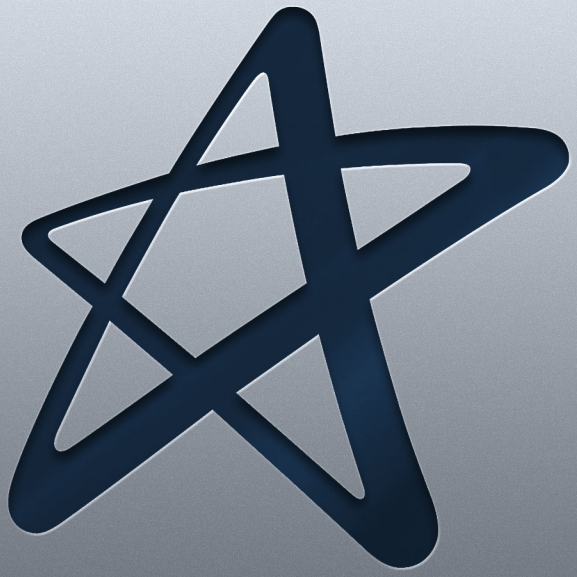
Machine Learning



Cruzeiro do Sul Virtual
Educação a distância

Material Teórico

Introdução aos Modelos Descritivos

Responsável pelo Conteúdo:
Prof. Me. Orlando da Silva Junior

Revisão Textual:
Prof.ª Dr.ª Selma Aparecida Cesarin

UNIDADE Introdução aos Modelos Descritivos

- Introdução aos Modelos Descritivos;
- Mineração de Padrões Frequentes;
- Análise de Agrupamentos;
- Algoritmos Hierárquicos.

OBJETIVOS DE APRENDIZADO

- Estudar as aplicações de aprendizagem não supervisionada;
- Conhecer as técnicas para a geração de modelos descritivos.

Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:

Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu "momento do estudo";
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e *sites* para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.

Introdução aos Modelos Descritivos

As tarefas de aprendizado descritivo (ou não supervisionado) trabalham com a identificação de informações relevantes nos dados, sem a necessidade de consultar ou observar um elemento externo para guiar o aprendizado, como acontece nas técnicas de aprendizado supervisionado.

Por causa disso, você deve lembrar que os métodos não supervisionados não compreendem a existência de atributos-alvo.

No conjunto de dados apresentado na Tabela 1, todos os atributos serão considerados atributos de entrada para o algoritmo. Para a aprendizagem não supervisionada, nenhum atributo será considerado atributo-alvo.

Ao contrário dos modelos preditivos, cujo objetivo central é a predição, os métodos não supervisionados induzirão à formação de representações que possam servir a diferentes propósitos na tomada de decisão.

Tabela 1 – Conjunto de dados de pacientes para o diagnóstico de doenças

Nome	Idade	Sexo	Temperatura	Dores	Diagnóstico
Maria	54	F	39.0	Sim	Doente
João	33	M	38.7	Não	Saudável
José	29	M	35.4	Sim	Saudável
Carlos	48	M	36.0	Não	Doente
Ana	21	F	36.5	Sim	Doente

As duas principais tarefas do aprendizado supervisionado e que você vai estudar aqui são a **mineração de padrões frequentes** e a **análise de agrupamentos**.

Você vai estudar técnicas para essas tarefas que são utilizadas sobretudo quando o objetivo é encontrar padrões ou descobrir tendências que ajudem a entender os dados.

Mineração de Padrões Frequentes

Vamos começar com a mineração de padrões frequentes, que tal?

A mineração de padrões frequentes (ou conjuntos de itens frequentes) é um dos temas principais em descoberta de conhecimento em bases dados, tendo começado com a análise de cesta de compras de Supermercado para determinar o comportamento de compra dos clientes.

O objetivo dessa tarefa é descobrir grupos de produtos que frequentemente são comprados em conjunto e, a partir desses grupos, inferir os produtos que serão comprados, haja vista que foram comprados outros produtos (FACELLI *et al.*, 2011).

Em geral, a mineração de padrões frequentes é realizada em bases de dados transacionais, que são aquelas que recebem periodicamente as transações realizadas pela Organização.

Ao contrário das bases analíticas que tem seu foco no nível estratégico, as bases de transacionais são focadas na operação. Enquanto o operador de Caixa de Supermercado opera uma base transacional, incluindo as compras que você está realizando, o Gerente do Supermercado olhará o banco de dados analítico para verificar o faturamento total realizado no dia.

Em nossos estudos, vamos aplicar a mineração de padrões frequentes considerando o processo ilustrado na Figura 1, que apresenta as 4 etapas dessa tarefa, sendo elas:

- **Pré-processamento da base de dados:** compreende as atividades de limpeza, integração, redução e transformação da base de dados original;
- **Geração do conjunto de itens frequentes:** construção dos conjuntos de itens que satisfazem algum critério de frequência. Você vai compreender melhor esta etapa mais à frente;
- **Mineração das regras de associação:** compreende a aplicação dos algoritmos de mineração para a extração das regras de associação que indicarão os padrões encontrados;
- **Avaliação de desempenho:** corresponde à avaliação de desempenho dos algoritmos aplicados e à avaliação das regras extraídas pelo processo.

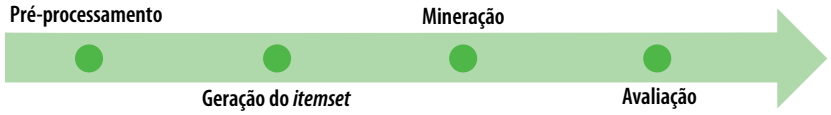


Figura 1 – Processo de mineração de itens frequentes

Como exemplo prático para geração do conjunto de itens frequentes, observe a base transacional da Tabela 2. Nela, cada linha representa uma transação que pode ser identificada pelo atributo Identificador.

O conjunto de itens dessa transação é registrado pelo atributo Itens.

Tabela 2 – Base de dados transacional de um Supermercado

Identificador	Itens
1	{pão, leite}
2	{café, leite, açúcar}
3	{pão, açúcar, manteiga}
4	{leite, açúcar}
5	{manteiga, banana, sal}

Podemos usar os dados da base transacional apresentada na Tabela para encontrar relações entre os itens. Por exemplo, observe a seguinte regra:

$$\{pão, leite\} \rightarrow \{café\}$$

Por essa regra, podemos entender que **se um consumidor compra pão e leite, então existe a possibilidade desse mesmo consumidor também comprar café**.

Os elementos presentes entre as chaves e fazem parte de um mesmo conjunto de itens (ou *itemset*).

A construção de regras é a forma que a mineração de padrões frequentes utiliza para estabelecer as relações entre as transações. Ao contrário da análise de agrupamentos, que estabelece relações a partir da similaridade entre os exemplos, a mineração de padrões frequentes utiliza as regras de associação para os itens presentes na Base de Dados.

Para que a mineração aconteça, precisaremos transformar a Base de Dados transacional em uma Base de Dados convencional.

A Tabela 3 ilustra como a base apresentada na Tabela 2 deverá ficar para o problema de mineração.

Tabela 3 – Base de Dados do Supermercado transformada

Identificador	Pão	Leite	Café	Açúcar	Manteiga	Banana	Sal
1	1	1	0	0	0	0	0
2	0	1	1	1	0	0	0
3	1	0	0	1	1	0	0
4	0	1	0	1	0	0	0
5	0	0	0	0	1	1	1

Nessa nova Base de Dados transformada, observe que os dados que trabalharemos com uma representação binária dos itens. Observe também que não estamos trabalhando com outras informações dos produtos, como o preço unitário e a quantidade adquirida em cada transação.

Com o problema definido, basta escolhermos, agora, qual algoritmo de aprendizado irá gerar as regras de associação.

O principal e mais conhecido dessa categoria é o Apriori. Ele surgiu a partir dos trabalhos de Agrawal e Srikant publicados em 1993 e 1994.

O objetivo originário desse algoritmo era descobrir associações de produtos em grades de Bases de Dados transacionais obtidas em Supermercados.

O *Apriori* segue a ideia de que qualquer subconjunto de *itemsets* frequentes deve ser um *itemset* frequente (FACELLI *et al.,* 2011).

Juntamente com essa premissa, a estratégia utiliza a busca em largura com um algoritmo de geração e teste. Para cada nível, são gerados *itemsets* candidatos, considerando os *itemsets* frequentes do nível anterior.

Após a geração, a frequência dos *itemsets* é testada.

A **busca em largura** é um algoritmo de busca em grafos que explora uma estrutura de árvores expandindo o nó raiz em primeiro lugar e, em seguida, os seus sucessores, repetidamente. A implementação da busca em largura toma a ideia da fila (o primeiro a entrar é o primeiro a sair), fazendo com que os nós antigos sejam expandidos primeiro. É um algoritmo ótimo e completo.

Outro algoritmo de busca em grafos bastante conhecido é a **busca em profundidade**. Ela segue a estratégia de sempre expandir o nó mais profundo da fronteira atual da árvore, procedendo imediatamente para o nível mais profundo da árvore de busca, onde os nós não têm sucessores. Assim que esses nós são expandidos, eles são removidos da fronteira para que a busca "guarde" o próximo nó mais profundo ainda não explorado.

Na prática, o *Apriori* realiza diversas varreduras sobre o Banco de Dados para calcular o suporte dos conjuntos de itens frequentes candidatos.

O suporte de uma regra que corresponde à uma medida que indica a frequência de ocorrência da regra. Regras com baixo suporte são regras que ocorrem ocasionalmente, tornando-se pouco interessantes para o negócio.

Podemos calcular o suporte de uma regra com a fórmula a seguir (CASTRO; FERRARI; 2016):

$$Suporte(A \rightarrow B) = P(A \cup B) = \frac{\sigma(A \cup B)}{n}$$

Onde:

- *Suporte (regra)* retorna a probabilidade de *regra* ser encontrada no conjunto total de transações;
- $A \rightarrow B$ é a regra de associação em que A implica em B;
- $\sigma(x)$ é a contagem do suporte da regra correspondendo à quantidade de transações de um determinado conjunto de itens; e
- n é a quantidade total de transações da base de dados.

As dificuldades do *Apriori* com o método implementado de geração e teste dos conjuntos candidatos foram combatidas com a chegada do ***FP-Growth***, que usa uma árvore *FP-Tree* para armazenar de forma comprimida a informação sobre os padrões frequentes, permitindo que o conjunto completo seja extraído.

A *FP-Tree* (*Frequent Pattern Tree*, Árvore de Padrões Frequentes) é uma estrutura de árvore que possui um nó raiz rotulado como nulo, um conjunto de subárvores de itens prefixos como sucessores diretos dessa raiz e uma tabela de cabeçalho com os itens frequentes.

Para cada transação que precisar ser inserida na árvore, o algoritmo de construção da árvore selecionará e ordenará o conjunto de itens frequentes da transação antes de inserir. Dessa forma, a árvore poderá ser construída com apenas duas leituras da base de dados: a primeira leitura determina e ordena o conjunto de itens frequentes; a segunda leitura realiza a construção da árvore.

Com a árvore *FP-Tree* construída, o algoritmo de mineração *FP-Growth* poderá minerar os itens frequentes da Base de Dados usando a árvore.

Após encontrar os conjuntos de itens frequentes para os nós folha, o algoritmo gera os conjuntos de dados para os pais da folha e assim sucessivamente, até chegar à raiz.

Ao contrário do *Apriori*, o *FP-Growth* emprega a busca em profundidade.

Análise de Agrupamentos

A análise de agrupamentos tem o propósito de encontrar estruturas de grupos nos dados buscando por características que sejam compartilhadas entre os exemplos dos grupos.

A análise de agrupamentos é um processo que pode ser conduzido em 5 etapas, conforme ilustrado na Figura 2 (FACELLI *et al.*, 2011):

- **Preparação dos Dados:** engloba aspectos relacionados ao pré-processamento e à forma de representação dos dados que servirão de entrada ao algoritmo de agrupamento. Entre as principais técnicas de pré-processamento estão a normalização dos dados, a conversão de tipos de dados e a redução do número de atributos. Em relação à representação, a maior parte dos algoritmos aceita as representações já estudadas. Em alguns casos, são também comuns a utilização de matrizes e grafos de similaridades;
- **Proximidade:** consiste na definição de medidas de proximidade relacionadas ao domínio da aplicação. A escolha deve considerar os tipos e as escalas dos atributos, vez que a similaridade admite que todos os atributos têm igual importância. Nesse sentido, a correlação é a medida de similaridade mais usada, enquanto a distância euclidiana é a medida de dissimilaridade mais usual. Quando lidamos com atributos qualitativos, a distância de Hamming pode ser a mais adequada;
- **Agrupamento dos Dados:** nesta etapa, é realizada a aplicação de um ou mais algoritmos de agrupamento para a identificação dos possíveis grupos existentes nos dados. Mais à frente, você conhecerá alguns algoritmos para realizar esta tarefa;
- **Validação dos Grupos:** esta etapa avalia o resultado do agrupamento, procurando determinar se os grupos gerados pelo algoritmo são significativos. Outro ponto importante nesta etapa é a validação de desempenho do algoritmo, que pode incluir uma comparação entre os algoritmos, a estrutura dos grupos formados, a presença de *outliers*, a sobreposição dos grupos e a escolha da medida de similaridade;

- **Interpretação dos Resultados:** compreende a inspeção dos grupos formados em relação aos exemplos que foram agrupados pelo algoritmo. Pode envolver também a descrição da natureza dos grupos pelo especialista, de modo a validar e rotular subjetivamente os grupos por meio de um significado prático.

Figura 2 – Processo de Análise de Agrupamentos

Considerando a comparação de similaridade entre dois exemplos, a distância de Hamming mede o número de atributos categóricos com valores diferentes entre esses exemplos. Na prática, essa medida informa a quantidade mínima de alteração que precisariam ser feitas para que um valor seja transformado no outro.

Nas próximas seções, você conhecerá os principais algoritmos que podem ser aplicados na terceira etapa (Agrupamento).

Para facilitar o seu entendimento em relação a como podem ser aplicados, eles foram organizados de acordo com o método adotado para a definição dos grupos:

- Algoritmos hierárquicos;
- Algoritmos particionais; e
- Algoritmos baseados em densidade.

Algoritmos Hierárquicos

Os algoritmos de agrupamento hierárquicos produzem uma sequência de partições rígidas aninhadas em que cada partição contém uma quantidade diferente de grupos.

Os métodos de agrupamento hierárquico usam métricas de integração (*linkage metrics*) que representam as medidas de distância entre os grupos.

Esses métodos inicializam um agrupamento como um conjunto de grupos de um elemento (aglomerativo) ou um único grupo com todos os elementos (divisivo) e iterativamente unem ou dividem os grupos mais apropriados até que um critério de parada seja satisfeito.

A Figura 3 ilustra o processo de construção dos grupos a partir dessas duas estratégias:

- **Abordagem aglomerativa:** inicia com um número pré-definido de grupos com um único exemplo e forma a sequência de partições reunindo os grupos sucessivamente;

- **Abordagem divisiva:** inicia com um grupo com todos os objetos e forma a sequência de partições dividindo os grupos sucessivamente.

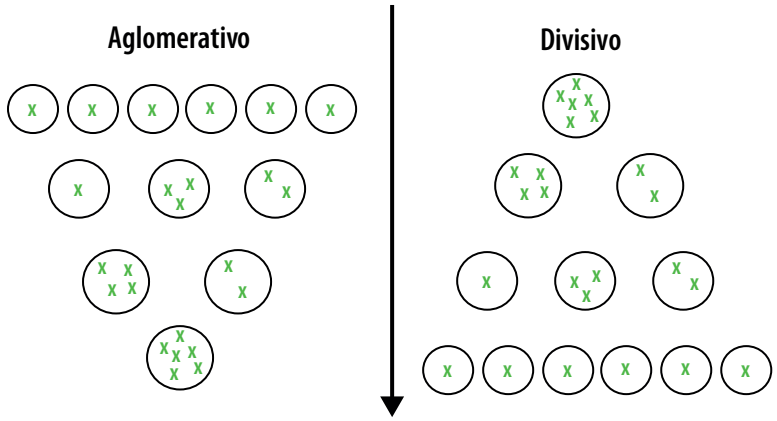


Figura 3 – Agrupamento hierárquico aglomerativo e divisivo

Para exemplificar, considere dois grupos: G_1 e G_2 , com seus respectivos centroides C_1 e C_2, que quantificam as distâncias entre os grupos. Tendo o propósito de calcular a distância de todos os exemplos entre C_1 e C_2, cada uma das distâncias entre os grupos estabelecidas pelo valor mínimo, médio ou máximo dos valores das distâncias dos exemplos é uma métrica de integração, que pode ser:

- **Ligação simples (*single-linkage*):** a distância estabelecida entre os grupos G_1 e G_2 corresponde à distância entre os exemplos dos dois grupos que estão mais próximos, ou seja, pela distância mínima entre quaisquer dois exemplos, sendo um de cada grupo. Esse tipo de ligação não exige o número de grupos como entrada e é indicado para manipular formas não elípticas, embora seja mais sensível a ruídos e *outliers*;
- **Ligação média (*average-linkage*):** a distância entre os exemplos dos dois grupos G_1 e G_2 é a distância média;
- **Ligação completa (*complete-linkage*):** a distância estabelecida entre os grupos G_1 e G_2 é a distância entre os exemplos mais distantes. Os algoritmos dessa categoria sofrem menos com ruídos e *outliers*.

Muitas vezes, a análise representada pela utilização dos algoritmos hierárquicos é um **dendograma**, que consiste em uma árvore binária que representa uma hierarquia de partições.

Ele é formado por camadas de nós, cada uma representando um grupo. O corte de um dendograma na horizontal representa uma partição.

Para ilustrar a análise pelo dendograma, observe os 10 pontos de dados no espaço presentes na Figura 4.

Você pode observar a separação desses dados entre dois grupos: o primeiro grupo considerando os pontos de 1 a 5, e o segundo grupo considerando os pontos de 6 a 10.

Utilizando *single-linkage* com distância euclidiana, o algoritmo hierárquico começa por localizar os pontos mais próximos. No nosso exemplo, tanto os pontos 3 e 4 quanto os pontos 7 e 8 são os mais próximos entre si e, a partir deles, um grupo pode ser formado para cada par de pontos.

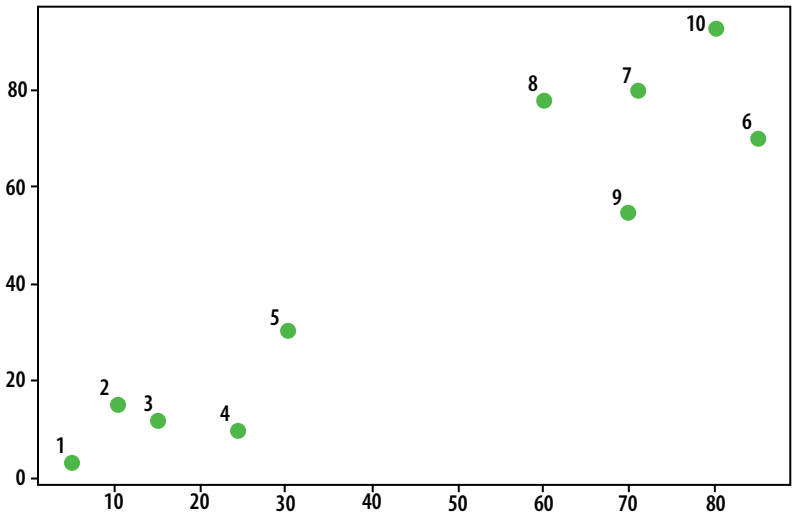


Figura 4 – Exemplo de dados no espaço

O gráfico ilustrado na Figura 5 corresponde ao processo completo de formação do dendograma para esse caso que, iterativamente, forma os grupos de maneira hierárquica, em que a altura vertical corresponde à distância euclidiana entre os pontos.

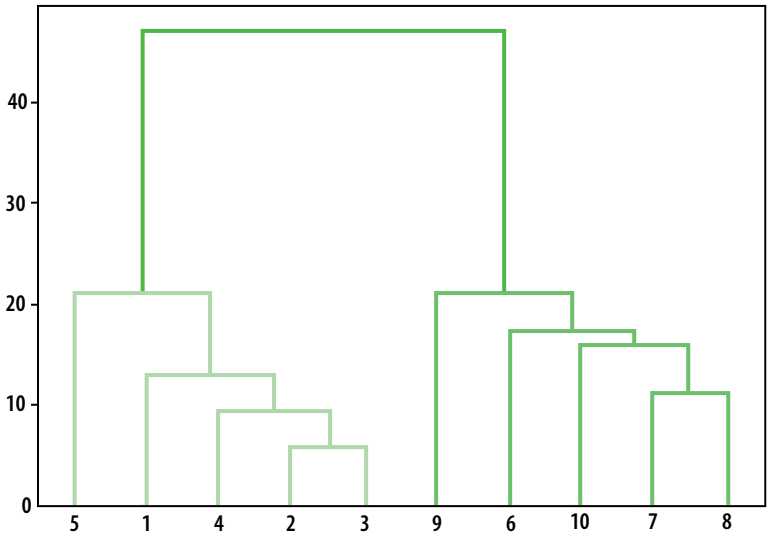


Figura 5 – Dendograma

Algoritmos Particionais

Outra categoria de algoritmos de agrupamento são os algoritmos particionais. O objetivo deles é otimizar o critério de agrupamento usando alguma técnica iterativa.

Inicialmente, esses algoritmos criam uma partição inicial para, em seguida, mover os exemplos de um grupo para outro no propósito de melhorar o valor do critério de agrupamento.

Em geral, o critério utilizado é o erro quadrático, que permite manter os grupos compactos. Dessa forma, para um número fixo de grupos, o algoritmo obtém uma partição que minimiza o erro quadrático (ou a variação intragrupo).

O algoritmo *k–means* (ou *k*–médias) é o principal e mais conhecido algoritmo de agrupamento particional. A ideia fundamental desse algoritmo é particionar o conjunto de dados em *k* grupos, considerando *k* um valor previamente determinado e os grupos sendo formados de acordo com alguma medida de similaridade.

Existem muitas versões do algoritmo *k–means*, mas a mais popular começa o processo de agrupamento inicializando um conjunto de *k* centroides, em que esses centroides são escolhidos aleatoriamente do conjunto de dados.

Em seguida, cada exemplo do conjunto de dados passa a ser associado ao grupo com o centroide mais próximo. Por fim, os centroides são recalculados e o processo se encerra quando nenhum mais centroide é alterado.

Você consegue dizer de que forma o algoritmo *k–means* pode ser utilizado no Varejo?

Algoritmos Baseados em Densidade

Os algoritmos de agrupamento baseados em densidade assumem que, no espaço de exemplos, os grupos são regiões de alta densidade de exemplos, sendo separadas por regiões com baixa densidade.

Um grupo é definido como um componente densamente conectado que cresce em qualquer direção dada pela densidade. Os métodos DENCLUE e DBSCAN são os algoritmos mais conhecidos dessa categoria.

Vamos estudar melhor o DBSCAN e conhecer as aplicações desejadas para os algoritmos de agrupamento baseados em densidade.

O *DBSCAN* (*Density Based Spatial Clustering of Applications with Noise*) é caracterizado por guiar o processo de descoberta dos grupos com base na densidade de exemplos existentes na vizinhança de um exemplar pertencente a um grupo.

Ele foi desenvolvido para encontrar agrupamentos em diferentes formatos e em bases de dados com ruído.

O algoritmo determina automaticamente a quantidade de grupos, obrigando todos os grupos a ter pelo menos um objeto de núcleo, que contém uma quantidade mínima de exemplos no seu raio de vizinhança.

O processo é iterativo e repetidamente adiciona os exemplos aos grupos até que todos sejam visitados. Os únicos exemplos que são excluídos do agrupamento são aqueles definidos como ruído como algoritmo.

Na prática, o DBSCAN admite que, para cada exemplo de um grupo, a vizinhança desse exemplo contém uma quantidade finita e limitada de exemplos, considerando um determinado raio.

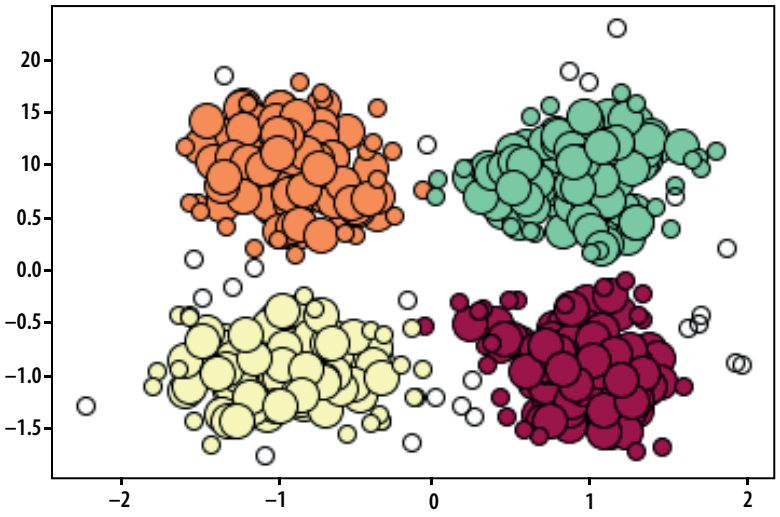


Figura 6 – Agrupamentos formados pelo DBSCAN

A Figura 6 ilustra a aplicação do método DBSCAN em uma base de dados gerada com dados aleatórios para a formação de 500 exemplos.

Nessa aplicação, construímos uma base de dados aleatórios por meio de funções geradoras de bolhas isotrópicas da Linguagem *Python*.

Você pode observar os grupos formados pelas bolhas das cores em destaque. Na cor branca, as bolhas representam os exemplos identificados como pontos ruidosos pelo algoritmo.

Existem diversas variações desse algoritmo e muitas aplicações, incluindo áreas como a Química, a Engenharia Civil e as Ciências Sociais.

A detecção de padrões e a identificação de ruídos tem sido a principal tarefa do DBSCAN e outros algoritmos semelhantes de agrupamento baseados em densidade.

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:

Vídeos

Mineração de Regras de Associação com *Python*, *Apriori* e SQL
https://youtu.be/uhWUkmdAuVI

Análise de padrão de compra com Regras de Associação
https://youtu.be/Oq_kM6M_KQ0

Leitura

Mineração de padrões frequentes em séries temporais para apoio à tomada de decisão em agrometereologia
https://bit.ly/34uosRB

Aplicação de Regras de Associação para Mineração de Dados na *Web*
https://bit.ly/3l34Wlz

Referências

CASTRO, L. N. de; FERRARI, D. G. **Introdução à mineração de dados**: conceitos básicos, algoritmos e aplicações. São Paulo: Saraiva, 2016.

FACELI, K. *et al*. **Inteligência Artificial**: Uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

SILVA, L. A. da; PERES, S. M.; BOSCARIOLI, C. **Introdução à mineração de dados**: com aplicações em R. São Paulo: Grupo GEN LTC, 2017.


Cruzeiro do Sul
Educacional