Estratégia Concursos

Aula 01

Banco do Brasil (Escriturário - Agente de Tecnologia) Passo Estratégico de Tecnologia de Informação - 2023 (Pós-Edital)

Autor:
Thiago Rodrigues Cavalcanti

10 de Janeiro de 2023

1. Aprendizagem de máquina: Fundamentos básicos; Noções de algoritmos de aprendizado não supervisionados

Sumário

ANÁLISE ESTATÍSTICA

Inicialmente, convém destacar os percentuais de incidência de todos os assuntos previstos no nosso curso – quanto maior o percentual de cobrança de um dado assunto, maior sua importância:

Assunto	Grau de incidência em concursos similares CESGRANRIO
Modelagem conceitual de dados (a abordagem entidade-relacionamento); Modelo relacional de dados (conceitos básicos, normalização);	23,40%
Linguagem SQL2008;	20,85%
5. Estrutura de dados e algoritmos: Busca sequencial e busca binária sobre arrays; Ordenação (métodos da bolha, ordenação por seleção, ordenação por inserção), lista encadeada, pilha, fila e noções sobre árvore binária.	17,87%
6. Ferramentas e Linguagens de Programação para manipulação de dados: Ansible; Java (SE 11 e EE 8); TypeScript 4.0;	14,47%
Data Warehouse (modelagem conceitual para data warehouses, dados multidimensionais);	13,19%
Python 3.9.X aplicada para IA/ML e Analytics (bibliotecas Pandas, NumPy, SciPy, Matplotlib e Scikit-learn).	4,68%
2. Banco de Dados: Banco de dados NoSQL (conceitos básicos, bancos orientados a grafos, colunas, chave/valor e documentos);	1,70%
4. Desenvolvimento Mobile: linguagens/frameworks: Java/Kotlin e Swift. React Native 0.59; Sistemas Android api 30 e iOS xCode 10.	1,70%
3. Big data: Fundamentos; Técnicas de preparação e apresentação de dados.	1,28%
1. Aprendizagem de máquina: Fundamentos básicos; Noções de algoritmos de aprendizado supervisionados e não supervisionados	0,43%
Postgre-SQL;	0,43%
Noções de processamento de linguagem natural.	0,00%
Conceitos de banco de dados e sistemas gerenciadores de bancos de dados (SGBD); MongoDB;	0,00%

ROTEIRO DE REVISÃO E PONTOS DO ASSUNTO QUE MERECEM DESTAQUE

A ideia desta seção é apresentar um roteiro para que você realize uma revisão completa do assunto e, ao mesmo tempo, destacar aspectos do conteúdo que merecem atenção.

Para revisar e ficar bem-preparado no assunto, você precisa, basicamente, seguir os seguintes tópicos:

Dicionário

Faremos uma lista de termos que são relevantes ao entendimento do assunto desta aula! Se durante sua leitura texto, você tenha alguma dúvida sobre conceitos básicos, esta parte da aula pode ajudar a esclarecer.

Aprendizado de Máquina Não Supersionado

Agrupamento hierárquico - Uma categoria de algoritmos *de agrupamento* que criam uma árvore de clusters. O agrupamento hierárquico é adequado para dados hierárquicos, como taxonomias botânicas. Existem dois tipos de algoritmos hierárquicos de clustering:

- **O agrupamento aglomerativo** primeiro atribui todos os exemplos ao seu próprio cluster, e funde iterativamente os clusters mais próximos para criar uma árvore hierárquica.
- **O agrupamento divisivo** agrupa todos os exemplos em um aglomerado e, em seguida, iterativamente divide o cluster em uma árvore hierárquica.

Aprendizagem não supervisionada - onde um modelo aprende sobre dados não rotulados, inferindo mais sobre estruturas ocultas para produzir saídas precisas e confiáveis.

Clustering - um processo de algoritmo não supervisionado de dividir pontos de dados em determinados grupos.

Redução de Recursos - é o processo de redução do número de recursos para melhorar a eficiência de uma tarefa intensiva em computação sem perder informações.

K-means - um algoritmo de aprendizagem não supervisionado usado para agrupar pontos de dados para o centroide mais próximo à distância.

Scikit-learn - uma biblioteca para os usuários python que contém ferramentas para aprendizado de máquina e modelagem estatística, como classificação, regressão, clustering e redução de dimensionalidade

Transfer Learning - um método de aprendizagem de máquina onde a aplicação de conhecimento obtido a partir de um modelo utilizado em uma tarefa, pode ser reutilizada como ponto de fundação para outra tarefa.

Supervisionado vs. Não supervisionado

O campo do aprendizado de máquina tem dois ramos principais — ***aprendizado supervisionado*** **e** ***aprendizagem não supervisionado*** — e muitos sub-ramos que fazem a ponte entre os dois.

No **aprendizado supervisionado**, o agente de inteligência artificial tem acesso a rótulos, que podem usar para melhorar seu desempenho em alguma tarefa. No problema do filtro de spam de e-mail, temos um conjunto de dados de e-mails com todo o texto dentro de cada e-mail. Também sabemos quais desses e-mails são spam ou não (os chamados ***rótulos***). Esses rótulos são muito valiosos para ajudar a IA de aprendizagem supervisionada a separar os e-mails de spam dos demais.

Em **aprendizado não supervisionado**, os rótulos **não** estão disponíveis. Portanto, a tarefa do agente de IA não está bem definida, e o desempenho não pode ser tão claramente medido. Considere o problema do filtro de spam de e-mail — desta vez **sem rótulos**. Agora, o agente de IA tentará entender a estrutura subjacente dos e-mails, separando o banco de dados de e-mails em diferentes grupos, de modo que os e-mails dentro de um grupo são semelhantes entre si, mas diferentes de e-mails em outros grupos.

Este **problema de aprendizagem não supervisionado** é **menos claramente definido** do que o problema de aprendizagem supervisionado e mais difícil para o agente de IA resolver. Mas, se bem tratada, a solução é mais poderosa.

Eis o porquê: **a IA de aprendizagem não supervisionada pode encontrar vários grupos que mais tarde marca como sendo "spam"** — mas a IA também pode encontrar grupos que mais tarde sejam marcados como sendo "importantes" ou categorizados como "família", "profissional", "notícias", "compras", etc. Em outras palavras, **como o problema não tem uma tarefa estritamente definida, o agente de IA pode encontrar padrões interessantes acima e além do que estávamos procurando inicialmente**.

Além disso, este sistema não supervisionado é melhor do que o sistema supervisionado para encontrar novos padrões em dados futuros, tornando a solução não supervisionada mais ágil em uma base contínua. Este é o poder do aprendizado não supervisionado que vamos tratar na nossa revisão hoje!

Os pontos fortes e fracos do aprendizado não supervisionado

O aprendizado supervisionado irá superar o aprendizado não supervisionado em tarefas estritamente definidas para as quais temos padrões bem definidos que não mudam muito ao longo do tempo e conjuntos de dados são suficientemente grandes e prontamente disponíveis e rotulados. No entanto, para problemas onde os padrões são desconhecidos ou em constante mudança ou

para os quais não temos conjuntos de dados grandes e rotulados, o aprendizado **não supervisionado realmente brilha**.

Em vez de ser guiado por rótulos, a aprendizagem não supervisionada funciona aprendendo **a estrutura subjacente dos dados em que treinou**. Ele faz isso tentando representar os dados que treina com um **conjunto de parâmetros** que é significativamente menor do que o número de exemplos disponíveis no conjunto de dados. Ao realizar esse aprendizado de representação, o aprendizado não supervisionado é capaz de **identificar padrões distintos no conjunto de dados**.

No exemplo do conjunto de dados de imagem (sem rótulos), a IA de aprendizagem não supervisionada pode ser capaz de **identificar e agrupar imagens com base no quão semelhantes elas são** umas com as outras e quão diferentes elas são das outras. Por exemplo, todas as imagens que parecem cadeiras serão agrupadas, todas as imagens que parecem cães serão agrupadas, etc.

É claro que a própria IA de aprendizagem não supervisionada não pode rotular esses grupos como "cadeiras" ou "cães", mas agora que imagens semelhantes são agrupadas, os humanos têm uma tarefa de rotulagem muito mais simples. Em vez de rotular milhões de imagens à mão, os humanos podem rotular manualmente todos os grupos distintos, e os rótulos se aplicarão a todos os membros de cada grupo.

Após o treinamento inicial, se a IA de aprendizagem não supervisionada encontrar imagens que não pertencem a nenhum dos grupos rotulados, a IA criará grupos separados para as imagens não classificadas, cabendo a um humano a rotular os novos grupos de imagens ainda não rotulados.

O aprendizado não supervisionado torna os **problemas anteriormente intratáveis mais solucionáveis e é muito mais ágil em encontrar padrões ocultos**, tanto nos dados históricos que estão disponíveis para treinamento quanto em dados futuros. Além disso, agora temos uma abordagem de IA para os enormes conjuntos de dados não rotulados que existem no mundo.

Embora a aprendizagem não supervisionada seja menos utilizada do que a aprendizagem supervisionada na resolução de problemas específicos e pouco definidos, ela é melhor para abordar problemas mais abertos e generalizar esse conhecimento. Igualmente importante, o aprendizado não supervisionado pode resolver muitos dos problemas comuns que os cientistas de dados encontram ao construir soluções de aprendizado de máquina.

Usando aprendizado não supervisionado para melhorar soluções de aprendizado de máquina

Os sucessos recentes no aprendizado de máquina têm sido impulsionados pela disponibilidade de muitos dados, avanços em hardware e recursos baseados em nuvem e avanços em algoritmos de aprendizado de máquina. Mas esses sucessos têm sido em problemas de IA mais estreitos, como classificação de imagem, visão computacional, reconhecimento de fala, processamento de linguagem natural e tradução automática.

Para resolver problemas mais ambiciosos de IA, **precisamos desbloquear o valor do aprendizado não supervisionado**. Vamos explorar os desafios mais comuns que os cientistas de dados enfrentam ao construir soluções e como o aprendizado não supervisionado pode ajudar.

Dados rotulados insuficientes

> *Acho que a IA é semelhante à construção de um foguete. Você precisa de um motor enorme e muito combustível. Se você tem um motor grande e uma pequena quantidade de combustível, você não vai chegar à órbita. Se você tem um motor minúsculo e uma tonelada de combustível, você não pode nem decolar. Para construir um foguete você precisa de um motor enorme e muito combustível.*
>
> *Andrew Ng*

Se o aprendizado de máquina fosse um foguete, os dados seriam o combustível — sem muitos e muitos dados, o foguete não pode voar. Mas nem todos os dados são criados iguais. Para usar algoritmos supervisionados, precisamos de muitos dados rotulados, o que é difícil e caro de gerar.

Com o aprendizado não supervisionado, podemos automaticamente rotular exemplos sem rótulo. Funciona assim: agrupamos todos os exemplos e, em seguida, aplicaríamos os rótulos de exemplos rotulados aos não rotulados dentro do mesmo cluster. Exemplos não rotulados receberiam o rótulo dos exemplos que são mais semelhantes.

Overfitting

Se o algoritmo de aprendizagem de máquina aprender uma função excessivamente complexa com base nos dados de treinamento, ele poderá ter um desempenho muito ruim em instâncias nunca vistas de conjuntos de holdout, como o conjunto de validação ou conjunto de testes. Neste caso, o algoritmo se entranhou demais nos dados de treinamento — extraindo muito do ruído nos dados — e tem um erro de generalização grande. Em outras palavras, o algoritmo está memorizando os dados de treinamento em vez de aprender a generalizar o conhecimento com base nele.

Para lidar com isso, podemos introduzir o aprendizado não supervisionado **como *um regularizador***. *A regularização* é um processo usado para **reduzir a complexidade de um algoritmo de aprendizagem de máquina**, ajudando-o a capturar o **sinal** nos dados sem se ajustar muito ao **ruído**. A pré-formação do conjunto de dados não supervisionada é uma dessas formas de regularização. Em vez de alimentar os dados de entrada originais diretamente em um algoritmo de aprendizagem supervisionado, podemos alimentar uma nova representação dos dados de entrada originais que geramos.

Essa nova representação captura a essência dos dados originais — a verdadeira estrutura subjacente — enquanto perde um pouco do ruído menos representativo ao longo do caminho. Quando alimentamos essa nova representação no algoritmo de aprendizagem supervisionado, ele tem menos ruído e captura mais do sinal, melhorando seu erro de generalização.

Maldição da dimensionalidade

Mesmo com os avanços no poder computacional, é difícil para os algoritmos de aprendizagem de máquina gerenciarem a grande quantidade de dados. Em geral, adicionar mais instâncias não é

muito problemático porque podemos fazer um paralelo com operações usando soluções modernas, como o Spark. No entanto, quanto mais características temos, mais difícil se torna o treinamento.

Em um espaço muito dimensional, algoritmos supervisionados precisam aprender a separar pontos e construir uma aproximação de função para tomar boas decisões. Quando os recursos são muito numerosos, essa pesquisa se torna muito cara, tanto do ponto de vista do tempo quanto da computação. Em alguns casos, pode ser impossível encontrar uma boa solução rápido o suficiente.

Este problema é conhecido como a *maldição da dimensionalidade*, e o aprendizado não supervisionado é adequado para ajudar a gerenciar isso. Com a **redução da dimensionalidade**, podemos encontrar as características mais importantes no conjunto de recursos originais, reduzir o número de dimensões para um número mais gerenciável, ao mesmo tempo em que perde muito poucas informações importantes no processo e, em seguida, aplicar algoritmos supervisionados para realizar com mais eficiência a busca por uma boa aproximação de função.

Engenharia de recursos

A engenharia de recursos é uma das tarefas mais vitais que os cientistas de dados realizam. Sem as características certas, o algoritmo de aprendizado de máquina não será capaz de separar pontos no espaço bem o suficiente para tomar boas decisões sobre exemplos nunca vistos. No entanto, a engenharia de recursos é tipicamente muito trabalhosa; requer que os humanos criem criativamente os tipos certos de recursos. Em vez disso, podemos usar o aprendizado de representação de algoritmos de aprendizagem não supervisionados para aprender automaticamente os tipos certos de representações de recursos para ajudar a resolver a tarefa em questão.

Outliers

A qualidade dos dados também é muito importante. Se os algoritmos de aprendizagem de máquina treinarem em outliers raros e distorcidos, seu erro de generalização será menor do que se eles ignoraram ou abordaram os outliers separadamente. Com um aprendizado não supervisionado, podemos realizar uma detecção outlier usando a redução de dimensionalidade e criar uma solução especificamente para os outliers e, separadamente, uma solução para os dados normais.

Data drift

Os modelos de aprendizado de máquina também precisam estar atentos ao data drift. Se os dados em que o modelo está fazendo previsões diferem estatisticamente dos dados em que o modelo treinou, o modelo pode precisar retreinar dados que são mais representativos dos dados atuais. Se o modelo não treinar ou não reconhecer o drift, a qualidade de previsão do modelo sobre os dados atuais sofrerá.

Ao construir distribuições de probabilidades usando aprendizados não supervisionados, podemos avaliar o quão diferentes os dados atuais são dos dados do conjunto de treinamento — se os dois forem diferentes o suficiente, podemos acionar automaticamente uma retreinamento.

Um olhar mais atento para algoritmos não supervisionados

Agora voltaremos nossa atenção para problemas onde não temos rótulos. Em vez de tentar fazer previsões, algoritmos de aprendizagem não supervisionados tentarão aprender **a estrutura subjacente dos dados**.

Redução de Dimensionalidade

Uma família de algoritmos — *conhecidos como algoritmos de redução de dimensionalidade* — **projeta os dados originais de entrada de alta dimensão para um espaço de baixa dimensão, filtrando as características não tão relevantes e mantendo o máximo possível das interessantes**. A redução da dimensionalidade permite que a IA de aprendizagem não supervisionada identifique mais efetivamente padrões e resolva de forma mais eficiente problemas de grande escala e computacionalmente caros (muitas vezes envolvendo imagens, vídeo, fala e texto).

Projeção linear

Existem dois ramos principais da dimensionalidade: **projeção linear e redução da dimensionalidade não linear**. Começaremos com projeção linear primeiro.

Análise de componentes principais (PCA)

Uma abordagem para aprender a estrutura subjacente dos dados é **identificar quais características fora do conjunto completo de recursos são mais importantes para explicar a variabilidade entre as instâncias dos dados**. Nem todos os recursos são iguais — para alguns recursos, os valores no conjunto de dados não variam muito, e esses recursos são menos úteis para explicar o conjunto de dados. Para outros recursos, os valores podem variar consideravelmente — esses recursos valem a pena explorar com mais detalhes, pois serão melhores em ajudar o modelo que projetamos a separar os dados.

No *PCA*, o algoritmo encontra **uma representação de baixa dimensão dos dados**, mantendo o máximo possível da variação. O número de dimensões que restam é consideravelmente menor do que o número de dimensões do conjunto de dados completo (ou seja, o número de características totais). Perdemos parte da variância movendo-se para este espaço de baixa dimensão, mas a estrutura subjacente dos dados é mais fácil de ser identificada, permitindo-nos executar tarefas como agrupamento de forma mais eficiente.

Existem várias variantes de PCA. Estes incluem variantes de mini-lote, *PCA incremental*, variantes não lineares, como *o KERNEL PCA*, e variantes esparsas, como *PCA esparso*.

Decomposição de valor singular (SVD)

Outra abordagem para aprender a estrutura subjacente dos dados é **reduzir a classificação da matriz original de características para uma classificação menor**, de tal forma que a matriz original pode ser recriada usando uma combinação linear de alguns dos vetores na matriz de classificação menor. Isso é conhecido como *SVD*. Para gerar a matriz de classificação menor, o

SVD mantém os vetores da matriz original que têm mais informações (ou seja, o maior valor singular). A matriz de classificação menor captura os elementos mais importantes do espaço original.

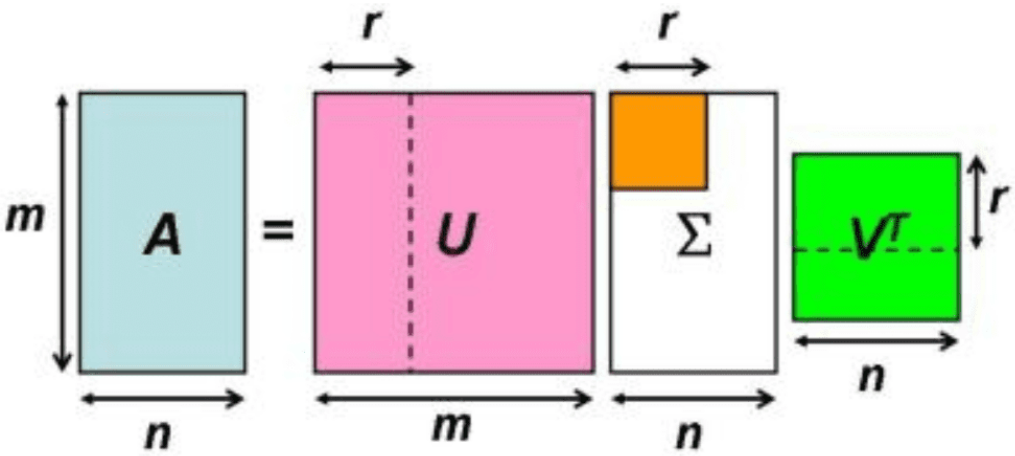


Figura 1 - Decomposição de valor singular

Projeção aleatória

Um algoritmo de redução de dimensionalidade semelhante envolve projetar pontos de um espaço de alta dimensão para um espaço de dimensões muito mais baixas de tal forma que a escala de distâncias entre os pontos seja preservada. Podemos usar uma *matriz gaussiana aleatória* ou uma *matriz esparsa aleatória* para conseguir isso.

Aprendizado múltiplo

Tanto o PCA quanto a projeção aleatória dependem da projeção linear dos dados de um espaço de alta dimensão para um espaço de baixa dimensão. **Em vez de uma projeção linear, pode ser melhor realizar uma transformação não linear dos dados — isso é conhecido como *aprendizado múltiplo* ou *redução de dimensionalidade não linear*.**

Isomap

Isomap é um tipo de abordagem de aprendizagem múltiplas. Este algoritmo aprende a geometria intrínseca do coletor de dados, estimando a distância *geodésica* ou *curva* entre cada ponto e seus vizinhos, em vez da distância euclidiana. Isomap usa isso para incorporar o espaço original de alta dimensão a um de baixa dimensão.

t-distributed Stochastic Neighbor Embedding (t-SNE)

Outra redução de dimensionalidade não linear — conhecida como *t-SNE* — incorpora dados de alta dimensão em um espaço de apenas duas ou três dimensões, permitindo que os dados transformados sejam visualizados. Neste espaço bi ou tridimensional, instâncias semelhantes são modeladas mais próximas e instâncias diferentes são modeladas mais longe.

Aprendizado de dicionário

Uma abordagem conhecida como *aprendizagem de dicionário* envolve aprender a representação dos dados subjacentes. Esses elementos representativos são vetores simples, e cada instância no conjunto de dados é representada como vetor de peso e pode ser reconstruída como uma soma ponderada dos elementos representativos. Os elementos representativos que esse aprendizado não supervisionado gera são chamados de *dicionário*.

Ao criar tal dicionário, este algoritmo é capaz de identificar eficientemente os elementos representativos mais salientes do espaço de características originais — estes são os que têm mais pesos não zero. Os elementos representativos menos importantes terão poucos pesos não-zero. Assim como no PCA, o aprendizado do dicionário é excelente para aprender a estrutura subjacente dos dados, o que será útil na separação dos dados e na identificação de padrões interessantes.

Análise de componentes independentes (ICA)

Um problema comum com dados não rotulados é que existem muitos sinais independentes incorporados nas características que nos são dados. Usando *a análise de componentes independentes (ICA)* podemos separar esses sinais misturados em seus componentes individuais. Depois que a separação estiver completa, podemos reconstruir qualquer um dos recursos originais, adicionando alguma combinação dos componentes individuais que geramos. O ICA é comumente usado em tarefas de processamento de sinais (por exemplo, para identificar as vozes individuais em um clipe de áudio).

Alocação de Dirichlet latente (latent dirichlet allocation - LDA)

O aprendizado não supervisionado também pode explicar um conjunto de dados aprendendo porque algumas partes do conjunto de dados são semelhantes umas às outras. Isso requer o aprendizado de elementos não observados dentro do conjunto de dados — uma abordagem conhecida como *alocação de Dirichlet latente (LDA).* Por exemplo, considere um documento de texto com muitas, muitas palavras. Estas palavras dentro de um documento não são puramente aleatórias; em vez disso, eles exibem alguma estrutura.

Esta estrutura pode ser modelada como elementos não observados conhecidos como tópicos. Após o treinamento, o LDA é capaz de explicar um determinado documento com um pequeno conjunto de tópicos, onde para cada tópico há um pequeno conjunto de palavras frequentemente usadas. Esta é a estrutura oculta que a LDA é capaz de capturar, ajudando-nos a explicar melhor um corpus de texto não estruturado anteriormente.

OBSERVAÇÃO

A redução da dimensionalidade reduz o conjunto original de recursos para um conjunto menor de apenas as características mais importantes. A partir daqui, podemos executar outros algoritmos de aprendizagem não supervisionados neste conjunto menor de recursos para encontrar padrões interessantes nos dados (ver a próxima seção sobre clustering), ou, se tivermos rótulos, podemos acelerar o ciclo de treinamento de algoritmos de aprendizagem supervisionados alimentando-os nesta matriz menor de recursos em vez de usar a matriz de recursos originais.

Clustering

Uma vez que tenhamos reduzido o conjunto de recursos originais para um conjunto menor e mais gerenciável, podemos encontrar padrões interessantes agrupando instâncias semelhantes de dados. Isso é conhecido como clustering e pode ser realizado com uma variedade de algoritmos de aprendizagem não supervisionados e ser usado para aplicações do mundo real, como segmentação de mercado.

k-means

Para agrupar bem, precisamos **identificar grupos distintos de modo que as instâncias dentro de um grupo sejam semelhantes entre si**, mas diferentes das instâncias de outros grupos. Um desses algoritmos é *o agrupamento k-means*. Com este algoritmo, **especificamos o número de clusters *k* desejados**, e o algoritmo atribuirá cada instância a exatamente um desses clusters *k*. O algoritmo otimiza o agrupamento minimizando a *variação dentro do cluster* (também conhecida como *inércia*) de modo que a soma das variações dentro do cluster em todos os clusters *k* seja o menor possível.

Para acelerar este processo de agrupamento, *k-means* atribui aleatoriamente k pontos para serem a referência inicial de cada grupo, em seguida, começa a atribuir as demais observações a cada grupo considerando a menor distância euclidiana entre cada observação e o ponto central do cluster, ou *centroide*. Quando terminar de executar esse ciclo, o algoritmo calcula o centro de cada cluster, atribui esse valor ao centroide e volta a classificar todos os elementos novamente. A condição de parada é dada pela convergência do centróide com o ponto médio do grupo ou após um número especificado de rodadas. Como resultado, diferentes execuções de *k-means* — cada uma com um início randomizado — resultarão em atribuições de agrupamento ligeiramente diferentes das observações. A partir dessas diferentes execuções, podemos escolher aquela que tem a melhor separação, definida como a menor soma total de variações dentro do cluster em todos os clusters *k*.

Agrupamento hierárquico

Uma abordagem alternativa de agrupamento — que não exige compromisso com um determinado número de clusters — é conhecida como *agrupamento hierárquico*. Uma versão de agrupamento hierárquico chamado *clustering agglomerative* usa um método de agrupamento baseado em árvores, e constrói o que é chamado *de dendrograma*. Um dendrograma pode ser retratado graficamente como uma árvore de cabeça para baixo, onde as folhas estão na parte inferior e o tronco da árvore está no topo.

As folhas na parte inferior são instâncias individuais no conjunto de dados. O agrupamento hierárquico então une as folhas — à medida que nos movemos verticalmente para cima da árvore de cabeça para baixo — com base no quão semelhantes elas são umas às outras. As instâncias (ou grupos de instâncias) mais semelhantes entre si são juntadas mais cedo, enquanto as instâncias que não são tão semelhantes são juntadas mais tarde. Com esse processo iterativo, todas as instâncias acabam sendo ligadas juntos formando o tronco único da árvore.

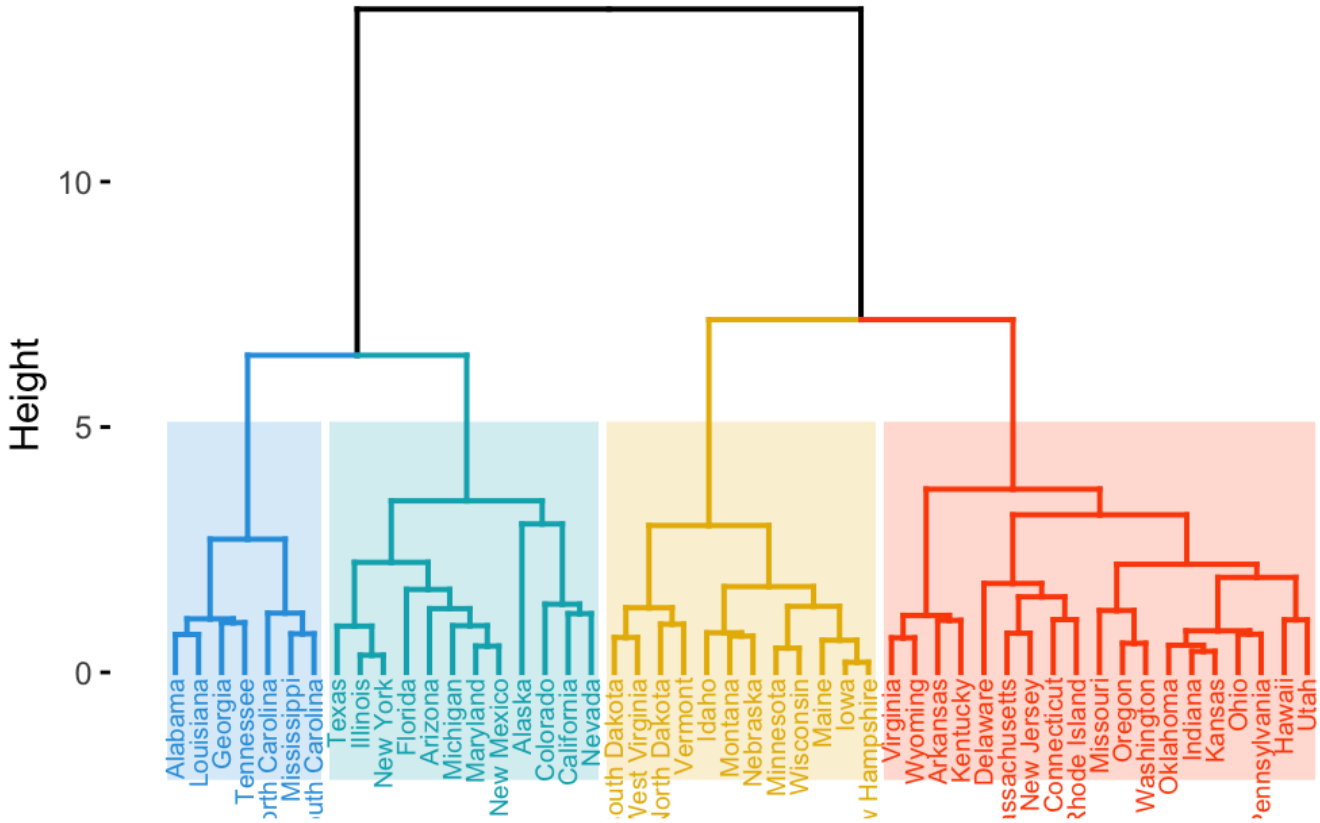


Figura 2 - Exemplo de dendrograma

Esta representação vertical é muito útil. Uma vez que o algoritmo hierárquico de clustering tenha terminado, podemos ver o dendrograma e determinar onde queremos cortar a árvore — quanto mais baixo cortamos, mais ramos individuais nos resta (ou seja, mais clusters). Se quisermos menos aglomerados, podemos cortar mais alto no dendrograma, mais perto do tronco único no topo desta árvore de cabeça para baixo. A colocação deste corte horizontal é semelhante à escolha do número de clusters *k* no algoritmo de clustering *k-means*.

DBSCAN

Um algoritmo de clustering ainda mais poderoso (**baseado na densidade de pontos**) é conhecido como *DBSCAN* (clustering espacial baseado em densidade de aplicações com ruído). Dadas todas as instâncias que temos no espaço, a DBSCAN agrupará aqueles que estão próximas, onde próximos são definidos como um número mínimo de instâncias que devem existir a uma certa distância. Especificamos tanto o número mínimo de instâncias necessárias quanto a distância.

Se uma instância estiver dentro desta distância especificada, ela será agrupada com o cluster ao qual está mais densamente localizada. Qualquer instância que não esteja dentro desta distância especificada de algum cluster é rotulada como outlier.

Ao contrário dos *k-means*, não precisamos especificar previamente o número de clusters. Também podemos ter aglomerados arbitrariamente moldados. O DBSCAN é muito menos propenso à distorção tipicamente causada por outliers nos dados.

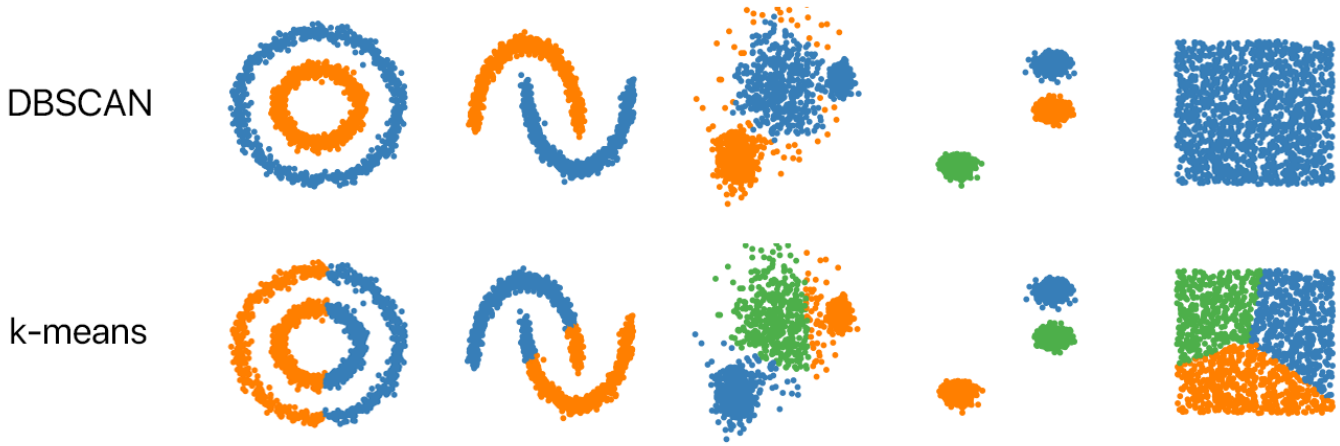


Figura 3 - Comparação de DBSCAN e K-MEANS

Extração de recursos

Com um aprendizado não supervisionado, podemos aprender **novas representações das características originais dos dados** — um campo conhecido como ***extração de recursos***. A extração de recursos pode ser usada para reduzir o número de características originais a um subconjunto menor, realizando efetivamente a redução da dimensionalidade. Mas a extração de recursos **também pode gerar novas representações de recursos para ajudar a melhorar o desempenho em problemas de aprendizagem supervisionados.**

Autoencoders

Para gerar novas representações de recursos, podemos usar uma rede neural não recorrente para executar o aprendizado de representação, onde o número de nós na camada de saída corresponde ao número de nós na camada de entrada. Esta rede neural é conhecida como um *autoencoder* e efetivamente reconstrói as características originais, aprendendo uma nova representação usando as camadas ocultas.

Cada camada oculta do autoencoder aprende uma representação das características originais, e camadas subsequentes se baseiam na representação aprendida pelas camadas anteriores. Camada por camada, o autoencoder aprende representações cada vez mais complicadas das mais simples.

A camada de saída é a representação final recém-aprendida das características originais. Essa representação aprendida pode então ser usada como um insumo em um modelo de aprendizagem supervisionada com o objetivo de melhorar o erro de generalização.

Aprendizado profundo não supervisionado

O aprendizado não supervisionado desempenha muitas funções importantes no campo do ***deep learning***. Este campo é conhecido como *aprendizado profundo não supervisionado*.

Até muito recentemente, o treinamento de redes neurais profundas era computacionalmente intratável. Nessas redes neurais, as camadas ocultas aprendem representações internas para ajudar a resolver o problema em questão. As representações melhoram com base na forma como a rede neural usa o *gradiente da função de erro* em cada iteração de treinamento para atualizar os pesos dos vários nós.

Essas atualizações são computacionalmente caras, e dois grandes tipos de problemas podem ocorrer no processo. Primeiro, o gradiente da função de erro pode se tornar muito pequeno, e, uma vez que *a retropagação* depende da multiplicação desses pequenos pesos juntos, os pesos da rede podem se atualizar muito lentamente ou não, impedindo o treinamento adequado da rede. Isso é conhecido como o *problema do vanishing gradient*.

Por outro lado, a outra questão é que o gradiente da função de erro pode se tornar muito grande; com a retropropagação, os pesos em toda a rede podem se atualizar em incrementos enormes, tornando o treinamento da rede muito instável. Isso é conhecido como o *problema do gradiente explosivo (exploding gradient problem)*.

Pré-treinamento não supervisionado

Para enfrentar essas dificuldades no treinamento de redes neurais muito profundas e multicamadas, pesquisadores de aprendizado de máquina treinam redes neurais em múltiplos estágios sucessivos, onde cada estágio envolve uma rede neural rasa. A saída de uma rede rasa é então usada como entrada da próxima rede neural. Normalmente, a primeira rede neural rasa neste fluxo envolve uma rede neural não supervisionada, mas as redes posteriores são supervisionadas.

Esta porção não supervisionada é conhecida como ***pré-treino ganancioso e não supervisionado***. Em 2006, Geoffrey Hinton demonstrou a aplicação bem-sucedida de pré-treinamento não supervisionado para inicializar o treinamento do pipeline de rede neural mais profundos, dando início à atual revolução do aprendizado profundo. O pré-treino não supervisionado permite que a IA capture uma representação melhorada dos dados de entrada originais, que a parte supervisionada então aproveita para resolver a tarefa específica em questão.

Essa abordagem é chamada de "gananciosa" porque cada parte da rede neural é treinada independentemente, não em conjunto. "Em termos de camada" refere-se às camadas da rede. Na maioria das redes neurais modernas, o pré-treino geralmente não é necessário. Em vez disso, todas as camadas são treinadas em conjunto usando a retropagação. Os principais avanços do computador tornaram o problema do gradiente de desaparecimento e o problema do gradiente explodindo muito mais gerenciáveis.

O pré-treino não supervisionado não só facilita a solução de problemas supervisionados, mas também facilita o *aprendizado da transferência*. O aprendizado de transferência envolve o uso de algoritmos de aprendizado de máquina para armazenar conhecimento adquirido com a resolução de uma tarefa para resolver outra tarefa relacionada muito mais rapidamente e com consideravelmente menos dados.

Máquinas Boltzmann Restritas (RBM)

Um exemplo aplicado de pré-treino não supervisionado é a *máquina Boltzmann restrita (RBM),* uma rede neural rasa de duas camadas. A primeira camada é a camada de entrada, e a segunda camada é a camada oculta. Cada nó está conectado a cada nó na outra camada, mas os nós não estão conectados a nós da mesma camada — é aí que ocorre a restrição.

As RBMs podem executar tarefas não supervisionadas, como redução de dimensionalidade e extração de recursos e fornecer pré-treinamento não supervisionado útil como parte de soluções de aprendizagem supervisionadas. Os RBMs são semelhantes aos autoencoders, mas diferem de algumas maneiras importantes. Por exemplo, os autoencoders têm uma camada de saída, enquanto os RBMs não.

Redes de crenças profundas

Os RBMs podem ser ligados para formar um pipeline de rede neural multiestágio conhecido como uma *rede de crenças profundas (DBN).* A camada oculta de cada RBM é usada como entrada para o próximo RBM. Em outras palavras, cada RBM gera uma representação dos dados que o próximo RBM então constrói. Ao vincular sucessivamente esse tipo de aprendizado de representação, a rede de crenças profundas é capaz de aprender representações mais complicadas que muitas vezes são usadas como *detectores de recursos*.

Redes contraditórias generativas (Generative adversarial networks)

Um grande avanço no aprendizado profundo não supervisionado tem sido o advento das *redes contraditórias geradoras (GANs)* introduzidas por Ian Goodfellow e seus colegas pesquisadores na Universidade de Montreal em 2014. Os GANs têm muitas aplicações; por exemplo, podemos usar GANs para criar dados sintéticos quase realistas, como imagens e fala, ou executar a detecção de anomalias.

Em GANs, temos duas redes neurais. Uma rede — conhecida como **geradora** — gera dados com base em uma distribuição de dados modelo usando amostras de dados reais que recebeu. A outra rede — conhecida como ***discriminator*** — discrimina entre os dados criados pelo gerador e os dados da distribuição que são verdadeiros.

Como uma simples analogia, o gerador é o falsificador, e o discriminador é a polícia tentando identificar a falsificação. As duas redes estão bloqueadas em um jogo de soma zero. O gerador está tentando enganar o discriminador a pensar que os dados sintéticos vêm da distribuição de dados verdadeira, e o discriminador está tentando chamar os dados sintéticos de falsos.

As GANs são algoritmos de aprendizagem não supervisionados porque o gerador pode aprender a estrutura subjacente da distribuição de dados verdadeira, mesmo quando não há rótulos. Os GANs aprendem a estrutura subjacente nos dados através do processo de treinamento e capturam eficientemente a estrutura usando um pequeno número gerenciável de parâmetros.

Esse processo é semelhante ao aprendizado de representação que ocorre no deep learning. Cada camada oculta na rede neutra de um gerador captura uma representação dos dados subjacentes — começando de forma muito simples — e as camadas subsequentes captam representações mais complicadas, baseando-se nas camadas anteriores mais simples.

Usando todas essas camadas juntas, o gerador aprende a estrutura subjacente dos dados e, usando o que aprendeu, o gerador tenta criar dados sintéticos quase idênticos à distribuição de dados verdadeira. Se o gerador capturou a essência da distribuição de dados verdadeira, os dados sintéticos parecerão reais.

Problemas de dados sequenciais usando aprendizado não supervisionado

O aprendizado não supervisionado também pode lidar com dados sequenciais, como dados de séries temporais. Uma dessas abordagens envolve aprender os estados ocultos de um *modelo Markov*. No *modelo simples de Markov*, os estados são plenamente observados e mudam estocasticamente (em outras palavras, aleatoriamente). Os estados futuros dependem apenas do estado atual e não dependem de estados anteriores.

Em um *modelo Markov oculto*, os estados são apenas parcialmente observáveis, mas, como com modelos simples de Markov, as saídas desses estados parcialmente observáveis são totalmente observáveis. Uma vez que as observações que temos são insuficientes para determinar completamente o estado, precisamos de aprendizado não supervisionado para ajudar a descobrir esses estados ocultos plenamente.

Algoritmos modelo Markov ocultos envolvem aprender o provável próximo estado, dado o que sabemos sobre a sequência de estados anteriormente observáveis e saídas totalmente observáveis. Esses algoritmos tiveram grandes aplicações comerciais em problemas de dados sequenciais envolvendo fala, texto e séries temporais.

TÉCNICAS DE AGRUPAMENTO, REDUÇÃO DE DIMENSIONALIDADE, TÉCNICAS DE ASSOCIAÇÃO

Técnicas de agrupamento

O que é análise de cluster?

A análise de cluster ou simplesmente **clustering** é o processo de particionar um conjunto de objetos de dados (ou observações) em subconjuntos. Cada subconjunto é um **cluster**, de modo que os objetos em um cluster são semelhantes uns aos outros, embora diferentes dos objetos em outros clusters. O conjunto de clusters resultante de uma análise de cluster pode ser referido como **clustering**. Nesse contexto, **diferentes métodos de agrupamento** podem gerar diferentes agrupamentos no mesmo conjunto de dados. O particionamento não é executado por humanos, mas pelo algoritmo de agrupamento. Consequentemente, o armazenamento em cluster é útil porque pode levar à **descoberta de grupos anteriormente desconhecidos nos dados**.

A análise de cluster tem sido amplamente usada em muitos aplicativos, como business intelligence, reconhecimento de padrões de imagem, pesquisa na Web, biologia e segurança. Em business intelligence, o clustering pode ser usado para organizar um grande número de clientes em grupos, onde os clientes dentro de um grupo compartilham fortes características semelhantes. Isso facilita

o desenvolvimento de estratégias de negócios para uma gestão aprimorada do relacionamento com o cliente. Além disso, considere uma empresa de consultoria com muitos projetos. Para melhorar o gerenciamento do projeto, o agrupamento pode ser aplicado para dividir os projetos em categorias com base na similaridade, para que a auditoria e o diagnóstico do projeto (para melhorar a entrega e os resultados do projeto) possam ser realizados de forma eficaz.

Como um ramo da estatística, a análise de cluster tem sido amplamente estudada, com foco principal na análise de cluster baseada na distância. Ferramentas de análise de cluster baseadas em k-means, k-medoids e em vários outros métodos também foram incorporadas a muitos pacotes ou sistemas de software de análise estatística.

No aprendizado de máquina, o agrupamento é conhecido como **aprendizado não supervisionado** porque as informações do rótulo da classe não estão presentes. Por este motivo, o agrupamento é uma forma de **aprendizagem por observação**, em vez de *aprender por exemplos como na classificação*. Na mineração de dados, os esforços têm se concentrado em encontrar métodos para análise de cluster eficiente e eficaz em *grandes bancos de dados*.

Podemos analisar as regras de associação de acordo com algumas perspectivas: focando na ***escalabilidade*** dos métodos de agrupamento, na eficácia dos métodos para agrupar ***formas*** *complexas* (por exemplo, não convexos) e ***tipos de dados*** (por exemplo, texto, gráficos e imagens), na técnicas de agrupamento de *alta dimensão* (por exemplo, objetos de agrupamento com milhares de recursos) e métodos para agrupar *dados numéricos e nominais combinados* em grandes bancos de dados.

Requisitos para análise de cluster

O agrupamento é um campo de pesquisa desafiador. Nesta seção, você aprenderá sobre os requisitos de armazenamento em cluster como uma ferramenta de mineração de dados, bem como os aspectos que podem ser usados para comparar métodos de armazenamento em cluster. A seguir estão os requisitos típicos de cluster em mineração de dados.

- **Escalabilidade**: muitos algoritmos de agrupamento funcionam bem em pequenos conjuntos de dados contendo menos de várias centenas de objetos de dados; entretanto, um grande banco de dados pode conter milhões ou até bilhões de objetos, especialmente em cenários que envolve pesquisa na web. O agrupamento em apenas uma amostra de um determinado conjunto de dados grande pode levar a resultados tendenciosos. Portanto, algoritmos de clustering altamente escalonáveis são necessários.

- **Capacidade de lidar com diferentes tipos de atributos**: muitos algoritmos são projetados para agrupar dados numéricos (baseados em intervalos). No entanto, os aplicativos podem exigir o armazenamento em cluster de outros tipos de dados, como dados binários, nominais (categóricos) e ordinais ou combinações desses tipos de dados. Recentemente, mais e mais

aplicativos precisam de técnicas de agrupamento para tipos de dados complexos, como gráficos, sequências, imagens e documentos.

- **Descoberta de clusters com forma arbitrária**: muitos algoritmos de clustering determinam clusters com base em medidas de distância euclidiana ou de Manhattan. Algoritmos baseados em tais medidas de distância tendem a encontrar aglomerados esféricos com tamanho e densidade semelhantes. No entanto, um cluster pode ter qualquer forma. Considere os sensores, por exemplo, que muitas vezes são implantados para vigilância ambiental. A análise de cluster nas leituras do sensor pode detectar fenômenos interessantes. Podemos querer usar o agrupamento para encontrar a fronteira de um incêndio florestal, que geralmente não é esférico. É importante desenvolver algoritmos que possam detectar clusters de forma arbitrária.

- **Requisitos de conhecimento do domínio para determinar os parâmetros de entrada**: muitos algoritmos de agrupamento exigem que os usuários forneçam conhecimento do domínio na forma de parâmetros de entrada, como o número desejado de clusters. Consequentemente, os resultados do agrupamento podem ser sensíveis a tais parâmetros (chamaremos de hiperparâmetros mais a frente). Os parâmetros costumam ser difíceis de determinar, especialmente para conjuntos de dados de alta dimensionalidade e onde os usuários ainda não compreendem profundamente seus dados.

- **Capacidade de lidar com dados ruidosos**: a maioria dos conjuntos de dados do mundo real contém outliers e/ou dados ausentes, desconhecidos ou errôneos. As leituras do sensor, por exemplo, costumam ser ruidosas - algumas leituras podem ser imprecisas devido aos mecanismos de detecção e algumas leituras podem ser errôneas devido a interferências de objetos transitórios. Os algoritmos de clustering podem ser sensíveis a esse ruído e podem produzir clusters de baixa qualidade. Portanto, precisamos de métodos de agrupamento que sejam robustos a ruídos.

- **Clustering incremental e insensibilidade à ordem de entrada**: em muitos aplicativos, atualizações incrementais (representando dados mais recentes) podem chegar a qualquer momento. Alguns algoritmos de clustering não podem incorporar atualizações incrementais em estruturas de clustering existentes e, em vez disso, precisam recalcular um novo clustering do zero. Os algoritmos de agrupamento também podem ser sensíveis à ordem dos dados de

entrada. Ou seja, dado um conjunto de objetos de dados, os algoritmos de agrupamento podem retornar agrupamentos dramaticamente diferentes dependendo da ordem (quantidade) em que os objetos são apresentados. São necessários algoritmos de clustering incremental e algoritmos insensíveis à ordem de entrada.

- **Capacidade de armazenamento em cluster de dados de alta dimensionalidade**: Um conjunto de dados pode conter várias dimensões ou atributos. Ao agrupar documentos, por exemplo, cada palavra-chave pode ser considerada como uma dimensão e geralmente há milhares de palavras-chave. A maioria dos algoritmos de agrupamento são bons no tratamento de dados de baixa dimensão, como conjuntos de dados envolvendo apenas duas ou três dimensões. Encontrar clusters de objetos de dados em um espaço de alta dimensão é um desafio, especialmente considerando que esses dados podem ser muito esparsos e altamente distorcidos.

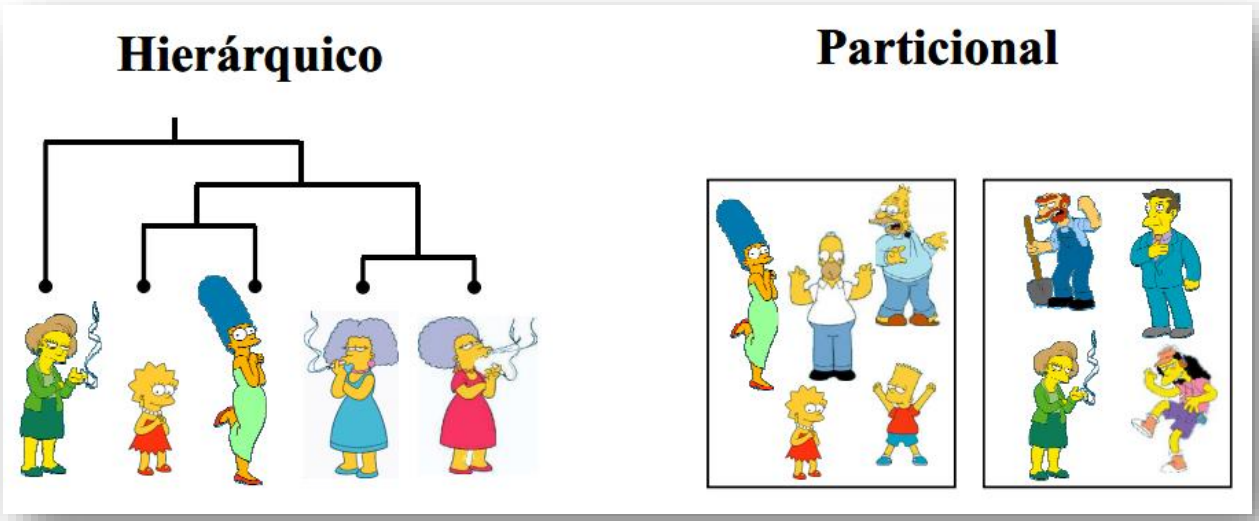
- **Clustering baseado em restrições**: os aplicativos do mundo real podem precisar realizar clustering sob vários tipos de restrições. Suponha que seu trabalho seja escolher os locais para um determinado número de novos caixas eletrônicos (ATMs) em uma cidade. Para decidir sobre isso, você pode agrupar famílias enquanto considera as restrições, como os rios e rodovias da cidade e os tipos e número de clientes por agrupamento. Uma tarefa desafiadora é encontrar grupos de dados com bom comportamento de clustering que satisfaçam as restrições especificadas.

- **Interpretabilidade e usabilidade**: os usuários desejam que os resultados do agrupamento sejam interpretáveis, compreensíveis e utilizáveis. Ou seja, o clustering pode precisar estar vinculado a interpretações e aplicativos semânticos específicos. É importante estudar como um objetivo do aplicativo pode influenciar a seleção de recursos e métodos de clustering.

Outro ponto impotante são os critérios utilizados para o particionamento. A seguir estão os aspectos ortogonais com os quais os métodos de agrupamento podem ser comparados:

Os critérios de particionamento: em alguns métodos, todos os objetos são particionados de forma que não haja hierarquia entre os clusters. Ou seja, todos os clusters estão conceitualmente no mesmo nível. Esse método é útil, por exemplo, para dividir clientes em grupos de modo que cada grupo tenha seu próprio gerente. Como alternativa, outros métodos particionam objetos de dados hierarquicamente, onde os clusters podem ser formados em diferentes níveis semânticos. Por exemplo, na mineração de texto, podemos querer organizar um corpus de documentos em vários

tópicos gerais, como "política" e "esportes", cada um dos quais pode ter subtópicos, por exemplo, "futebol", "basquete", "beisebol" e “hóquei” podem existir como subtópicos de “esportes”. Os últimos quatro subtópicos estão em um nível inferior na hierarquia do que "esportes".



Separação de clusters: Alguns métodos particionam objetos de dados em clusters **mutuamente exclusivos**. Ao agrupar clientes em grupos para que cada grupo seja atendido por um gerente, cada cliente pode pertencer a apenas um grupo. Em algumas outras situações, os clusters podem **não ser exclusivos**, ou seja, um objeto de dados pode pertencer a mais de um cluster. Por exemplo, ao agrupar documentos em tópicos, um documento pode estar relacionado a vários tópicos. Assim, os tópicos como clusters podem não ser exclusivos.

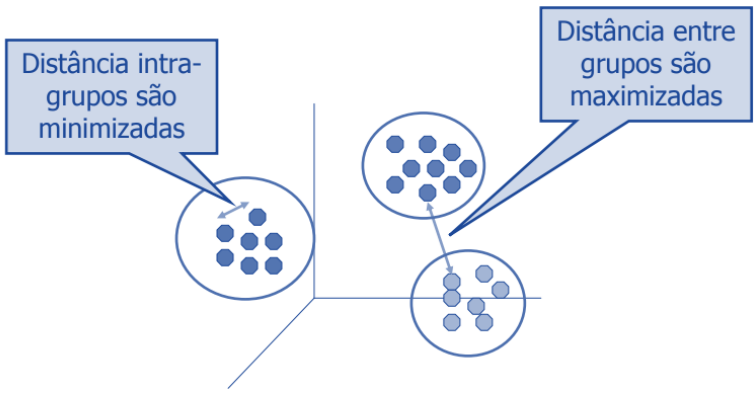
Medida de similaridade: Alguns métodos determinam a similaridade entre dois objetos **pela distância entre eles**. Essa distância pode ser definida no espaço euclidiano, uma rede rodoviária, em um espaço vetorial ou qualquer outro espaço. Em outros métodos, a similaridade pode ser definida pela **conectividade** com base na **densidade** ou **contiguidade** e não pode depender da distância absoluta entre dois objetos. As medidas de similaridade desempenham um papel fundamental no projeto de métodos de agrupamento. Embora os métodos baseados em distância possam muitas vezes tirar proveito de técnicas de otimização, os métodos baseados em densidade e continuidade podem frequentemente encontrar clusters de forma arbitrária.

Espaço de clustering: Muitos métodos de clustering procuram clusters em todo o espaço de dados fornecido. Esses métodos são úteis para conjuntos de dados de baixa dimensionalidade. Com dados de alta dimensão, no entanto, pode haver muitos atributos irrelevantes, que podem tornar as medições de similaridade não confiáveis. Consequentemente, os clusters encontrados em todo o espaço costumam não ter sentido. Geralmente, é melhor procurar clusters em diferentes subespaços do mesmo conjunto de dados. *O clustering de* subespaço descobre clusters e subespaços (geralmente de baixa dimensionalidade) que manifestam similaridade de objetos.

Visão geral dos métodos básicos de clustering

Existem muitos algoritmos de agrupamento na literatura. É difícil fornecer uma categorização precisa dos métodos de agrupamento porque essas categorias podem se sobrepor, de modo que um método pode ter recursos de várias categorias. No entanto, é útil apresentar um quadro relativamente organizado dos métodos de agrupamento. Em geral, os principais métodos de agrupamento fundamentais podem ser classificados nas seguintes categorias:

- **Métodos de particionamento:** Dado um conjunto de *n* objetos, um método de particionamento constrói *k* partições dos dados, onde cada partição representa um cluster *e* $k \leq n$. Ou seja, ele divide os dados em *k* grupos de forma que **cada grupo deve conter pelo menos um objeto**. Em outras palavras, os métodos de particionamento conduzem o particionamento sobre os conjuntos de dados. Os métodos básicos de particionamento geralmente adotam *separação de cluster exclusiva*. Ou seja, cada objeto deve pertencer a exatamente um grupo.

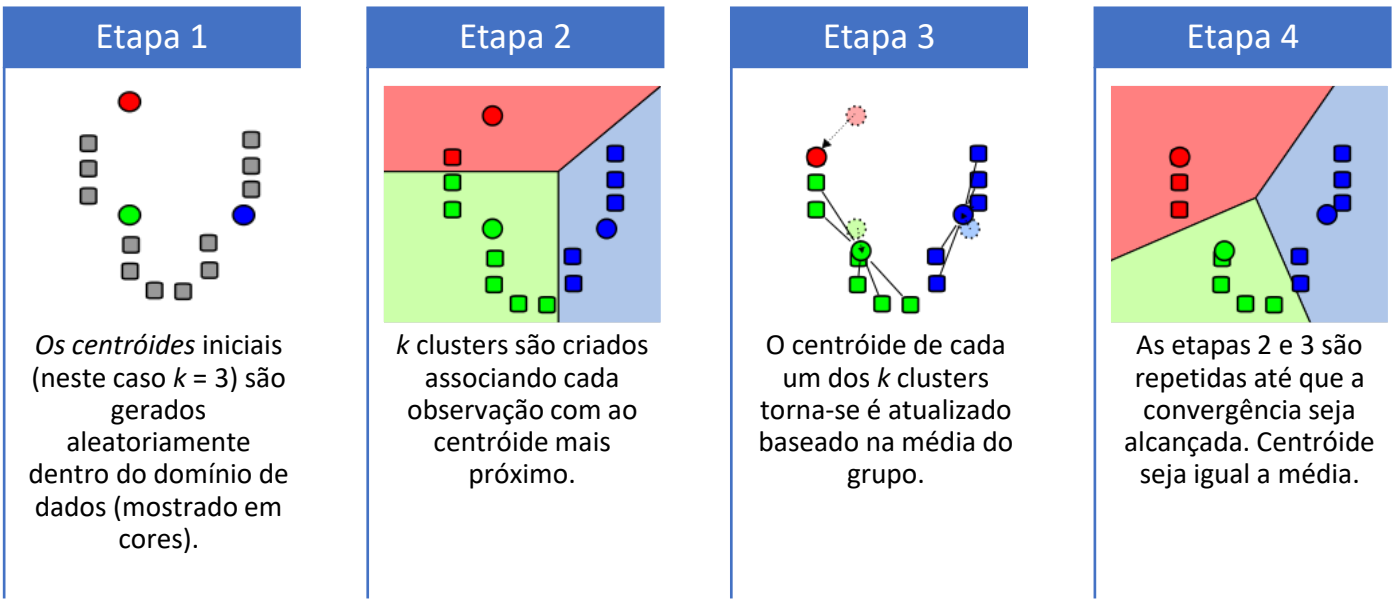


A maioria dos métodos de particionamento **são baseados em distância**. Dado *k*, o número de partições a serem construídas, um método de particionamento cria um particionamento inicial. Em seguida, ele usa uma **técnica de realocação iterativa** que tenta melhorar o particionamento movendo objetos de um grupo para outro. O critério geral de um bom particionamento é que os objetos no mesmo cluster são “próximos” ou relacionados entre si, enquanto os objetos em diferentes clusters são “distantes” ou muito diferentes. Existem vários tipos de outros critérios para julgar a qualidade das partições. Os métodos de particionamento tradicionais podem ser estendidos para clustering de subespaço, em vez de pesquisar todo o espaço de dados. Isso é útil quando há muitos atributos e os dados são esparsos.

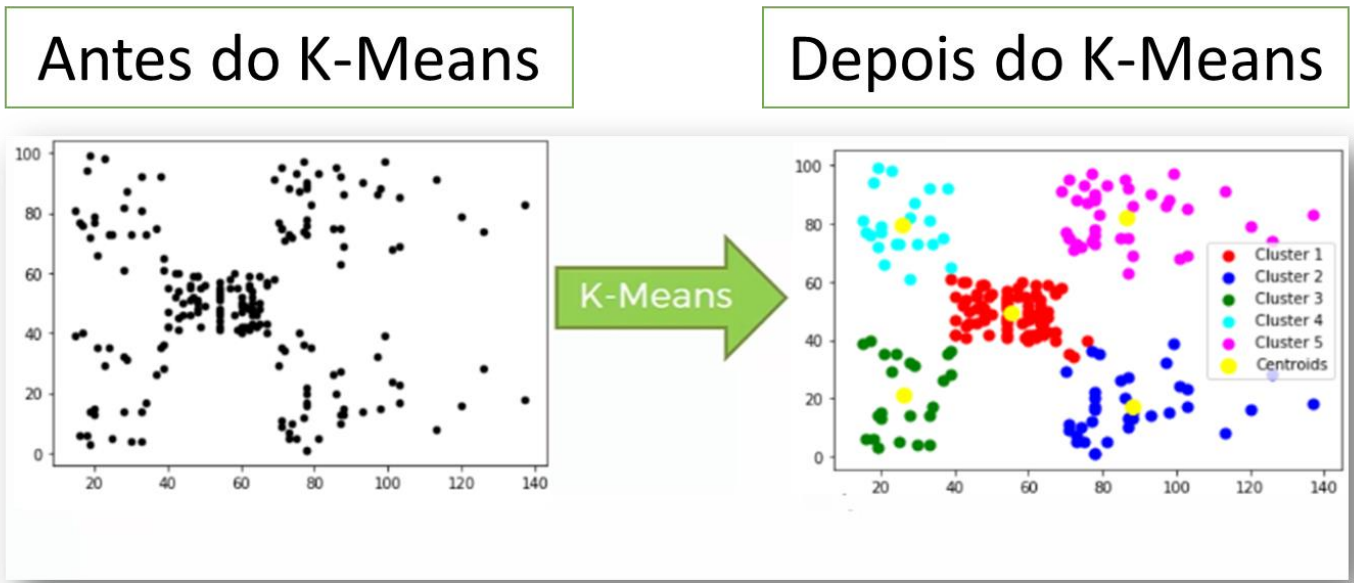
Atingir a otimização global em clustering baseado em particionamento costuma ser computacionalmente proibitivo, exigindo potencialmente uma enumeração exaustiva de todas as partições possíveis. Em vez disso, a maioria dos aplicativos adota métodos heurísticos populares, como abordagens gananciosas como os algoritmos ***k-means*** e ***k-medoids***, que melhoram progressivamente a qualidade do agrupamento e se aproximam de um ótimo local. Esses métodos heurísticos de agrupamento funcionam bem para localizar clusters em formato esférico em bancos de dados de pequeno a médio porte. Para localizar clusters com formas complexas e para conjuntos de dados muito grandes, os métodos baseados em particionamento precisam ser estendidos.

Vejamos um exemplo ... para agrupar os elementos, o algoritmo k-means usa uma medida de distância para encontrar a semelhança ou proximidade entre os pontos de dados. Antes de usar o algoritmo k-means, a medida de distância mais apropriada precisa ser selecionada. Por padrão, a medida de distância euclidiana será usada. Além disso, se o conjunto de dados tiver outliers, um mecanismo precisa ser desenvolvido para determinar os critérios que devem ser identificados e remover os outliers do conjunto de dados.

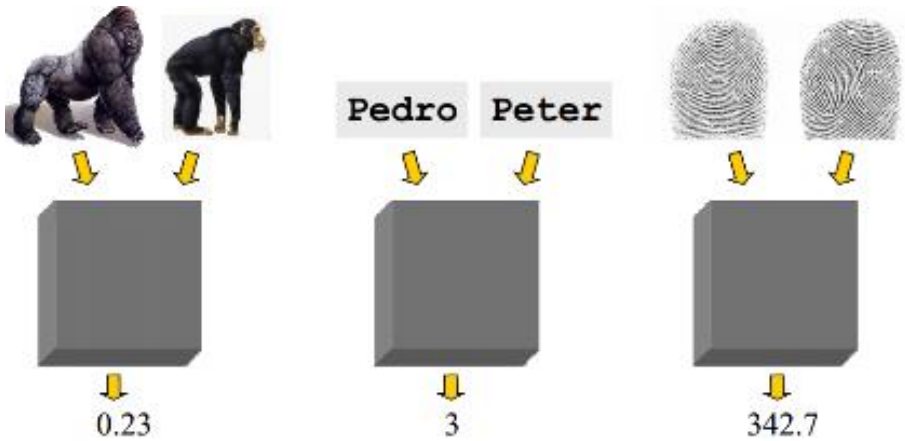
Depois desta etapa inicial, escolhemos a quantidade de grupos nos quais os registros serão divididos (k). A partir deste valor escolhemos K pontos no espaço de registros para serem nossos centroides. A sequência do algoritmo vai pegar cada registro dos dados, calcular a distância dele para cada centroide, agrupar o ponto ao conjunto do centroide mais próximo. Essa etapa é feita até que todos os registros sejam agrupados. Agora precisamos testar se os nossos centroides realmente são centro cada grupo, fazemos isso calculando a média dos pontos dos agrupamentos. Caso o centroide não seja o centro de cada grupo, os centros calculados passam a ser o novo valor de centroide e todos os registros são novamente agrupados. Esse ciclo é repetido até que o valor do centroide convirja para os centros de cada grupo, ou que outra condição de parada seja atingida. Ficou complicado! Eu sei ... deixar eu tentar apresentar algumas figuras:



Perceba que existe uma sequência de passos para que os grupos seja definidos de acordo com o critério de distância. A ideia do algoritmo é fazer com que o centroide convirja para a média, por isso nome do algoritmo k-means ou k-médias. Na figura abaixo, apresentamos apenas os estágios inicial e final da execução do algoritmo de K-means sobre um conjunto de dados:



As medidas de distância podem ser calculadas entre dois objetos. Sejam O1 e O2 dois objetos de um universo de possíveis objetos. A distância (dissimilaridade) entre O1 e O2 é um número real denotado por D (O1, O2). Observem a figura abaixo para entender melhor o conceito:



Algumas propriedades podem ser analisadas nas medidas de distância. A **simetria** (D (A, B) = D (B,A)), caso contrário você poderia afirmar que “Alex parece com Bob, mas Bob não parece com Alex”. A **constância de autos simetria** (D (A, A) = 0), caso contrário você poderia afirmar que “Alex parece mais com Bob, do que o próprio Bob”.

A **positividade** (D(A,B) = 0 ←→ A = B) caso contrário existiriam objetos no seu mundo que são diferentes, mas você não consegue diferenciá-los. E a **desigualdade triangular** (D (A,B) ≤ D(A,C) + D (B,C)) caso contrário você poderia afirmar que “Alex é parecido com Bob, e Alex é parecido com Carl, mas Bob não se parece com Carl”.

- **Métodos hierárquicos:** um método hierárquico cria uma decomposição hierárquica de um determinado conjunto de objetos de dados. Um método hierárquico pode ser classificado como ***aglomerativos*** ou ***divisivo***, com base em como a decomposição hierárquica é formada. A *abordagem aglomerativa*, também chamada *de* abordagem ***ascendente***, começa com cada objeto formando um grupo separado. Ele mescla sucessivamente os objetos ou grupos próximos uns dos outros, até que todos os grupos sejam mesclados em um (o nível mais alto da hierarquia) ou uma condição de encerramento seja alcançada.

  A *abordagem divisionista*, também chamada de ***top-down (descendente)***, começa com todos os objetos no mesmo cluster. Em cada iteração sucessiva, um cluster é dividido em clusters menores, até que, eventualmente, cada objeto esteja em um cluster ou uma condição de parada seja alcançada.

  Os métodos de agrupamento hierárquico podem ser baseados em distância ou baseados em densidade e continuidade. Várias extensões de métodos hierárquicos consideram o agrupamento em subespaços também.

Figura 4 - A figura acima representa um dendrograma que mostra os diversos níveis de um cluster hierárquico

  Os métodos hierárquicos sofrem com o fato de que uma vez que uma etapa (mesclar ou dividir) é concluída, ela nunca pode ser desfeita. Essa rigidez é útil porque leva a custos de computação menores por não ter que se preocupar com um número de escolhas diferentes. No entanto, essas técnicas não podem corrigir decisões erradas.

- **Métodos baseados em densidade:** a maioria dos métodos de particionamento agrupa objetos com base na distância entre os objetos. Esses métodos podem encontrar apenas agrupamentos de forma esférica e encontram dificuldade em descobrir aglomerados de formas arbitrárias. Outros métodos de agrupamento foram desenvolvidos com base na noção de ***densidade***. A ideia geral deles é continuar a crescer um determinado cluster, desde que a densidade (número

de objetos ou pontos de dados) na “vizinhança” exceda algum limite. Por exemplo, para cada ponto de dados dentro de um determinado cluster, a vizinhança deve conter pelo menos um número mínimo de pontos.

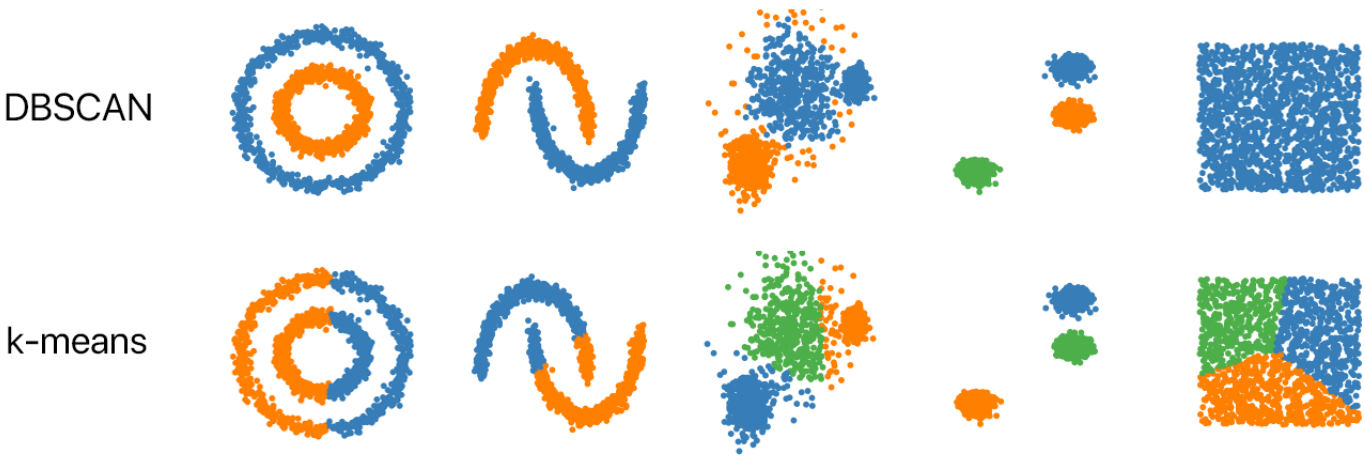


Figura 5 - Na figura acima temos vários conjuntos de dados separados pelo método de densidade (DBSCAN) e pelo método de distância (K-means). Perceba a diferença entre eles! :)

Esse método pode ser usado para filtrar ruído ou outliers e descobrir clusters de forma arbitrária. Os métodos baseados em densidade podem dividir um conjunto de objetos em vários clusters exclusivos ou em uma hierarquia de clusters. Normalmente, os métodos baseados em densidade consideram apenas clusters exclusivos e não consideram clusters difusos. Além disso, os métodos baseados em densidade podem ser estendidos do espaço total para o agrupamento de subespaço.

- **Métodos baseados em grid:** Os métodos baseados em *grid* dividem o espaço em um número finito de células que formam uma estrutura de *grid*. Todas as operações de agrupamento são realizadas na estrutura da grid (ou seja, no espaço quantizado). A principal vantagem dessa abordagem é seu tempo de processamento rápido, que normalmente é independente do número de objetos de dados e dependente apenas do número de células em cada dimensão no espaço quantizado.

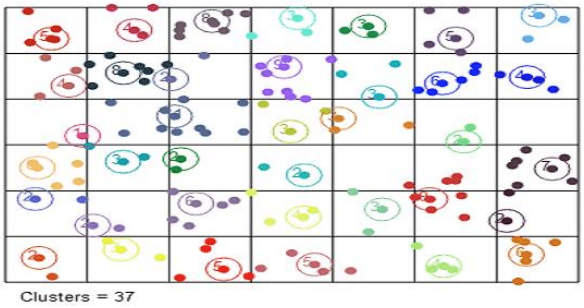


Figura 6 - Clustering baseado em grid

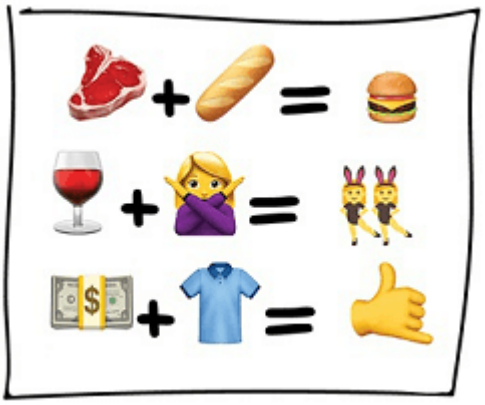
O uso de grid costuma ser uma abordagem eficiente para muitos problemas de mineração de dados espaciais, incluindo clustering. Portanto, os métodos baseados em *grid* podem ser integrados a outros métodos de agrupamento, como métodos baseados em densidade e métodos hierárquicos.

Esses métodos são resumidos brevemente na figura abaixo. Alguns algoritmos de agrupamento integram as ideias de vários métodos de agrupamento, de modo que às vezes é difícil classificar um determinado algoritmo como pertencendo exclusivamente a apenas uma categoria de método de agrupamento. Além disso, alguns aplicativos podem ter critérios de clustering que requerem a integração de várias técnicas de clustering.

Métodos baseados em partição	• Procura por cluster mutualmente exclusivos em formato esférico • Baseado em distância • Pode usar a média ou os meóides para representar o centro do cluster • Efetivo em conjuntos de dados pequenos e médios
Métodos hierárquico	• O cluster segue uma decomposição hierárquica em múltiplos níveis • Não pode corrigir erros gerados por uma junção ou separação • Pode incorporar outras técnicas como microclustering ou considerar relacionamentos entre objetos
Métodos baseados em densidade	• Podem encontrar formatos arbitrários para o cluster • Os cluster são regiões densas de objetos no espaço que são separados por regiões de baixa densidade.
Métodos baseados em grids	• Usam uma estrutura de dados em grid • Processam os dados mais rapidamente (independentemente do número de objetos de dados, e dependente do tamanho do grid)

Vejamos como esse assunto foi cobado em provas anteriores:

Regras de associação



A mineração frequente de conjuntos de itens leva à descoberta de associações e correlações entre itens em grandes conjuntos de dados transacionais ou relacionais. Com grandes quantidades de dados continuamente sendo coletados e armazenados, muitos setores estão se interessando em extrair esses padrões de seus bancos de dados. A descoberta de relacionamentos de correlação interessantes entre grandes quantidades de registros de transações de negócios pode ajudar em muitos processos de tomada de decisão de negócios, como design de catálogo, marketing cruzado e análise do comportamento de compra do cliente

As regras de associação relacionam a presença de um conjunto de itens com outra faixa de valores de outro conjunto de variáveis. Podemos pensar nos seguintes exemplos: 1. Quando uma mulher compra uma bolsa em uma loja, ela está propensa a comprar sapatos (na mesma loja) e 2. Uma imagem de raio X contendo as características a e b provavelmente exibirá também a característica c (o mesmo raio-x). Veja as figuras abaixo que ilustram esses exemplos:

Uma regra de associação é um padrão da forma X → Y, onde X e Y são conjuntos de valores. O seguinte padrão “clientes que compram pão também compram leite” representa uma regra de associação que reflete um padrão de comportamento dos clientes do supermercado. Descobrir regras de associação entre produtos comprados por clientes numa mesma compra pode ser útil para melhorar a organização das prateleiras, facilitar (ou dificultar) as compras do usuário ou induzi-lo a comprar mais.

Os autores definem os conceitos de lado da mão direita e lado da mão esquerda para ilustrar essa ideia de compra casada. É como se eu estivesse propenso a consumir os dois produtos. A união entre o lado da mão esquerda e o lado da mão direita gera outra definição conhecida como conjunto-item (o conjunto de todos os itens comprados pelos clientes). Observe a figura abaixo com o conjunto-item formado por picanha (Friboi é claro!) e carvão!

Suporte: % (LME U LMD)

Confiança: Suporte (LME U LMD)/ Suporte (LME)

Para que uma regra de associação seja do interesse de um pesquisador de dados, a regra precisa satisfazer algumas medidas. O suporte que define quão frequente a regra acontece no banco de dados e a confiança que é a força da regra. Vamos detalhar um pouco mais essas definições.

O **Suporte** é uma **medida objetiva** para avaliar o **interesse** de uma **regra de associação.** Representa **a porcentagem de transações (%)** de um banco de dados de transações **onde a regra se verifica.** A medida de suporte responde a seguinte questão: quão frequente a regra acontece no banco de dados?

A **Confiança** é outra medida **objetiva** para **regras de associação** que mede o **grau de certeza** de uma associação. Em termos estatísticos, trata-se simplesmente da **probabilidade condicional P (Y | X)**, isto é, a porcentagem de transações contendo os itens de X que também contêm os itens de Y.

Se pensarmos no universo como o conjunto de itens disponíveis em uma loja, então cada item possui uma variável booleana que representa a presença ou ausência daquele item. Cada cesta pode então ser representada por um vetor booleano de valores atribuídos a essas variáveis. Os vetores booleanos podem ser analisados para padrões de compra que refletem itens que são frequentemente associados ou comprados juntos. Esses padrões podem ser representados na forma de regras de associação. Por exemplo, as informações de que os clientes que compram computadores também tendem a comprar software antivírus ao mesmo tempo são representadas na seguinte regra de associação:

$$computador \rightarrow antivírus\ (suporte\ 2\%, confiança: 60\%)$$

Como já falamos, o **suporte** e a **confiança** das regras são duas medidas de das regras. Eles refletem respectivamente a **utilidade** e a **certeza** das regras descobertas. Um suporte de 2% para a regra acima significa que 2% de todas as transações em análise mostram que o computador e o software antivírus são comprados juntos. Uma confiança de 60% significa que 60% dos clientes que compraram um computador também compraram o software. Normalmente, as regras de associação são consideradas interessantes se satisfizerem **um limite mínimo de suporte** e **um limite mínimo de confiança**. Esses limites podem ser definidos por usuários ou especialistas no domínio. Uma análise adicional pode ser realizada para descobrir correlações estatísticas interessantes entre os itens associados.

APOSTA ESTRATÉGICA

A ideia desta seção é apresentar os pontos do conteúdo que mais possuem chances de serem cobrados em prova, considerando o histórico de questões da banca em provas de nível semelhante à nossa, bem como as inovações no conteúdo, na legislação e nos entendimentos doutrinários e jurisprudenciais[1].

Algoritmos de aprendizagem de máquina são programas que podem aprender com dados e melhorar a partir da experiência, sem intervenção humana. Tarefas de aprendizagem podem incluir aprender a função que mapeia a entrada para a saída, aprender a estrutura oculta em dados não rotulados; ou "aprendizagem baseada em instâncias", onde um rótulo de classe é produzido para uma nova instância, comparando a nova instância (linha) com instâncias dos dados de treinamento, que foram armazenados na memória. O 'aprendizado baseado em instâncias' não cria uma abstração a partir de instâncias específicas.

K-Means

O clustering tem muitas aplicações. Por exemplo, na detecção de fraudes de cartão de crédito, o clustering pode agrupar transações fraudulentas, separando-as de transações normais. Ou, se tivéssemos apenas alguns rótulos para as observações em nosso conjunto de dados, poderíamos usar o clustering para agrupar as observações primeiro (sem usar rótulos). Então, poderíamos transferir os rótulos das poucas observações rotuladas para o resto das observações dentro do mesmo grupo. Esta é uma forma de aprendizado de transferência, um campo de rápido crescimento no aprendizado de máquina.

Em áreas como compras online e varejo, marketing, mídias sociais, sistemas de recomendação para filmes, música, livros, namoros, etc., o cluster pode agrupar pessoas semelhantes com base em seu comportamento. Uma vez estabelecidos esses grupos,

[1] Vale deixar claro que nem sempre será possível realizar uma aposta estratégica para um determinado assunto, considerando que às vezes não é viável identificar os pontos mais prováveis de serem cobrados a partir de critérios objetivos ou minimamente razoáveis.

os usuários de negócios terão uma melhor visão de sua base de usuários e poderão criar estratégias de negócios direcionadas para cada um dos grupos distintos.

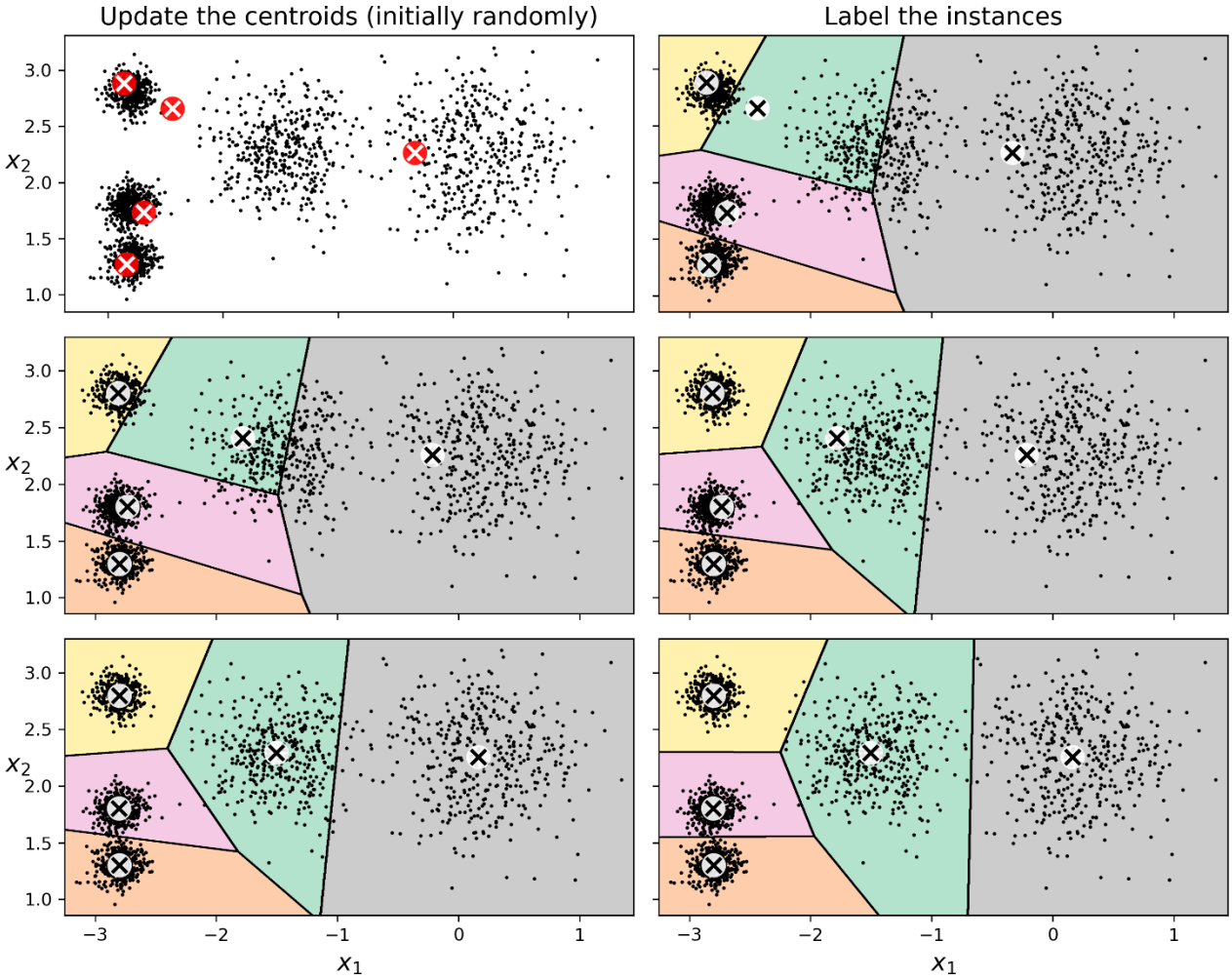
Algoritmo

O objetivo do agrupamento é identificar grupos distintos em um conjunto de dados de modo que as observações dentro de um grupo sejam semelhantes entre si, mas diferentes das observações em outros grupos. Em clustering k-means, especificamos o número de clusters desejados k, e o algoritmo atribuirá cada observação a exatamente um desses clusters k. O algoritmo otimiza os grupos minimizando a variação dentro do cluster (também conhecida como **inércia**) de modo que a soma das variações dentro do cluster em todos os clusters K seja o menor possível.

Considere o conjunto de dados não rotulado representado da figura acima: você pode ver claramente cinco bolhas de instâncias. O algoritmo K-Means é um algoritmo simples capaz de agrupar esse tipo de conjunto de dados de forma muito rápida e eficiente, muitas vezes em apenas algumas iterações. Foi proposto por Stuart Lloyd no Bell Labs em 1957 como uma técnica para modulação de código de pulso, mas só foi publicado fora da empresa em 1982. Em 1965, Edward W. Forgy publicou praticamente o mesmo algoritmo, então K-Means é às vezes referido como Lloyd-Forgy.

Então, como funciona o algoritmo? Bem, suponha que você recebeu os centroids. Você pode facilmente rotular todas as instâncias do conjunto de dados atribuindo cada uma delas ao cluster cujo centroide está mais próximo. Por outro lado, se você recebesse todos os rótulos de instância, você poderia facilmente localizar o centroide de cada cluster computando a média das instâncias nesse cluster. Mas você não recebe nem os rótulos nem os centroides, então como você pode proceder? Bem, basta **começar definindo os centroides aleatoriamente** (por exemplo, escolhendo k instâncias aleatoriamente do conjunto de dados e usando suas localizações como centróides). Em seguida, rotule as instâncias, atualize os centróides baseado no ponto médio de cada grupo, rotule as instâncias novamente, atualize os centróides e assim por diante até que os centroides parem de se mover. O algoritmo garante convergir em um número finito de passos (geralmente pequeno). Isso porque a distância média entre as instâncias e seu centroide mais próximo só pode descer a cada passo, e como não pode ser negativa.

Você pode ver o algoritmo em ação na figura abaixo: os centroides são inicializados aleatoriamente (canto superior esquerdo), então as instâncias são rotuladas (superior direita), então os centroides são atualizados (centro-esquerda), as instâncias são rotuladas novamente (centro-direita) e assim por diante. Como você pode ver, em apenas três iterações, o algoritmo atingiu um clustering que parece próximo do ideal.



DBSCAN

Este algoritmo define clusters como regiões contínuas de alta densidade. É assim que funciona:

Para cada instância, o algoritmo conta quantas instâncias estão localizadas a uma pequena distância ε (epsilon) dele. Essa região é chamada de ε-vizinhança.

Se uma instância tem pelo menos um número mínimo de instâncias em sua vizinhança ε (incluindo a si mesmo), então é considerada uma instância central. Em outras palavras, as instâncias centrais são aquelas que estão localizadas em regiões densas.

Todas as instâncias na vizinhança de uma instância central pertencem ao mesmo aglomerado. Esta vizinhança pode incluir outras instâncias centrais; portanto, uma longa sequência de instâncias centrais vizinhas forma um único cluster.

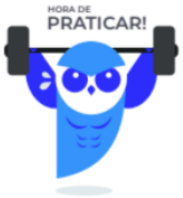
Qualquer instância que não seja uma instância central e não tenha uma vizinhança é considerada uma anomalia.

Imprima o capítulo Aposta Estratégica separadamente e dedique um tempo para absolver tudo o que está destacado nessas duas páginas. Caso tenha alguma dúvida, volte ao Roteiro de Revisão e Pontos do Assunto que Merecem Destaque. Se ainda assim restar alguma dúvida, não hesite em me perguntar no fórum.

QUESTÕES ESTRATÉGICAS

Nesta seção, apresentamos e comentamos uma amostra de questões objetivas selecionadas estrategicamente: são questões com nível de dificuldade semelhante ao que você deve esperar para a sua prova e que, em conjunto, abordam os principais pontos do assunto.

A ideia, aqui, não é que você fixe o conteúdo por meio de uma bateria extensa de questões, mas que você faça uma boa revisão global do assunto a partir de, relativamente, poucas questões.



1.

João trabalha no setor de BI da empresa e recebeu a tarefa de identificar agrupamentos de alunos de uma escola segundo seu desempenho acadêmico. A partir das notas obtidas, João deve formar grupos tal que integrantes de um grupo tenham desempenho similar, e que integrantes de grupos distintos sejam dissimilares. O algoritmo mais apropriado para essa tarefa é:

A Apriori;

B decision tree;

C PageRank;

D CART;

E k-means

Cometário: Veja a importância de saber os nomes dos algoritmos usados para cada tarefa de mineração, de forma direta:

a) Apriori → Regra de associação

b) Decision Tree → Classificação e Regressão

c) PageRank → é um algoritmo que prioriza as páginas exibidas na busca do Google.

d) CART → Classification and Regression Tree → Classificação e Regressão

e) K-means → Agrupamento.

Gabarito: E

2.

Maria está preparando um relatório sobre as empresas de serviços de um município, de modo a identificar e estudar o porte dessas empresas com vistas ao estabelecimento de políticas públicas e previsões de arrecadação. Maria pretende criar nove grupos de empresas, de acordo com os valores de faturamento, e recorreu às técnicas usualmente empregadas em procedimentos de data mining para estabelecer as faixas de valores de cada grupo. Assinale a opção que apresenta a técnica diretamente aplicável a esse tipo de classificação.

A Algoritmos de associação.

B Algoritmos de clusterização.

C Árvores de decisão.

D Modelagem de dados.

E Regressão linear.

Comentário: Vamos comentar cada uma das alternativas acima:

A) Algoritmos de associação dão suporte a uma tarefa descritiva, ou seja, de Aprendizado Não Supervisionado que tem por objetivo encontrar item que são consumidos em conjunto. Outros termos usados para descrever as regras de associação são: ARM (Association Rule Mining – Mineração de regras de associação), Link Analysis (Análise de links), Affinity Analysis (Análsie de afinidade), Market Basket Analysis – Análise de cesta de compras. Veja que a Maria não está comprando nada! 😊 Logo, essa não é a nossa resposta.

B) Algoritmos de clusterização estão associados a uma tarefa descritiva de agrupar os itens de dados de acordo com a similaridade. Logo, essa é a nossa resposta.

C) Árvore de decisão dão a base para algoritmos usados para classificação.

D) Modelagem de dados serve para descrever o conjunto de dados armazenados em algum sistema.

E) Regressão linear procura descrever uma função que melhor se adeque ao conjunto de dados. Geralmente usando a abordagem de Mínimos Quadrados Ordinários (MQO).

Gabarito: B

3.

São considerados algoritmos para redução de dimensionalidade:

a) PCA, Autoencoder e aprendizagem múltipla

b) KNN, CART, K-means

c) C4.5, Apriori, FP-Growth

d) Árvore de decisão, SVM e PageRank

e) SVM, KNN, Naive Bayes

Comentário: A alternativa A apresentar algoritmos usados para a redução de dimensionalidade. As demais letras apresentam algoritmos de:

Classificação: KNN, SVM, Naive Bayes

Árvore de decisão: CART, C4.5

Regra de associação: Apriori, FP-Growth

Clusterização: K-means

Gabarito: A

4.

A Inteligência Artificial (IA) apoia o desenvolvimento de soluções tecnológicas capazes de realizar atividades similares às capacidades cognitivas humanas. Como exemplo, a plataforma Sinapses, desenvolvida pelo Tribunal de Justiça do Estado de Rondônia (TJRO) e adaptada para uso nacional, gerencia o treinamento supervisionado de modelos de IA.

Em soluções de IA, a tecnologia que possui a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência usando dados de treinamento, podendo ser supervisionado ou não, é o(a):

A Motor de Inferência (Inference Engine) de Sistemas Especialistas (Expert Systems);

B Raciocínio Automatizado (Automated Reasoning);

C Compreensão de Linguagem Natural (Natural-Language Understanding);

D Representação do Conhecimento (Knowledge Representation) usando Lógica de Primeira Ordem (First Logic Order);

E Aprendizado de Máquina (Machine Learning).

Comentário:

Gabarito: E

Comentários

Veja que tratamos de desse conceito de melhoria de eficiência por meio da aprendizagem de máquina no início desta seção, logo, temos a resposta na alternativa E. As outras assertivas descrevem outras áreas/conceitos da Inteligência Artificial:

a) Um motor de inferência é uma ferramenta informatizada "caixa preta", também utilizada em Sistema Especialista (Inteligência Artificial), que após ser estimulada com solicitações predeterminadas, oferece as soluções possíveis. Este é o núcleo da inteligência artificial de um sistema especialista, onde a capacidade do motor de inferência é baseada numa combinação de procedimentos de raciocínios de forma regressiva (partindo de uma conclusão, feita pelo usuário ou pelo sistema, é feita uma pesquisa por meio do conhecimento acumulado para se provar a afirmação inicial) e progressiva (respostas fornecidas pelo usuário desencadeando um processo de

busca até que se encontre a solução ótima). Ela é provocada por requisições e tem que ter base de dados de conhecimento. Maior parte baseado em lógica de primeira ordem.

b) Raciocínio Automatizado (Automated Reasoning): cria formas de simular raciocínio lógico

c) Compreensão de Linguagem Natural (Natural-Language Understanding); meios para Compreensão da linguagem humana a partir de texto, áudio ou vídeo.

d) Representação do Conhecimento (Knowledge Representation) usando Lógica de Primeira Ordem (First Logic Order): descreve objetos e predicados relacionando objetos. Admite quantificadores (∀ e ∃). Representação por lógica matemática

Veja que aprendizado de máquina vai além desses conceitos porque propõe maior independência na execução dos sistemas, com base em aprendizado contínuo como se fosse um neurônio.

Gabarito: E.

5.

Sobre as técnicas de redução de dimensionalidade, assinale a alternativa correta:

a) As técnicas de compressão aplicam uma codificação ou transformação para que uma representação compacta dos dados ou atributos originais seja obtida.

b) A seleção de atributos, um dos métodos mais úteis e eficazes na compressão de dados, é um procedimento estatístico que converte um conjunto de objetos com atributos possivelmente correlacionados em um conjunto de objetos com atributos linearmente descorrelacionados, chamados de componentes principais.

c) O número de componentes principais é maior ou igual ao número de atributos da base, e a transformação é definida de forma que o primeiro componente principal possua a menor variância.

d) Compressão de atributos trata os valores de atributos que são substituídos por intervalos ou níveis conceituais mais elevados, reduzindo a quantidade final de atributos.

e) Discretização efetua uma redução de dimensionalidade na qual atributos irrelevantes, pouco relevantes ou redundantes são detectados e removidos.

Comentários: Vamos comentar cada uma das alternativas:

a) (CERTO) As técnicas de compressão aplicam uma codificação ou transformação para que uma representação compacta dos dados ou atributos originais seja obtida.

b) (ERRADO) A análise de componentes principais (Principal Component Analysis – PCA), um dos métodos mais úteis e eficazes na compressão de dados, é um procedimento estatístico que converte um conjunto de objetos com atributos possivelmente correlacionados em um conjunto de objetos com atributos linearmente descorrelacionados, chamados de componentes principais.

c) (ERRADO) O número de componentes principais é menor ou igual ao número de atributos da base, e a transformação é definida de forma que o primeiro componente principal possua a maior variância.

d) (ERRADO) Discretização trata os valores de atributos que são substituídos por intervalos ou níveis conceituais mais elevados, reduzindo a quantidade final de atributos.

e) (ERRADO) Seleção de atributos (ou características) efetua uma redução de dimensionalidade na qual atributos irrelevantes, pouco relevantes ou redundantes são detectados e removidos.

Assim, temos a nossa resposta na alternativa A.

Gabarito: A.

6.

Considerando que a tabela abaixo apresenta a lista de itens compradas por diferentes pessoas para seus respectivos churrascos de posse:

	Picanha	Carvão	Maminha	Pão de alho	Cerveja
1	X		X	X	
2		X	X		X
3	X	X		X	X
4		X		X	X
5	X	X	X		
6	X			X	X
7		X	X	X	
8	X		X	X	X
9	X	X	X	X	
10		X	X	X	X

Qual o percentual abaixo que melhor representa o nível de confiança da regra picanha → maminha?

a) 20%

b) 40%

c) 50%

d) 66%

e) 33%

Comentários: Para calcular a confiança temo que verificar o percentual de vezes em que a regra se verifica (40%) e dividir pela quantidade de vezes que picanha aparece (60%), assim temos que 40/60 é aproximadamente 66%, o que nos leva a resposta na alternativa D.

Gabarito: D.

7.

É considerado um algoritmo de regra de associação:

a) KNN

b) CART

c) K-means

d) FP-Growth

e) SVM

Comentários: Sabemos que:

KNN é um algoritmo de classificação baseado em votação, onde os K vizinhos mais próximos votam para escolha da classe.

CART também é um algoritmo de classificação e regressão baseado em árvores de decisão.

K-means é uma algoritmo de clusterização.

FP-Growth é um algoritmo de Regra de associação.

SVM também é uma técnica de classificação.

Logo, nossa resposta está na letra D.

Gabarito: D.

8.

Uma das questões importantes na formação do agrupamento é a medida usada para quantificar o grau de similaridade entre objetos. Na lista abaixo, qual dos nomes não é considerado um método de particionamento:

a) Baseados em partição

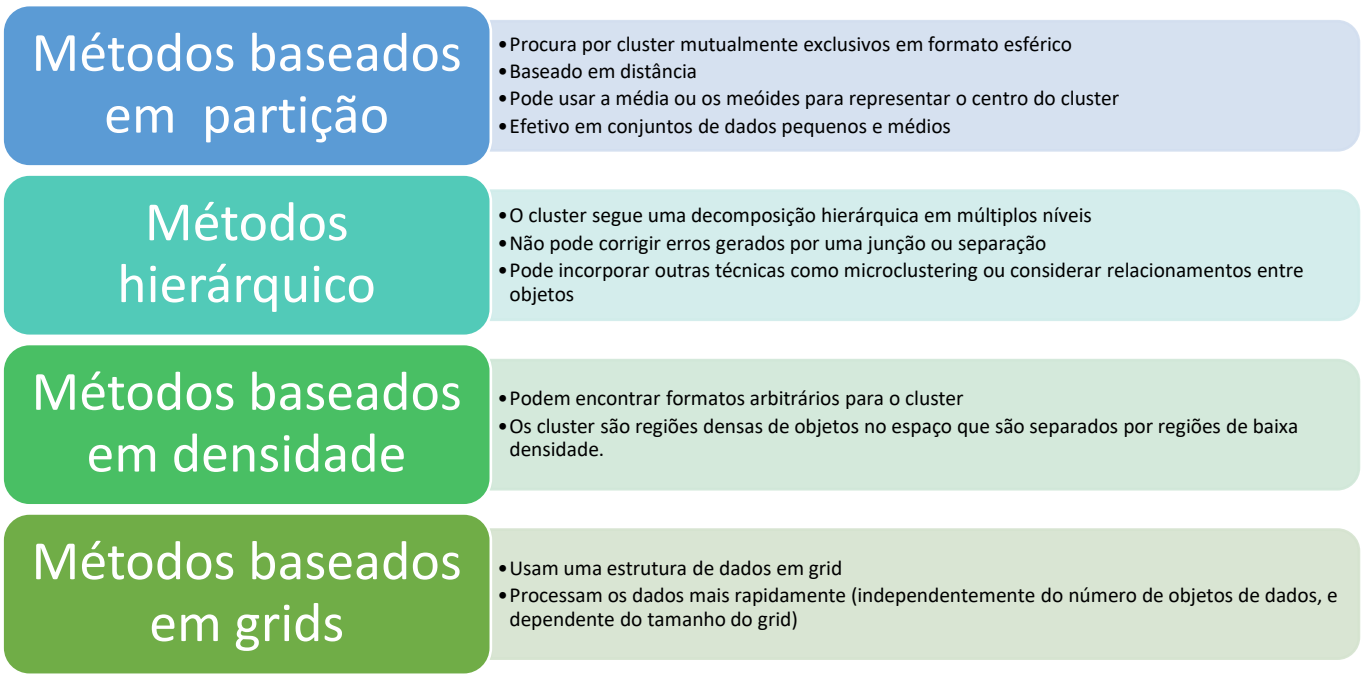
b) Hierárquico

c) Baseados em densidade

d) Métodos baseados em grids

e) Baseado em diferenciação

Comentários: A figura abaixo mostra os principais métodos de partições:



Veja que a única opção que não aparece na lista acima é o método baseado em diferenciação. Logo, nossa resposta encontra-se na alternativa E.

Gabarito: E.

9.

Sobre o algoritmo de k-means assinale a alternativa correta.

a) O algoritmo k -means define o centróide de um cluster como o valor médio dos pontos dentro do cluster.

b) O método k -means tem garantia de convergência para o ótimo global.

c) O primeiro passo do algoritimo de K-means é atribuir a cada ponto de dados no espaço do problema ao grupo do centróide mais próximo.

d) A medida de distância é sempre a mesma para todas as execuções do algoritmo, no caso usamos a distância euclidiana.

e) Os outlier não geram impacto negativo sobre o algoritmo visto que o objetivo é calcular a média de cada grupo.

Comentários: Vamos comentar cada uma das alternativas:

a) CERTO. Embora seja inicializado aleatoriamente, os valores dos centróides sempre vão ser ajustados nas iterações com o valor médio dos pontos no estado atual do algoritmo.

b) O método k -means não tem garantia de convergência para o ótimo global.

c) Atribuir a cada ponto de dados no espaço do problema ao centroide do cluster mais próximo é o quarto passo do k-means que começa com:

Passo 1 Escolher o número de clusters, k.

Passo 2 Entre os pontos de dados, escolher aleatoriamente k pontos como centróides de cluster.

Passo 3 Com base na medida de distância selecionada, calculamos iterativamente a distância de cada ponto no espaço do problema para cada um dos k centróides do cluster. Com base no tamanho do conjunto de dados, este pode ser um passo demorado, por exemplo, se existem 10.000 pontos no cluster e k = 3, isto significa que 30.000 distâncias têm de ser calculados.

d) Existem várias métricas de distâncias que podem ser escolhidas, além da euclidiana, apenas para citar exemplos temos a distância de Manhattan e a distância de Chebychev.

e) Outlier gera problema sim no K-means pois ele acaba distorcendo a média do grupo para o seu valor. Uma solução é usar o método k-medoids que usa o elemento mais significativo do grupo para representar o grupo (ao invés da média).

Logo, temos a nossa resposta na alternativa A.

Gabarito: A.

QUESTIONÁRIO DE REVISÃO E APERFEIÇOAMENTO

A ideia do questionário é elevar o nível da sua compreensão no assunto e, ao mesmo tempo, proporcionar uma outra forma de revisão de pontos importantes do conteúdo, a partir de perguntas que exigem respostas subjetivas.

São questões um pouco mais desafiadoras, porque a redação de seu enunciado não ajuda na sua resolução, como ocorre nas clássicas questões objetivas.

O objetivo é que você realize uma autoexplicação mental de alguns pontos do conteúdo, para consolidar melhor o que aprendeu ;)

Além disso, as questões objetivas, em regra, abordam pontos isolados de um dado assunto. Assim, ao resolver várias questões objetivas, o candidato acaba memorizando pontos isolados do conteúdo, mas muitas vezes acaba não entendendo como esses pontos se conectam.

Assim, no questionário, buscaremos trazer também situações que ajudem você a conectar melhor os diversos pontos do conteúdo, na medida do possível.

É importante frisar que não estamos adentrando em um nível de profundidade maior que o exigido na sua prova, mas apenas permitindo que você compreenda melhor o assunto de modo a facilitar a resolução de questões objetivas típicas de concursos, ok?

Nosso compromisso é proporcionar a você uma revisão de alto nível!

Vamos ao nosso questionário:

Perguntas

1) **Quais são as aplicações de Aprendizagem Não Supervisionada?**
2) **Quais são alguns problemas comuns de Machine Learning que o Aprendizado Não Supervisionado pode ajudar?**
3) **Como *a Análise de Componentes Principais (PCA)* é usada para redução de dimensionalidade?**
4) **As Redes Neurais podem ser usadas para aprendizado não supervisionado?**
5) **Qual é a diferença entre KNN e K-means Clustering?**
6) **O que é a *Maldição da Dimensionalidade* e como o Aprendizado Não Supervisionado pode ajudar a resolver?**
7) **Quais são os parâmetros de entrada envolvidos no DBSCAN?**

Perguntas com respostas

1) Quais são as aplicações de Aprendizagem Não Supervisionada?

Algumas aplicações comuns do mundo real de aprendizagem não supervisionada são:

Seleções de notícias: O Google News usa aprendizado não supervisionado para categorizar artigos sobre a mesma história de vários meios de comunicação online.

Visão computacional: Algoritmos de aprendizagem não supervisionados são usados para tarefas de percepção visual, como reconhecimento de objetos.

Imagem médica: O aprendizado de máquina não supervisionado fornece características essenciais aos dispositivos de imagem médica, como detecção, classificação e segmentação de imagens, usados em radiologia e patologia para diagnosticar pacientes de forma rápida e precisa.

Detecção de anomalias: Modelos de aprendizagem não supervisionados podem vasculhar grandes quantidades de dados e descobrir pontos de dados atípicos dentro de um conjunto de dados. Essas anomalias podem aumentar a conscientização em torno de equipamentos defeituosos, erros humanos ou falhas na segurança.

2) Quais são alguns problemas comuns de Machine Learning que o Aprendizado Não Supervisionado pode ajudar?

Alguns desafios comuns que o aprendizado não supervisionado pode ajudar são:

Dados rotulados insuficientes: Para o aprendizado supervisionado, há a exigência de que muitos dados rotulados para que o modelo se apresente bem. O aprendizado não supervisionado pode rotular automaticamente exemplos não rotulados. Isso funciona agrupando todos os pontos de dados e, em seguida, aplicando os rótulos dos rotulados aos não rotulados.

Overfitting: Algoritmos de aprendizagem de máquina às vezes podem sobreajustar aos dados de treinamento extraindo muito do ruído nos dados. Quando isso acontece, o algoritmo está memorizando os dados de treinamento em vez de aprender a generalizar o conhecimento dos dados de treinamento. O aprendizado não supervisionado pode ser introduzido como um regularizador. A regularização é um processo que ajuda a reduzir a complexidade de um algoritmo de aprendizagem de máquina, ajudando-o a capturar o sinal nos dados sem ajustar muito ao ruído.

Outliers: A qualidade dos dados é muito importante. Se os algoritmos de aprendizagem de máquina treinarem em outliers (casos raros), então seu erro de generalização será menor do que se eles forem ignorados. O aprendizado não supervisionado pode realizar detecção outlier usando redução de dimensionalidade e criar soluções especificamente para os outliers e, separadamente, uma solução para os dados normais.

Engenharia de recursos: A engenharia de recursos é uma tarefa vital para os cientistas de dados realizarem, mas a engenharia de recursos é muito trabalhosa, e requer um humano para projetar criativamente os recursos. O aprendizado de representação com aprendizados não supervisionados pode ser usado para aprender automaticamente o tipo certo de recursos para ajudar a tarefa em questão.

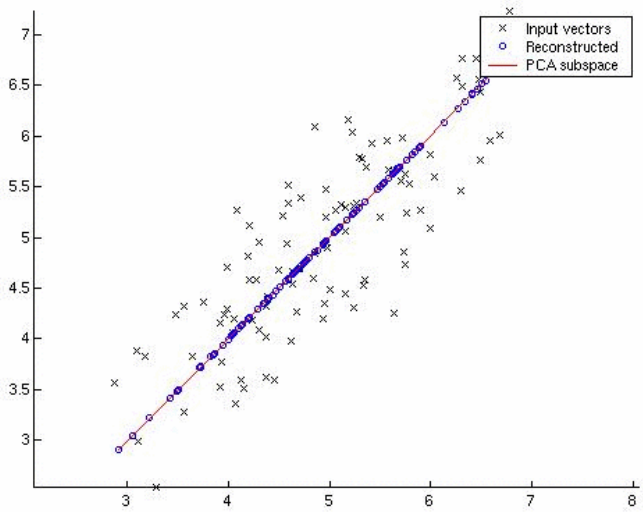
3) Como *a Análise de Componentes Principais (PCA)* é usada para redução de dimensionalidade?

A Análise de Componentes Principais (PCA) é uma técnica estatística não supervisionada e não paramétrica usada principalmente para **redução de dimensionalidade** na aprendizagem de máquina.

A análise dos componentes principais é uma técnica útil ao lidar com grandes conjuntos de dados. Em alguns campos, (bioinformática, marketing na internet, etc) acabamos coletando dados que tem milhares ou dezenas de milhares de dimensões. Manipular os dados neste formato não é desejável, devido a considerações práticas como memória e tempo de CPU. No entanto, não podemos ignorar arbitrariamente dimensões pois podemos perder algumas das informações que estamos tentando capturar!

A análise dos componentes principais é um método comum usado para gerenciar esse tradeoff. A ideia é que podemos de alguma forma selecionar as dimensões "mais importantes", e mantê-las, enquanto jogamos fora as que contribuem principalmente com o ruído.

Por exemplo, esta imagem abaixo mostra um conjunto de dados 2D sendo mapeado para uma dimensão:



Note que a dimensão escolhida não foi uma das duas originais: em geral, não será, porque isso significa que suas variáveis não foram ajustadas antes de começar o treinamento. Também podemos ver que a direção do componente principal é aquela que maximiza a variância dos dados projetados. Isto é o que queremos dizer com "manter o máximo de informações possível."

4) As Redes Neurais podem ser usadas para aprendizado não supervisionado?

Redes Neurais são usadas em aprendizado não supervisionado para aprender melhores representações dos dados de entrada.

As redes neurais podem aprender um mapeamento de documento para vetor de valor real de tal forma que vetores resultantes são semelhantes para documentos com conteúdo semelhante. Isso pode ser alcançado usando autoencoder, que é um modelo treinado para reconstruir o vetor original a partir de uma representação menor com erro de reconstrução como função de custo.

Existem redes neurais que são especificamente projetadas para agrupamento também. O mais conhecido são os mapas auto-organizados (SOM).

5) Qual é a diferença entre KNN e K-means Clustering?

K-mais próximo vizinhos ou KNN é um algoritmo de classificação supervisionado. Isso significa que precisamos de dados rotulados para classificar um ponto de dados sem rótulo. Ele tenta classificar um ponto de dados com base em sua proximidade com outros pontos de dados no espaço de recurso.

K-means clustering é um algoritmo de classificação não supervisionado. Ele requer apenas um conjunto de pontos não rotulados, por isso coleta e agrupa dados em número de clusters.

6) O que é a *Maldição da Dimensionalidade* e como o Aprendizado Não Supervisionado pode ajudar a resolver?

À medida que a quantidade de dados necessários para treinar um modelo aumenta, torna-se cada vez mais difícil para os algoritmos de aprendizagem de máquina lidarem com problema. À medida que mais recursos são adicionados ao processo de aprendizado de máquina, mais difícil o treinamento se torna.

Em espaço muito *dimensional*, algoritmos supervisionados aprendem a separar pontos e construir aproximações de função para fazer boas previsões. Quando o número de *recursos* aumenta, essa pesquisa se torna cara, tanto do ponto de vista do tempo quanto da computação. Podendo se tornar impossível encontrar uma boa solução rápido o suficiente. Esta é a ***maldição da dimensionalidade***.

Usando *a* ***redução dimensional de*** **aprendizagem não supervisionada**, as *características mais importantes* podem ser descobertas no conjunto de recursos originais. Em seguida, a dimensão deste conjunto de recursos pode ser reduzida a um número mais gerenciável, perdendo muito pouca informação no processo. Isso ajudará o aprendizado supervisionado a encontrar a função ideal para aproximar o conjunto de dados.

7) Quais são os parâmetros de entrada envolvidos no DBSCAN?

Há dois parâmetros empregados no DBSCAN:

Eps: Conhecido como epsilon e dita quais pontos são considerados vizinhos, pois é a distância máxima entre dois pontos que podem ser considerados como tal. Para mantê-lo simples, eps pode ser visto como o raio em torno de cada ponto.

min_pts: Este é conhecido como pontos mínimos ou amostras mínimas e é basicamente o número de observações que têm que existir em torno de um ponto (dentro de um raio) para que esse ponto seja considerado um ponto de dados central.

Forte abraço e bons estudos.

"Hoje, o 'Eu não sei', se tornou o 'Eu ainda não sei'"

(Bill Gates)

Thiago Cavalcanti

Face: www.facebook.com/profthiagocavalcanti
Insta: www.instagram.com/prof.thiago.cavalcanti
YouTube: youtube.com/profthiagocavalcanti

ESSA LEI TODO MUNDO CONHECE:
PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.

1. 

Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.

2. 

Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).

3. 

Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).

4. 

Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).

5. 

Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).

6. 

Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).

7. 

Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.

8. 

O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.

Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.