritmos têm a mesma complexidade. A complexidade é O(*n.* log *n*) se é usado um índex espacial, e O(n2) caso contrário, onde **n** é o número de objetos.

* **DENCLUE** (*DENsity-based CLUstEring*) - É um método de agrupamento baseado num conjunto de funções de distribuição de densidade. Primeiro é dado uma estimativa da densidade, e então descreve o algoritmo DENCLUE.

Nos métodos DBSCAN e OPTICS, a densidade é calculada contando o número de objetos em uma vizinhança definida pelo parâmetro raio ε. Tal estimativa de densidade pode ser altamente sensível ao valor do raio. Por exemplo, na Figura 3.2, a densidade muda significativamente quando o raio aumenta por um pequeno valor.

**Figura 3.2** Estimando a densidade no DBSAN e OPTPICS. Aumentando o raio de ε1 para ε2, resulta em uma densidade muito maior. Fonte: (Han; Kamber & Pei, 2011).



Segundo Han; Kamber & Pei (2011), para superar este problema, pode ser usado estimativa de densidade de núcleo (***kernel density estimation***), que é uma abordagem da estatística de estimativa de densidade não-paramétrica. A ideia geral por trás da estimativa de densidade de núcleo é simples. Trata-se um objeto observado com um indicador de densidade de alta probabilidade na região circundante. A densidade probabilística de um ponto depende da distância a partir desse ponto para o objeto observado.

Formalmente, seja **x1**, ..., **xn** seja uma amostra independente e identicamente distribuída de uma variável aleatória ƒ. A função de densidade de núcleo de probabilidade estimativa é

**(3.20)**

Onde K() é o núcleo e h especifica o grau de suavização de um objeto no espaço de parâmetros. O núcleo pode ser considerado como uma função que modela a influência de um ponto dentro de uma amostra. Tecnicamente, um núcleo K() é uma função integrável não negativa que satisfaz dois requisitos: e K(-u) = K(u) para todos os valores de u. Uma função núcleo frequentemente utilizada é a função de núcleo Gaussiano com uma média 0 e uma variância de 1:

**(3.21)**

O DENCLUE usa um núcleo Gaussiano para estimar a densidade baseada no dado conjunto de objetos a serem agrupados. Um ponto **x**\* é chamado um ***density attractor*** se ele é máximo local da função de densidade estimada (Han; Kamber & Pei, 2011). Para evitar pontos máximo locais triviais, DENCLUE usa um limite de ruído ξ, e somente considera os atratores de densidade **x**\* tais que ) ≥ ξ. Estes não triviais atratores de densidade são centro de clusters.

Os objetos analisados são associados para o cluster através do atrator de densidade usando um procedimento de subida. Para um objeto x, o procedimento de subida inicia de x e é guiada pelo gradiente da função de estimativa de densidade. O atrator de densidade para x é calculado como

**(3.22)**

Onde é um parâmetro para controlar os passos da convergência, e

**(3.23)**

O procedimento de subida para k > 0 se (), e associar x para o atrator de densidade **x**\* = **x**k. Um objeto é um ruído se ele converge no procedimento de subida para um máximo local x\* com ) ≥ ξ.

De acordo com Han; Kamber & Pei (2011), um cluster é um conjunto de atrator de densidade X e um conjunto de objetos input C tais que cada objeto em C é associado para o atrator de densidade em X, e existe um caminho entre cada par de atrator de densidade onde a densidade é acima de ξ. Por usar múltiplos atratores de densidade conectados por caminhos, **DENCLUE** pode encontrar clusters de forma arbitrária.

Para Han; Kamber & Pei (2011), **DENCLUE** tem várias vantagens. Ele pode ser considerado com uma generalização de vários bem-conhecidos métodos de agrupamento, tais como *single-linkage* e DBSACN. Além disso, DENCLUE é invariante contra ruídos. A função de estimativa de densidade de núcleo pode efetivamente reduzir a influência de ruído por distribuir uniformemente o ruído nos dados de entrada (input).

**3.4.3.8 Métodos Baseados em Grade**

Os métodos baseados em grade (***Grid***) dividem os objetos em um número finito de células que formam uma estrutura de grade multidimensional na qual todas as operações de agrupamento são realizadas. A principal vantagem do método é que as operações independem do número de objetos da base de dados, e sim do número de células da estrutura da grade, o que melhora o desempenho dos algoritmos baseados nesta heurística. Como exemplos de algoritmo baseado em grade pode-se citar o algoritmo CLIQUE e STING (Han; Kamber & Pei, 2011).

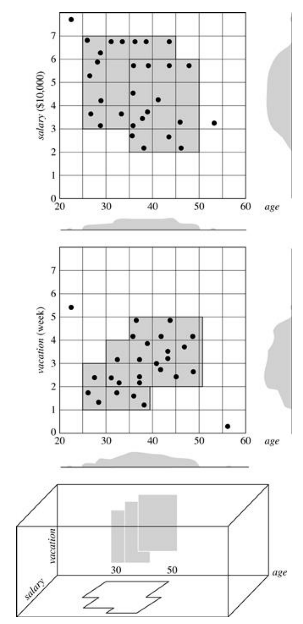
* **CLIQUE** - O algoritmo CLIQUE (Clustering in Quest), baseado em grade e em densidade, segmenta o conjunto de dados em subespaços (grade de células) para encontrar agrupamentos suficientemente densos. Cada grade organiza um conjunto de dados, separando os valores contínuos de cada atributo em um número de intervalos discretos. Por fim, cada objeto é atribuído a uma célula a qual seu intervalo contém o valor original do objeto. Os agrupamentos são formados a partir da junção de células densas adjacentes (Han; Kamber & Pei, 2011; Oliveira, 2007).

Segundo Han; Kamber & Pei (2011), o método CLIQUE particiona cada dimensão em intervalos não sobrepostos, dividindo, assim, todo espaço de incorporação dos objetos de dados em células.

A principal estratégia por trás do método CLIQUE é encontrar todas as células nos espaços unidimensionais correspondentes a cada atributo produzido pelas projeções em um dado número de partes iguais. São mantidas somente células que contenham número de elementos acima de um patamar (*MinPts*), denominados de células densa (Han; Kamber & Pei, 2011).

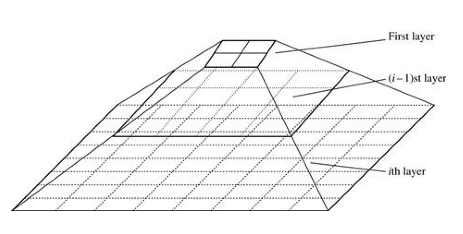
No caso geral, o algoritmo faz a identificação dos clusters em cada um dos atributos. Os atributos que não tem clusters, ou seja, que não atendem ao critério (MinPts), são eliminados. Os atributos restantes são combinados entre si, verificando-se a existência de clusters nesses espaços de mais alta dimensão. O processo continua até que não haja mais formação de clusters. Como resultado, os grupos são formados pela união de hiper-retângulos entre subespaços projetados atendendo fundamentalmente ao critério de densidade (MinPts). Considerando a Figura 3.2, onde o espaço de dado incorporado contém três dimensões: idade (*age*), salário (*salary*) e férias (*vacation*). Uma célula de 2-D, que está no subespaço formado por idade e salário, contém L pontos somente se a projeção desta célula em todas as dimensões, ou seja, idade e salário, respectivamente, contém pelo menos l pontos (Han; Kamber & Pei, 2011).

**Figura 3.2** Unidades densas encontradas em relação as dimensões age, salary e vacation, interceptadas para fornecer um espaço de busca candidato para unidade s densas de alta dimensionalidade. **Fonte**: Han; Kamber & Pei (2011)



* **STING** - STING (Statistical INformation Grid-based) é uma técnica de agrupamento no qual a área espacial dos objetos de entrada é dividida em células retangulares. O algoritmo inicia dividindo a área espacial em níveis de células retangulares, correspondendo a níveis de resolução, formando uma estrutura hierárquica. As células de níveis mais alto são compostas a partir de células de nível mais baixo. Isso gera uma estrutura hierárquica das células produzidas pela subdivisão consecutiva do espaço. A Figura 3.3 mostra uma estrutura hierárquica para agrupamento STING. Os parâmetros estatísticos das células de mais alto nível podem ser facilmente calculadas a partir dos parâmetros das células de mais baixo nível. Os parâmetros são os seguintes: o parâmetro independente, *count*; e os parâmetros dependentes, média (*mean*), *stdev* (desvio padrão), *min* (mínimo), *max* (máximo), e os tipos de distribuição que os valores das células, como segue normal, uniforme e exponencial, ou nenhum (se a distribuição não é conhecida). Quando os dados são carregados do banco de dados, os parâmetros count, mean, stdev, min e max são diretamente calculados.

**Figura 3.3** Estrutura hierárquica para agrupamento STING. **Fonte**:(Han; Kamber & Pei, 2011).



A Figura 3.3, ilustra o processo da subdivisão de espaços e da consequente formação da hierarquia das células efetuado pelo algoritmo STING. A estrutura hierárquica de células representa a informação em diferentes níveis de agrupamento. Em cada nível, são registradas informações estatísticas básicas sobre os atributos de cada célula da grade como: frequência, média, desvio padrão, valor máximo e valor mínimo.

O algoritmo STING tem a propriedade de gerar bons resultados de agrupamento em um curto espaço de tempo de execução, entretanto existem duas grandes dificuldades com este algoritmo. Em primeiro lugar, o desempenho de STING depende da granularidade do nível mais baixo da estrutura de grade. Em segundo lugar, os grupos resultantes são somente limitados horizontalmente ou verticalmente, não existindo qualquer limitação na diagonal. Esta lacuna pode afetar significativamente a qualidade dos grupos (Han, Kamber & Pei, 2011).

De acordo com Han, Kamber & Pei (2011), uma interessante propriedade do STING é que ele aproxima-se do resultado de agrupamento DBSCAN se a granularidade é baixa.

Resumindo, a etapa de Extração de Padrões é direcionada ao cumprimento dos objetivos definidos na Identificação do problema. Nessa etapa é realizada a escolha, a configuração e execução de um ou mais algoritmos para extração de conhecimento. Como esta etapa é um processo iterativo, pode ser necessário que seja executada diversas vezes para ajustar o conjunto de parâmetros visando à obtenção de resultados mais adequados aos objetivos preestabelecidos (Rezende, 2005).

A disponibilização do conjunto de padrões extraídos nesta etapa ao usuário ou a sua incorporação a um Sistema Inteligente ocorre após a análise e/ou o processamento dos padrões na etapa de Pós-processamento.

* 1. **Pós-processamento**

Segundo Liu & Hsu (1996), a obtenção do conhecimento não é o passo final do processo de Extração de Conhecimento de Bases de dados. O conhecimento extraído pode ser utilizado na resolução de problemas da vida real, seja por meio de um Sistema Inteligente ou de um ser humano como apoio a algum processo de tomada de decisão. Para isso é importante que algumas questões sejam respondidas aos usuários:

* O conhecimento extraído representa o conhecimento do especialista?
* De que maneira o conhecimento do especialista difere do conhecimento extraído?
* Em que parte o conhecimento do especialista está correto?

Em geral, os algoritmos de Extração de Padrões podem gerar uma quantidade enorme de padrões, muitos dos quais podem não ser importantes, relevantes ou interessantes para o usuário. Então, fornecer ao usuário uma grande quantidade de padrões descobertos não é produtivo pois, normalmente, ele procura uma pequena lista de padrões interessantes. Portanto, é de vital importância desenvolver algumas técnicas de apoio no sentido de fornecer os usuários apenas os padrões mais interessantes (Goldschmidt, 2003).

Segundo Rezende (2005), diversas medidas para avaliação de conhecimento têm sido pesquisadas com a finalidade de auxiliar o usuário no entendimento e na utilização do conhecimento adquirido.

O Processo de Extração de Conhecimento, tem como um dos principais objetivos, é que o usuário possa compreender e utilizar o conhecimento descoberto. Entretanto, podem ocorrer casos em que os modelos são muito complexos ou não fazem sentido para os especialistas (Pazzani, 2000). Assim, a compreensibilidade do conhecimento extraído é um aspecto bastante importante para o processo de Extração de Conhecimento.

A compreensibilidade de um dado conjunto de regras está relacionada com a facilidade de interpretação dessas regras por um ser humano. Segundo Pazzini (2000), a compreensibilidade de um modelo pode ser estimada, por exemplo, pelo número de regras e o número de condições por regra. Nesse caso, quanto menor a quantidade de regras de um dado modelo e menor o número de condições por regra, maior será a compreensibilidade das regras descobertas.

A maneira de avaliar a qualidade tentando estimar o quanto de conhecimento interessante (ou inesperado) existe, deve combinar fatores numa medida que reflita como o especialista julga o padrão (Piatetsky-Shapiro & Matheus, 1994). As medidas de interesse estão baseadas em vários aspectos, principalmente na utilidade que as regras representam para o usuário final do processo de Extração de Conhecimento (Dong & Li, 1998). Estas medidas podem ser divididas em objetivas e subjetivas (Silberschatz & Tuzhilin, 1995; Piatetsky-Shapiro & Matheus, 1994).

Segundo Horst (1999), medidas objetivas são aquelas que estão relacionadas somente com a estrutura de padrões e do conjunto de dados de teste. Elas não levam em consideração fatores específicos do usuário nem do conhecimento do domínio para avaliar um padrão. Algumas medidas de interesse são: modelos de regras, cobertura de regras mínimas, custo da classificação incorreta.

Como no Processo de Extração de Conhecimento, pode ter diferentes usuários que, podem ter diferentes graus de interesse para um determinado padrão, medidas subjetivas são necessárias. Estas medidas consistem em fatores específicos do conhecimento do domínio e de interesse do usuário, que devem ser tratadas ao selecionar um conjunto de regras interessantes ao usuário final. Algumas regras subjetivas são inesperabilidade e utilidade (Silberschatz & Tuzhilin, 1995).

Segundo Rezende (2005), em um ambiente para avaliação de conhecimento, aspectos objetivos de interesse podem ser utilizados como um primeiro filtro para selecionar regras potencialmente interessantes ao usuário. Os aspectos subjetivos pode ser utilizados como um filtro final para selecionar regras realmente interessantes.

Após a análise do conhecimento, caso este não seja de interesse do usuário final ou não cumpra com os objetivos propostos, o processo de extração pode ser repetido ajustando-se os parâmetros ou melhorando o processo de escolha dos dados para a obtenção de resultados melhores numa próxima iteração (Rezende, 2005).

Existem várias estratégias propostas na literatura para pós-processar o conhecimento descoberto, entre elas a atribuição de medidas de potencial grau de interesse as quais são organizadas em dois grupos, ditas user-driven e data-driven (Silberschatz & Tuzhilin, 1996), (Freitas, 1998).

Outra estratégia é proposta por Hussain et al. (2000) que constitui um método que identifica, a partir de um conjunto de padrões descobertos, um subconjunto de regras que representam regras de exceção e, além disso, atribui uma medida de interesse para cada regra. A tabela 1 mostra a estrutura geral das regras de exceção. Nesta tabela A, B e C são conjuntos não-vazios de itens de dados associados e o símbolo “¬” denota a negação lógica. É importante observar que uma regra de exceção é uma especialização de uma regra geral e uma regra de exceção associa a um item de dados que nega aquele identificado pela regra geral. Este método assume que regras de senso comum representam padrões conhecidos pelo usuário, tendo em vista que aquelas regras têm uma grande cobertura, ao contrário das regras de exceção, que em geral são desconhecidas, uma vez que elas têm baixa cobertura. Sendo assim, as regras de exceção tendem a ser surpreendentes, dado o fato de representarem uma contradição em relação à regra de senso comum. É importante observar que a regra de referência auxilia na explicação da causa da regra de exceção.

**Tabela 1 Estrutura de Regras de Exceção**.

|  |
| --- |
| A 🡪 C regra geral (alta cobertura e alta confiança) |
| A, B 🡪 ¬ C regra de exceção (baixa cobertura, alta confiança) |
| B 🡪 ¬ C regra de referência (baixa cobertura e/ou baixa confiança) |

Formalmente, a medida proposta por Hussain et al. (2000) é definida da seguinte forma:

**(3.24)**

Quanto maior o valor da medida de interesse, maior é a chance de a regra ser surpreendente.

Outra estratégia para o Pós-processamento é o filtro de regras de associação, que objetiva selecionar aquelas que associem alguns elementos previamente selecionados (ou descartados) pelo especialista. Esta estratégia, a partir da redução do conjunto de regras, além de facilitar a análise elo especialista também melhora significativamente o desempenho de algoritmos que a partir deste conjunto reduzido venha executar outras estratégias de Pós-processamento. A parametrização do processo de filtro é a partir dos identificadores dos itens de dados que compõem as regras, conforme exemplo:

Regra 1: A 🡪 C

Regra 2: A, B 🡪 D

Regra 3: C 🡪 E

Regra 4: A, D 🡪 C

Supondo que o especialista não esteja interessando em regras em que apresentem o item “D”, independentemente se este consta do antecedente ou do consequente, apenas as Regras 1 e 3 serão selecionadas. Vale destacar novamente que essa estratégia somente é interessante quando o especialista tem conhecimento prévio do que deseja que seja contemplado ou eliminado, caso contrário, padrões interessantes podem ser eliminados. Uma alternativa a esta funcionalidade seria eliminar os itens de dados da base, porém esta alternativa onera computacionalmente a etapa de Pré-processamento, exigindo que um novo conjunto de dados seja construído a cada experimento (Milani & Carvalho, 2013).

Ainda segundo (Milani & Carvalho, 2013), existem ainda outras estratégias para a eliminação de regras extraídas que não agregam novos conhecimentos ao especialista, como por exemplo, a eliminação de redundância. Por exemplo, a partir do conjunto das Regras 1, 2, 3 e 4 percebe-se uma redundância da Regra 4 em relação a Regra 1, ou seja, a Regra 4 já está contemplada pela Regra 1, desta forma a Regra 4 pode ser eliminada do conjunto.

No que se refere aos padrões descobertos pela tarefa de classificação também existem diversas formas para pós-processar os padrões extraídos, entre elas: transcrição da árvore de decisão em regras, eliminação de redundância e atribuição de medidas de interesse, por exemplo, a partir de generalizações sucessivas.

* 1. **Tecnologia de Suporte a Mineração de Dados**

O processo de KDD é realizado incorporando-se várias técnicas de diferentes áreas como Aprendizado de Máquina, Data Warehousing, Banco de Dados, Estatísticas, Visualização de Dados, dentre outras. A etapa de Extração de Conhecimento, por exemplo, utiliza muitos recursos da área de Aprendizado de Máquina, e as demais são consideradas como áreas de apoio ao processo de KDD.

**3.6.1 Estatísticas**

A estatística estuda as coleções, análise, interpretação ou explicação, e apresentação dos dados. A mineração de dados tem conexão própria com a estatísticas (Han; Kamber & Pei, 2011).

Segundo Han; Kamber & Pei (2011), um modelo estatístico, é um conjunto de funções matemáticas, que descrevem as ações dos objetos, em uma classe alvo, em termos de suas variáveis aleatórias e sua distribuição de probabilidades associadas. Ainda segundo autores, os modelos estatísticos são muito usados para modelar dados e classes de dados. Por exemplo, pode-se utilizar os modelos estatísticos para caracterização e classificação de dados em tarefas de Mineração de Dados. Em outras palavras, tais modelos estatísticos podem ser o resultado de uma tarefas de Mineração de Dados. Alternativamente, tarefas de Mineração de Dados podem ser construídas em cima dos modelos estatísticos.

As áreas de Mineração de Dados e Estatística estão fortemente ligadas. Ambas as disciplinas têm como objetivo encontrar padrões e regularidade nos dados. Em geral, a Mineração de Dados enfatiza não só a precisão, mas também a facilidade de entendimento do conhecimento adquirido (Rezende, 2005).

Assim, a Mineração de Dados usa diversos mecanismos disponíveis na Estatística para realizar a descoberta de padrões, calcular aproximações, médias, taxas de erros e desvios. Para a Mineração de Dados é necessário o suporte de várias outras técnicas incluindo organização e estrutura de dados (Rezende, 2005).

Vale ressaltar que técnicas estatísticas também são utilizadas para avaliação de hipóteses e do conhecimento obtido no processo de Mineração de Dados. Além disso, no processo de extração de padrões o aprendizado Bayesiano exerce um papel importante. Entre esses algoritmos utilizados em Mineração de Dados estão o *Naive Bayes* e o *AutoClass* (Rezende, 2005).

Existem muitas outras áreas da Estatísticas que tem sido fundamentais para a Mineração de Dados. No entanto, não é o objetivo desse trabalho analisar todas, pois trata-se de um assunto bastante abrangente.

* + 1. **Aprendizado de Máquina**

De acordo com Han; Kamber & Pei (2011), aprendizagem de máquina, investiga como os computadores pode aprender, com base em dados.

Segundo Rezende (2003), aprendizado de máquina, é uma área de Inteligência Artificial (**IA**), cujo objetivo, é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

Para Han; Kamber & Pei (2011), a principal área de pesquisa está voltada para programas de computador, aprenderem automaticamente reconhecer padrões complexos, com base em dados, e a partir daí, tomar decisões inteligentes. Por exemplo, um problema típico de aprendizado de máquina, é programar um computador para que ele possa reconhecer automaticamente os códigos postais escritos, depois de aprender a partir com um conjunto de exemplos.

De acordo com Mitchell (1997), aprendizado de máquina, é uma sub-área da Inteligência Artificial **(IA**), cujo objetivo é desenvolver métodos, técnicas e ferramentas para construir máquinas inteligentes, que se modificam para realizar cada vez melhor sua(s) tarefa(s).

Para aprender, o sistema, bem como os seres humanos, podem se valer de estratégias de aprendizado. A seguir, será apresentado, algumas dessas estratégicas clássicas de aprendizagem de máquina relacionado a mineração de dados (*data mining*) (Han; Kamber & Pei, 2011):

* **Aprendizado supervisionado**: é basicamente um sinônimo para classificação. No aprendizado supervisionado o objetivo é induzir conceitos a partir de exemplos que estão pré-classificados, ou seja, exemplos que estão rotulados com uma classe conhecida. Se as classes possuírem valores discretos, o problema é caracterizado como classificação. Caso as classes possuam valores contínuos, o problema é caracterizado, como regressão (Han; kamber & Pei, 2011). Em geral, de acordo com Rezende (2003), cada exemplo é descrito por um vetor de valores de características, ou atributos, e pelo rótulo da classe associada. O objetivo do algoritmo de indução, é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados, ou seja, exemplos que não tenham rótulo da classe. Um algoritmo de aprendizado supervisionado pode ser utilizado quando, em um banco de dados, se tem tanto as perguntas com as respostas. Usado para a realização de treinamento de redes neurais na obtenção de classificação, funções de aproximação ou modelagem e previsões baseadas no tempo (Bigus, 1999).
* **Aprendizado não-supervisionado**: é essencialmente, um sinônimo para agrupamento (*cluster*). No aprendizado não-supervisionado, o indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou *clusters*. A tarefa do algoritmo é agrupara exemplos não rotulados, i.e., exemplos que não possuem o atributo classe especificado. Nesse caso, é possível utilizar algoritmos de aprendizado para descobrir padrões nos dados a partir de alguma caracterização de regularidade, sendo esses padrões denominados *clusters* (Decker & Focardi, 1995; McCallum, Nigan, & Ungar, 200). Exemplos contidos em um mesmo cluster são mais similares, segundo alguma medida de similaridade, do que aqueles contidos em clusters diferentes. O processo de formação dos clusters é geralmente conhecido por *clustering*. Tipicamente, pode-se usar agrupamento para descobrir classes dentro de dados (Han; Kamber & Pei, 2011).
* **Aprendizado semi**-**supervisionado**: quando existem exemplos rotulados, pode-se utilizar o aprendizado supervisionado para induzir classificadores a partir destes exemplos. Caso contrário, quando os exemplos não estão rotulados, pode-se utilizar o aprendizado não-supervisionado com o objetivo de encontrar os *cluster*s. Já o aprendizado semi-supervisionado, consiste em utilizar algoritmos que aprendam apartir de exemplos rotulados e não rotulados. Ou seja, o aprendizado semi-supervisionado, pode ser aplicável tanto em tarefas de classificação quanto em tarefas de agrupamento (Han; kamber & Pei, 2011).
* **Aprendizado ativo**: é uma abordagem de aprendizado de máquina, que permite que o usuário execute um papel ativo no processo de aprendizagem. Uma abordagem de aprendizado ativo, pode pedir ao usuário (i.e., o especialista do domínio) para rotular um exemplo, que pode ser a partir de um conjunto de exemplos não-rotulados, ou sintetizado pelo programa de aprendizagem. O objetivo é otimizar a qualidade do modelo, através de aquisição de conhecimento ativamente do usuário humano, a partir da quantidade exemplos a serem rotulados (Han; Kamber & Pei, 2011).

Pode-se ver que existem muitas similaridades, entre mineração de dados e aprendizado de máquina. Para as tarefas de classificação e agrupamento, as pesquisas em aprendizado de máquina, frequentemente focam, na precisão do modelo. Em adicional a precisão, as pesquisas em mineração de dados, dão muita ênfase na eficiência e escalabilidade de métodos de mineração em grandes massas de dados, bem como na maneira de lidar com tipos de dados complexos e, explorar novos métodos alternativos (Han; Kamber & Pei, 2011).

* 1. **Aplicações de Mineração de Dados**

As tecnologias de Mineração de Dados podem ser aplicadas em grande variedade de contextos de tomada de decisão. Em particular, áreas de significativo retorno de investimento esperado incluem:

* ***Marketing*** – aplicações como análise de comportamento do consumidor baseados em padrões de consumo; a definição de estratégias de *marketing* incluem propaganda, localização de lojas e mala direta direcionada; segmentação de clientes, layouts de lojas e campanhas de publicidades.
* **Finanças** – aplicações incluem análise de crédito de clientes, segmentação de contas a receber, análise de performance de investimentos financeiros como ações, bonds e fundos mútuos e detecção de fraudes.
* **Produção**- aplicações envolvem otimização de recursos como máquinas, força de trabalho e materiais; projeto ótimo de processos de fabricação e projeto de produção de automóveis baseados nos requisitos dos clientes.
* **Saúde** – aplicações incluem descoberta de padrões em imagens radiológicas, análise de dados experimentais em microarray (gene-chip) para relação com doenças, análises de efeitos colaterais de remédios e efetividade de certos tratamentos, etc.
  1. **Visualização de Dados**

As técnicas e ferramentas para Visualização de Dados são indispensáveis ao processo de Mineração de dados. Elas podem ser usadas durante a execução das etapas do processo de extração de conhecimento melhorando a compreensão dos resultados obtidos e a comunicação entre os usuários (Rezende et al., 1998).

As técnicas de Visualização de Dados estimulam naturalmente a percepção e a inteligência humana, aumentando a capacidade de entendimento e associação de novos padrões. Logo, a Visualização de Dados utiliza a percepção humana como um primeiro método para descobrir valores. Poderosas ferramentas de visualização que consigam gerar diversas formas de visualização (árvores, regras, gráficos 3D/2D, espectro) combinadas com técnicas de Mineração de Dados podem melhorar o processo de Mineração de dados (Fayyad, Grinstein, & Wierse, 2002).

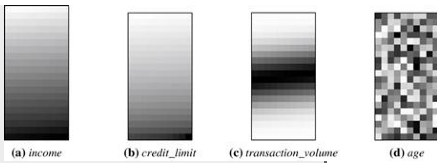
Segundo Han, Kamber & Pei (2011), a visualização de dados, utiliza várias abordagens, incluindo técnicas orientada a pixel, técnicas de projeção geométrica, técnicas baseada em ícone, hierárquica e técnicas baseada em gráfico.

* + 1. **Técnicas de Visualização Orientada a Pixel**

**U**ma maneira simples de visualizar os valores de uma dimensão é usar pixels, onde a cor do pixel reflete os valores da dimensão. Para um conjunto de dados de *m* dimensões, as técnicas ***pixel-oriented***, criam *m* janelas na tela, uma para cada dimensão. Os *m* valores das *m* dimensões de um registro, são mapeados para *m* pixels para uma posição correspondente na janela. As cores dos pixels refletem os valores correspondentes (Han, Kamber & Pei, 2011).

De acordo com Han, Kamber & Pei (2011), dentro de uma janela, os dados são organizados em um ordem globalmente compartilhada por todas as janelas. A ordem global, pode ser obtida, por classificar todos os registros de dados, de uma forma significativa para a tarefa em questão. A **Figura 3.4**, a seguir, mostra um gráfico orientado a pixel, onde, são observados quatro atributos dos clientes: renda (*income*), limite de crédito (*credit\_limit*), volume de transações (*transaction\_volume*) e idade (*age*).

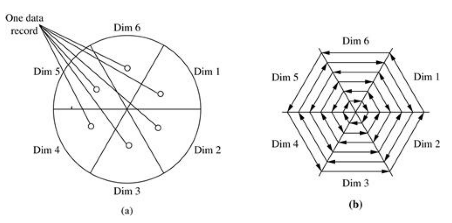
**Figura 3.4**. Adotada de Haa, Kamber & Pei (2011). Visualização de dados de quatro atributos, ordenados pela renda do cliente (ascendente).



Segundo Han, Kamber & Pei (2011), a representação dos registros de dados de forma linear, pode não funcionar bem, quando o número de atributos é muito grande. Isso se deve, segundo Branco (2002), ao fato da dependência direta em relação a resolução da tela, pois quanto maior a dimensionalidade dos dados, maior será o número de janelas e, consequentemente, menor será o número de atributos que poderão ser vistos simultaneamente. Para resolver esse problema, usa-se a técnica Segmentos Circulares (*Circle Segment*s), essa técnica, mapeia os dados em pixels coloridos em segmentos circulares. A **Figura 3.5**, ilustra exemplos dessa técnica (Han, Kamber & Pei, 2011).

**Figura 3.5**. Adotada de Han, Kamber & Pei (2011). A técnica Segmento Circular. Representando um registro de dados em segmentos circulares.

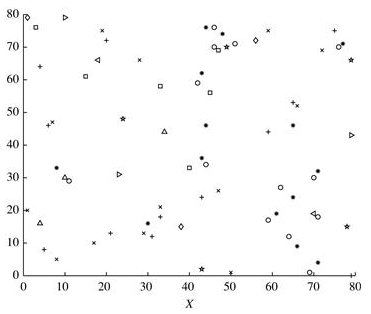
* + 1. **Técnica de Projeção Geométrica**



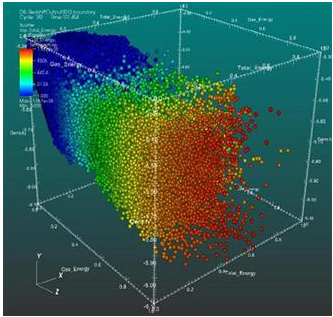
De acordo com Han, Kamber & Pei (2011), as técnicas de projeções geométricas, projetam os dados multidimensionais em um espaço bidimensional, buscando apresentar projeções interessantes do conjunto de dados. Ainda, segundo Han & kember (2011), em particular, uma técnica bastante utilizada desta categoria, é a técnica gráfico de dispersão (scatter plot), que plota os pontos de dados, usando coordenadas cartesianas. Pode-se adicionar uma terceira dimensão, usando cores ou formas diferentes, para representar pontos de dados. A **Figura 3.6**, mostra um exemplo, onde X e Y são dois atributos espaciais e a terceira dimensão é representada por diferentes formas.

Um gráfico em 3-D, usando gráfico de dispersão, usa três eixos em um sistema de coordenadas cartesianas. No entanto, se ele usar cores, ele pode representar pontos de dados em 4-D, como mostra a **Figura 3.7**.

**Figura 3.6**. Adotada de Han, Kamber & Pei (2011). Visualização de um conjunto de dados em 2-D, usando gráfico de dispersão (*scatter plot*).

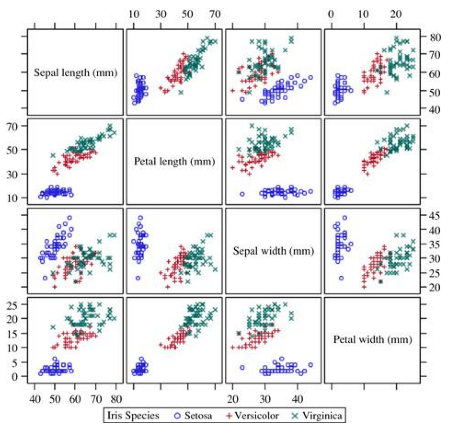


**Figura 3.7**. Adotada de Han, Kamber & Pei (2011). Visualização de um conjunto de dados em 3-D, usando gráfico de dispersão (*scatter plot*).



Para conjunto de dados com mais de quatro dimensões, os gráficos de dispersão, são usualmente ineficientes, segundo Han, Kamber & Pei (2011). A técnica matriz de dispersão (*scatter-plot matrix*), é uma generalização para o gráfico de dispersão (*scatter plot*). Para um conjunto de dados n-dimensional, a matriz de dispersão, é uma grade n x n de gráficos de dispersão 2-D, que fornecem uma visualização de cada dimensão com todas as outras dimensões. A **Figura 3.8**, mostra um exemplo, no qual visualiza o conjunto de dados de íris. O conjunto de dados, consiste de 450 exemplos de cada três espécies de flores de íris. Há cinco dimensões no conjunto de dados: comprimento (*length*) e tamanho (*width*) da pétala, e espécie.

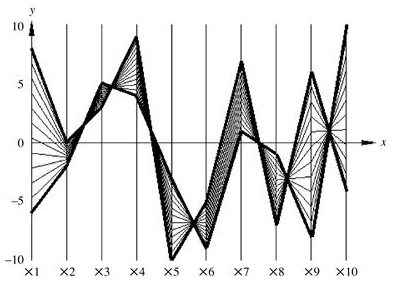
**Figura 3.8**. Adotada de Han, Kamber & Pei (2011). Visualização de um conjunto de dados de íris, usando uma matriz de dispersão (*scatter-plot matrix*).



A técnica de matriz de dispersão, torna-se menos efetiva, à medida que aumenta a dimensionalidade. Uma outra técnica, chamada Coordenadas Paralelas, pode manipular alta dimensionalidade. É uma técnica de visualização, onde as dimensões são representadas como uma série de eixos paralelos, uns aos outros, e com igual espaçamento entre eles, nos quais os valores estão representados (Inselberg; Avidan, 1999). Um registro de dados é representado por uma linha poligonal, que intercepta cada eixo para um correspondente ponto ao valor associado da dimensão (Han, Kamber & Pei, 2011). Veja a **Figura 3.9**.

**Figura 3.9**. Adotada de Han, Kamber & Pei (2011). Visualização que usa Coordenadas Paralelas.

* + 1. **Visualização Iconográfica**



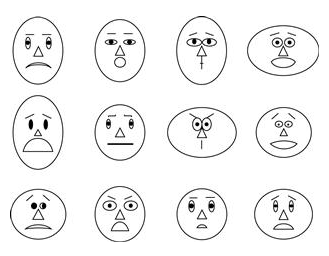
Essa técnica usa pequenos ícones para representar dados multidimensionais. Ou seja, essa técnica, tem como objetivo, mapear os atributos em características particulares de ícones. Cada característica do ícone, representa um atributo dos dados multidimensional (Han, Kamber & Pei, 2011]. Popularmente, essa técnica, trabalha com dois ícones: ***Chernoff faces*** and ***Stick figures***.

* + - 1. **Chernoff faces**

Esta técnica foi introduzido em 1973 pelo estatístico Herman Chernoff. Ele mapeou dados multidimensionais acima de 18 variáveis (ou dimensões) usando características faciais humanas (**Figura 3.10**). Essa técnica ajuda a revelar tendências nos dados. Componentes faciais, tais como os olhos, a boca, as orelhas, e nariz, representam valores atuais das dimensões pelo sua forma, tamanho, posicionamento e localização. Por exemplo, as dimensões podem ser mapeadas para seguir as características: tamanho dos olhos, espaço entre os olhos, comprimento do nariz, curvatura da boca, tamanho da boca, abertura da boca, tamanho da pupila, inclinação da sobrancelha, excentricidade dos olhos e cabeça.

As ***Chernoff*** faces explora a capacidade humana de reconhecer e analisar faces, para perceber pequenas diferenças em características faciais e assimilar muitas características faciais de uma só vez. Desta forma, elas facilitam a visualização de regularidades e irregularidades, presentes nos dados, embora seja limitado para relacionamentos múltiplos (Han & Kamber & Pei, 2011]. Ainda segundo os mesmos autores, uma outra limitação é que, os dados, propriamente dito, não são mostrados, uma vez que aquelas figuras, não transmitem qualquer informação sobre os reais valores ou relacionamentos.

**Figura 3.10**. Adotada de Han & Kamber & Pei (2011). Chernoff faces. Cada face representando um ponto de dado n-dimensional (n 18).



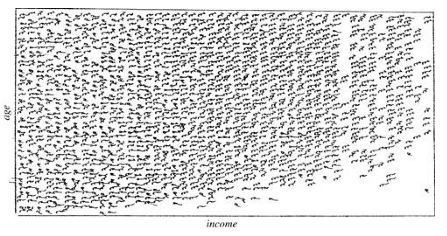
***3.8.4 Stick figures***

A técnica de visualização Stick figures, utiliza duas dimensões da tela para mapear duas dimensões dos dados e as demais dimensões são mapeadas para os ângulos e/ou comprimento dos segmentos de um ícone formado por múltiplos segmentos de retas (Han & Kamber & Pei, 2011). A Figura 3.11, mostra os dados do censo, onde idade e renda são mapeados para os eixos e as demais dimensões (sex, educação, e assim por diante) são mapeadas para *stick figues*.

Segundo Han & kamber & Pei (2011), quando mapeados na tela, os ícones (um para cada item de dado) formam texturas que variam de acordo com as características dos dados, permitindo identificar padrões na imagem que podem indicar dependência funcional entre os atributos visualizados.

**Figura 3.11**. Adotada de Han & Kamber & Pei (2011). Representação de Censo de dados, usando *Stick Figures*.

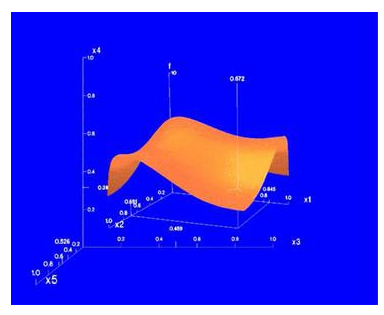
* + 1. **Técnica de Visualização Hierárquica**



De acordo com Han, kamber & Pei (2011), as técnicas hierárquicas, geralmente são aplicadas em dados, cuja própria natureza, apresenta uma correlação explicita entre níveis e subconjuntos. Sendo assim, o espaço n-dimensional dos dados (não necessariamente hierárquico) é então dividido em subespaços que são organizados uns dentro do outro de forma hierárquica.

“Worlds-within-Worlds”, também conhecido como n-Vision, é uma representação do método de visualização hierárquico. Suponha que se deseje visualizar um DateSet 6-D, onde as dimensões são **F**, **X1**, ..., **X5**. Deseja-se observar como a dimensão **F** muda em relação as outras dimensões. Se primeiro, fixarmos os valores das dimensões **X3**, **X4**, **X5** para alguns valores selecionados, **C3**, **C4**, **C5**. Então, pode-se visualizar **F**, **X1**, **X2** usando um plot 3-D, chamado um World, como ilustrado na Figura 3.12

**Figura** 3.12 “Worlds-Within-Worlds” (também conhecindo como n-Vision). Adaptada de Han, Kamber & Pei (2011).



A posição de origem no interior do gráfico é localizada pelo ponto (C1, C2, C3). O usuário pode interagir, alterar os valores desse ponto, e visualizar as mudanças no interior do gráfico.

* 1. **Conclusões**

Neste capítulo foi apresentada a disciplina de Mineração de Dados, que utiliza tecnologias para descobrir conhecimento adicional ou padrões nos dados. Foram discutidas várias técnicas, priorizando os detalhes das regras de associação, a classificação e o agrupamento. Apresentou-se também, alguns algoritmos para algumas dessas áreas.

Como a Mineração de Dados é uma área em constante desenvolvimento, diversas dificuldades e desafios estão surgindo continuamente, as quais precisam ser atacadas. Portanto, nessa sessão foram apresentadas algumas das principais tecnologia que dão suporte ao processo de Mineração de dados, sem a pretensão de esgotá-las.