Machine Learning Tom M. Mitchell

O campo do ML preocupa-se com a questão de como construir programas de computador que melhoram automaticamente com a experiência. Nos últimos anos têm sido desenvolvidas muitas aplicações de aprendizagem de máquinas de sucesso, desde programas de mineração de dados que aprendem a detectar transacções fraudulentas com cartões de crédito, a sistemas de filtragem de informação que aprendem as preferências de leitura dos utilizadores, a veículos autónomos que aprendem a conduzir em auto-estradas públicas. Ao mesmo tempo, houve avanços importantes na teoria e algoritmos que formam os alicerces deste campo.

DECISION TREE LEARNING

A aprendizagem da árvore de decisão é um dos métodos mais amplamente utilizados e práticos para a inferência indutiva. É um método de aproximação de funções de valor discreto que é robusto para dados ruidosos e capaz de aprender expressões distintivas. Este capítulo descreve uma família de algoritmos de aprendizagem em árvore de decisão que inclui algoritmos amplamente utilizados, tais como ID3, ASSISTANT, e C4.5. Estes métodos de aprendizagem em árvore de decisão procuram um espaço de hipóteses completamente expressivo e assim evitam as dificuldades de espaços de hipóteses restritos. O seu viés indutivo é uma preferência por árvores pequenas em vez de árvores grandes.

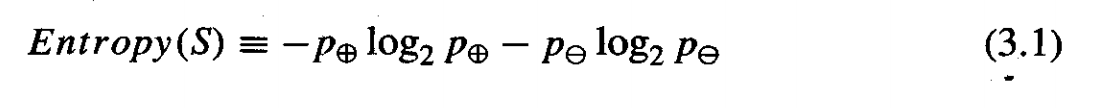
3.4 O ALGORITMO BÁSICO DE APRENDIZAGEM DA ÁRVORE DE DECISÃO

A maioria dos algoritmos que foram desenvolvidos para a aprendizagem de árvores de decisão são variações de um algoritmo central que emprega uma pesquisa de cima para baixo e ambiciosa através do espectro de possíveis árvores de decisão. Esta abordagem é exemplificada pelo algoritmo ID3 (Quinlan 1986) e o seu sucessor C4.5 (Quinlan 1993), que constituem o foco principal da nossa discussão aqui. Nesta secção apresentamos o algoritmo básico para a aprendizagem de árvores de decisão, correspondendo aproximadamente ao algoritmo ID3. Na Secção 3.7 consideramos uma série de extensões a este algoritmo básico, incluindo extensões incorporadas em C4.5 e outros algoritmos mais recentes para aprendizagem de árvore de decisão.

O nosso algoritmo básico, ID3, aprende árvores de decisão construindo-as de cima para baixo, começando com a pergunta "que atributo deve ser testado na raiz da árvore?'Para responder a esta pergunta, cada atributo de instância é avaliado utilizando um teste estatístico para determinar quão bem classifica por si só os exemplos de formação. O melhor atributo é seleccionado e utilizado como o teste na raiz do nodo da árvore. Um descendente do nodo raiz é então criado para cada valor possível deste atributo, e os exemplos de treino são ordenados para o nodo descendente apropriado (ou seja, para baixo do ramo correspondente ao valor do exemplo para este atributo). Todo o processo é então repetido utilizando os exemplos de treino associados a cada nodo descendente para seleccionar o melhor atributo a testar nesse ponto da árvore. Isto forma uma procura ambiciosa de uma árvore de decisão aceitável, na qual o algoritmo nunca retrocede para reconsiderar escolhas anteriores. Uma versão simplificada do algoritmo, especializada na aprendizagem de funções com valor booleano (ou seja, aprendizagem de conceitos), é descrita no Quadro 3.1.

É definida uma propriedade estatística, chamada ganho de informação, que mede o quão bem um dado atributo separa os exemplos de treino de acordo com a sua classificação alvo. O ID3 utiliza esta medida de ganho de informação para seleccionar entre os atributos candidatos em cada passo enquanto cresce a árvore.

A fim de definir com precisão o ganho de informação, começamos por definir uma medida comummente utilizada na teoria da informação, chamada entropia, que caracteriza a (im)pureza de uma colecção arbitrária de exemplos. Dada uma colecção S, contendo exemplos positivos e negativos de algum conceito alvo, a entropia de S relativa a esta classificação booleana é



onde p+, é a proporção de exemplos positivos em S e p-, é a proporção de exemplos negativos em S. Em todos os cálculos envolvendo entropia definimos 0 log 0 a ser 0.

Até agora, discutimos a entropia no caso especial em que a classificação alvo é booleana. Mais genericamente, se o atributo alvo pode assumir valores c diferentes, então a entropia de S relativa a esta classificação c-wise é definida como

A picture containing text

Description automatically generated

onde pi é a proporção de S pertencente à classe i. Note-se que o logaritmo ainda é base 2 porque a entropia é uma medida do comprimento de codificação esperado medido em pedaços. Note-se também que se o atributo alvo pode assumir c valores possíveis, a entropia pode ter valores tão grandes quanto o log2c.

3.4.1.2 MEDIDAS DE OBTENÇÃO DE INFORMAÇÃO A REDUÇÃO ESPERADA

EM ENTRÓPIA

Dada a entropia como medida da impureza numa colecção de exemplos de formação, podemos agora definir uma medida da eficácia de um atributo na classificação dos dados de formação. A medida que utilizaremos, denominada ganho de informação, é simplesmente a redução esperada da entropia causada pela partição dos exemplos de acordo com este atributo. Mais precisamente, o ganho de informação, Ganho(S, A) de um atributo A, relativo a uma colecção de exemplos S, é definido como

onde Valores(A) é o conjunto de todos os valores possíveis para o atributo A, e S, é o subconjunto de S para o qual o atributo A tem valor v (ou seja, S, = {s E SIA(s) = v))). Nota o primeiro termo em Equação (3.4) é apenas a entropia da colecção original S, e o segundo termo é o valor esperado da entropia após a partição de S utilizando o atributo A. A entropia esperada descrita por este segundo termo é simplesmente a soma das entropia de cada subconjunto S, ponderada pela fracção de exemplos que pertencem a S,. Ganho(S, A) é, portanto, a redução esperada da entropia causado por conhecer o valor do atributo A. Por outras palavras, Gain(S, A) é o informação fornecida sobre o valor alvo e o valor da acção, dado o valor de alguns outro atributo A. O valor de Gain(S, A) é o número de bits guardados quando codificando o valor-alvo de um membro arbitrário de S, ao conhecer o valor de atributo A.

O ganho de informação é precisamente a medida utilizada pelo ID3 para seleccionar o melhor atributo em cada etapa do crescimento da árvore.

O algoritmo descrito na Tabela 3.1 cresce cada ramo da árvore apenas o suficiente para classificar perfeitamente os exemplos de treino. Embora esta seja por vezes uma estratégia razoável, na realidade pode levar a dificuldades quando há ruído nos dados, ou quando o número de exemplos de treino é demasiado pequeno para produzir uma amostra representativa da verdadeira função alvo. Em qualquer um destes casos, este algoritmo simples pode produzir árvores que se sobreajustam aos exemplos de formação.

Uma forma de evitar o sobreajustamento é a chamada poda de redução-erro (Quinlan 1987), é considerada cada um dos nós de decisão na árvore como sendo candidatos à poda. A poda de um nó de decisão consiste em remover a sub-árvore enraizada nesse nó, tornando-o num nó de folha, e atribuindo-lhe a classificação mais comum dos exemplos de formação afiliados a esse nó.

Os nós só são removidos se a árvore podada resultante não tiver um desempenho pior do que o original sobre o conjunto de validação. Isto tem o efeito de que qualquer nó de folha adicionado devido a regularidades coincidentes no conjunto de treino é susceptível de ser podado porque estas mesmas coincidências são improváveis de ocorrer no conjunto de validação. Os nós são podados iterativamente, escolhendo sempre o nó cuja remoção mais aumenta a precisão da árvore de decisão sobre o conjunto de validação. A poda dos nós continua até que uma nova poda seja prejudicial (ou seja, diminui a precisão da árvore em relação ao conjunto de validação).

Na prática, um método bastante bem sucedido para encontrar hipóteses de alta precisão é uma técnica a que chamaremos regra após a poda. Uma variante deste método de poda é utilizado por C4.5 (Quinlan 1993), que é um resultado do algoritmo ID3 original.

A regra pós-pruning envolve os seguintes passos:

1. Inferir a árvore de decisão desde o conjunto de treino, cultivando a árvore até ao treino Os dados são o mais adequados possível e permitem a ocorrência de sobreajustamentos.

2. Converter a árvore aprendida num conjunto equivalente de regras, criando uma regra para cada caminho desde o nó de raiz até um nó de folha.

3. Podar (generalizar) cada regra, removendo quaisquer condições prévias que resultem em melhorando a sua precisão estimada.

4. Ordene as regras podadas pela sua precisão estimada, e considere-as neste ao classificar as instâncias subsequentes.

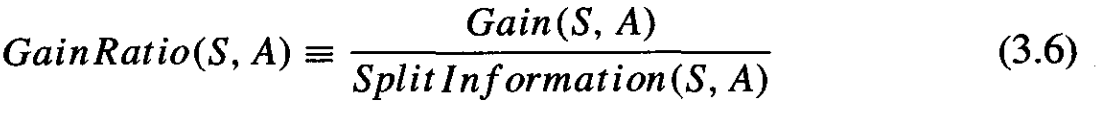
Uma forma de evitar esta dificuldade é seleccionar atributos de decisão com base noutra medida que não o ganho de informação. Uma medida alternativa que tem sido utilizada com sucesso é a taxa de ganho (Quinlan 1986). A medida da razão de ganho penaliza atributos tais como Data, incorporando um termo, chamado de informação dividida, que é sensível à amplitude e uniformidade com que o atributo divide os dados:

Text

Description automatically generated with medium confidence

onde S1 a S, são os subconjuntos c de exemplos resultantes da partição S pelo atributo c-valorizado A. Note que Splitlnfomzation é na realidade a entropia de S com respeito aos valores do atributo A. Isto está em contraste com o nosso anterior usos da entropia, nos quais consideramos apenas a entropia de S no que diz respeito ao atributo alvo cujo valor deve ser previsto pela árvore aprendida.

A medida de Gain Ratio é definida em termos da medida de Ganho anterior, como bem como esta Splitlnfomzation, como se segue



Note que o termo Splitlnfomzation desencoraja a selecção de atributos com muitos valores uniformemente distribuídos. Por exemplo, considerar uma colecção de n exemplos que estão completamente separados pelo atributo A (por exemplo, Data). Neste caso, o O valor da divisão será log, n. Em contraste, um atributo booleano B que divide os mesmos n exemplos terão Splitlnfomzation de 1. se atributos A e B produzem o mesmo ganho de informação, depois claramente B terá uma pontuação mais alta de acordo com a medida Gain Ratio.

3.8 RESUMO E POSTERIOR LEITURA

Os pontos principais deste capítulo incluem:

A aprendizagem em árvore de decisão fornece um método prático para a aprendizagem de conceitos e para a aprendizagem de outras funções de valor discreto. A família de algoritmos ID3 infere as árvores de decisão ao cultivá-las a partir da raiz para baixo, seleccionando avidamente o próximo melhor atributo para cada novo ramo de decisão adicionado à árvore.

O ID3 procura um espaço de hipóteses completo (ou seja, o espaço das árvores de decisão pode representar qualquer função de valor discreto definida sobre instâncias de valor discreto). Evita assim a maior dificuldade associada às abordagens que consideram apenas conjuntos restritos de hipóteses: que a função alvo possa não estar presente no espaço de hipóteses.

O viés indutivo implícito no ID3 inclui uma preferência por árvores mais pequenas; ou seja, a sua procura através do espaço de hipóteses cresce a árvore apenas como grande conforme necessário, a fim de classificar os exemplos de formação disponíveis.

O sobreajustamento dos dados de formação é uma questão importante na aprendizagem da árvore de decisão.

Como os exemplos de formação são apenas uma amostra de todas as instâncias possíveis, é possível adicionar ramos à árvore que melhoram o desempenho na exemplos de formação ao mesmo tempo que diminuem o desempenho em outras instâncias exteriores este conjunto. Os métodos de pós-prancelamento da árvore de decisão são, portanto, importantes para evitar o sobreajustamento na aprendizagem da árvore de decisão (e outras inferências indutivas métodos que empregam um viés de preferência).

Foi desenvolvida uma grande variedade de extensões para o algoritmo básico ID3 por diferentes investigadores. Estes incluem métodos para árvores pós-desramação, manipulação de atributos de valor real, acomodação de exemplos de formação com valores de atributos em falta, refinação incremental de árvores de decisão como nova formação exemplos tornam-se disponíveis, utilizando outras medidas de selecção de atributos para além de informação, e considerando os custos associados aos atributos da instância.

BAYESIAN LEARNING

O raciocínio Bayesiano fornece uma abordagem probabilística para a inferência. Baseia-se no pressuposto de que as quantidades de interesse são regidas por distribuições de probabilidades e que as decisões óptimas podem ser tomadas por raciocínio sobre estas probabilidades juntamente com os dados observados. É importante para o ML porque proporciona uma abordagem quantitativa à ponderação das provas que suportam hipóteses alternativas. O raciocínio Bayesiano fornece a base para algoritmos de aprendizagem que manipulam directamente as probabilidades, bem como uma estrutura para analisar o funcionamento de outros algoritmos que não manipulam explicitamente as probabilidades.

Os métodos de aprendizagem Bayesianos são relevantes para o nosso estudo da aprendizagem mecânica por duas razões diferentes. Primeiro, os algoritmos de aprendizagem Bayesianos que calculam probabilidades explícitas para hipóteses, tais como o classificador Bayes ingénuo, estão entre as abordagens mais práticas para certos tipos de problemas de aprendizagem.

A segunda razão pela qual os métodos Bayesianos são importantes para o nosso estudo da aprendizagem mecânica é que fornecem uma perspectiva útil para a compreensão de muitos algoritmos de aprendizagem que não manipulam explicitamente as probabilidades.

Na aprendizagem mecânica estamos muitas vezes interessados em determinar a melhor hipótese

de algum espaço H, tendo em conta os dados de formação D observados. Uma forma de especificar o que queremos dizer com a melhor hipótese é dizer que exigimos a hipótese mais provável, dados os dados D mais qualquer conhecimento inicial sobre as probabilidades anteriores das várias hipóteses no teorema de H. Bayes fornece um método directo para calcular tais probabilidades. Mais precisamente, o teorema de Bayes fornece uma forma de calcular a probabilidade de uma hipótese com base na sua probabilidade anterior, as probabilidades de observar vários dados dada a hipótese, e os próprios dados observados.

Para definir com precisão o teorema de Bayes, vamos primeiro introduzir uma pequena notação. Nós deve escrever P(h) para denotar a probabilidade inicial que a hipótese h tem, antes de nós observaram os dados de formação. P(h) é muitas vezes chamada a priori-probabilidade de h e pode reflectir qualquer conhecimento de base que tenhamos sobre a hipótese de h ser um correcto hipótese. Se não tivermos esse conhecimento prévio, então podemos simplesmente atribuir a mesma probabilidade anterior a cada hipótese de candidato. Da mesma forma, iremos escrever P(D) para indicar a probabilidade prévia de que os dados de formação D sejam observados (ou seja a probabilidade de D dada a ausência de conhecimento sobre qual a hipótese que se mantém). A seguir, escreveremos P(D1h) para denotar a probabilidade de observar os dados D, dado que alguns mundo em que a hipótese h se mantém. Mais genericamente, escrevemos P(xly) para denotar a probabilidade de x dado y. Em problemas de aprendizagem de máquinas, estamos interessados em a probabilidade P (h 1 D) que h detém dados de formação observados D. P (h 1 D) é chamada a posterior probabilidade do h, porque reflete a nossa confiança que o h detém depois de termos visto os dados de formação D. Repare na probabilidade posterior P(h1D) reflecte a influência dos dados de formação D, em contraste com a probabilidade anterior P(h) , que é independente de D.

O teorema de Bayes é a base dos métodos de aprendizagem Bayesianos porque fornece uma forma de calcular a probabilidade posterior P(hlD), a partir da probabilidade anterior P(h), juntamente com P(D) e P(D(h).

A picture containing background pattern

Description automatically generated

Como se poderia esperar intuitivamente, P(h ID) aumenta com P(h) e com P(D|h) de acordo com o teorema de Bayes. Também é razoável ver que P(hl D) diminui à medida que P(D) aumenta, porque quanto mais provável é que D seja observado independentemente de h, menos evidência D fornece em apoio de h.

O classificador ingénuo Bayes aplica-se a tarefas de aprendizagem onde cada instância x é descrita por uma conjunção de valores de atributos e onde a função alvo f (x) pode assumir qualquer valor de algum conjunto finito V. É fornecido um conjunto de exemplos de formação da função alvo, e é apresentada uma nova instância, descrita pelo conjunto de valores de atributos (al, a2...a,). O aprendente é convidado a prever o valor-alvo, ou classificação, para esta nova instância.

A picture containing text

Description automatically generated

Para resumir, o método de aprendizagem ingénuo Bayes envolve uma etapa de aprendizagem em que os vários termos P(vj) e P(ai Jvj) são estimados, com base nas suas frequências sobre os dados de formação. O conjunto destas estimativas corresponde à hipótese. Esta hipótese é então utilizada para classificar cada nova instância através da aplicação a regra da Equação (6.20).

Os pontos principais deste capítulo incluem:

Os métodos Bayesianos fornecem a base para métodos de aprendizagem probabilísticos que acomodar (e exigir) conhecimento sobre as probabilidades anteriores de hipóteses alternativas e sobre a probabilidade de observar vários dados dada a hipótese. Os métodos Bayesianos permitem atribuir uma probabilidade posterior a cada hipótese candidata, com base nestes pressupostos priores e nos dados observados.

Os métodos Bayesianos podem ser utilizados para determinar a hipótese mais provável dada a hipótese do máximo a posteriori (MAP) dos dados. Esta é a hipótese hipótese ideal no sentido de que nenhuma outra hipótese é mais provável.

O classificador óptimo da Bayes combina as previsões de todas as hipóteses alternativas, ponderadas pelas suas probabilidades posteriores, para calcular as mais classificação provável de cada nova instância.

O classificador ingénuo Bayes é um método de aprendizagem Bayesiano que foi encontrado para ser útil em muitas aplicações práticas. Chama-se "ingénuo" porque incorpora a hipótese simplificadora de que os valores dos atributos são condicionalmente independente, dada a classificação da instância. Quando esta suposição é cumprida, o classificador ingénuo da Bayes produz a classificação MAP. Mesmo quando este pressuposto não é cumprido, como no caso de aprender a classificar o texto, o O classificador Bayes, ingénuo, é frequentemente bastante eficaz. As redes de crenças Bayesianas proporcionam uma representação mais expressiva de conjuntos de independência condicional. pressupostos entre os subconjuntos dos atributos.

O quadro do raciocínio Bayesiano pode fornecer uma base útil para analisar certos métodos de aprendizagem que não aplicam directamente o teorema de Bayes. Por exemplo, sob certas condições, pode ser demonstrado que a minimização do erro ao quadrado quando a aprendizagem de uma função alvo com valor real corresponde a calcular a hipótese de máxima probabilidade.

O princípio do comprimento mínimo da descrição recomenda a escolha da hipótese que minimiza o comprimento da descrição da hipótese mais a descrição comprimento dos dados dada a hipótese. O teorema de Bayes e os resultados básicos da teoria da informação podem ser utilizados para fornecer uma justificação para este princípio.

Em muitas tarefas práticas de aprendizagem, algumas das variáveis relevantes da instância pode ser inobservável. O algoritmo EM proporciona uma abordagem bastante geral à aprendizagem na presença de variáveis inobserváveis. Este algoritmo começa com uma hipótese inicial arbitrária. Em seguida, calcula repetidamente a valores esperados das variáveis ocultas (assumindo a hipótese actual está correcto), e depois recalcula a hipótese de máxima probabilidade (assumindo que as variáveis ocultas têm os valores esperados calculados pela primeira passo). Este procedimento converge para uma hipótese local de máxima probabilidade, juntamente com os valores estimados para as variáveis ocultas.

k-NEAREST NEIGHBOR LEARNING

O método mais básico baseado na instância é o algoritmo k-NEAREST NEIGHBOR. Este algoritmo assume que todas as instâncias correspondem a pontos no espaço n-dimensional 8". Os vizinhos mais próximos de uma instância são definidos em termos da distância Euclidiana padrão. Mais precisamente, que uma instância arbitrária x seja descrita pelo vector de característica

onde ar (x) denota o valor do atributo r da instância x. Então a distância entre duas instâncias xi e xj é definida como d(xi, xj), onde

Na aprendizagem do vizinho mais próximo, a função alvo pode ser discretamente valorizada ou com valor real. Consideremos primeiro a aprendizagem de funções-alvo discretas e valorizadas do forma f : W -+ V, onde V é o conjunto finito {vl, . . . v,}.

O algoritmo k-NEAREST NEIGHBOR para aproximar uma função alvo de valor discreto é dado na Tabela 8.1. Como aí se mostra, o valor f (x,) devolvido por este algoritmo como a sua estimativa de f (x,) é apenas o valor mais comum de f entre os exemplos de treino k mais próximos de x,. Se escolhermos k = 1, então o algoritmo 1-NEAREST NEIGHBOR atribui a f(x,) o valor f (xi) onde xi é a instância de treino mais próxima de x,. Para valores maiores de k, o algoritmo atribui o valor mais comum entre os exemplos de treino de k mais próximos.

**8.7 RESUMO E POSTERIOR LEITURA**

**Os pontos principais deste capítulo incluem:**

Os métodos de aprendizagem baseados na instância diferem de outras abordagens de aproximação de funções porque atrasam o processamento de exemplos de formação até deve rotular uma nova instância de consulta. Como resultado, não precisam de formar uma hipótese de toda a função do alvo sobre todo o espaço da instância, independentemente da instância de consulta. Em vez disso, podem formar uma aproximação à função alvo para cada instância de consulta.

As vantagens dos métodos baseados na instância incluem a capacidade de modelar funções-alvo complexas através de uma colecção de aproximações locais menos complexas e o facto de a informação presente nos exemplos de formação nunca se perder (porque os exemplos em si são armazenados explicitamente). As principais dificuldades práticas incluem a eficiência da etiquetagem de novas instâncias (todo o processamento é feito no momento da consulta e não antecipadamente), dificuldades em determinar uma métrica de distância adequada para recuperar instâncias "relacionadas" (especialmente quando os exemplos são representados por descrições simbólicas complexas), e o impacto negativo de características irrelevantes na métrica de distância.

k-NEAREST NEIGHBOR é um algoritmo baseado em instâncias para aproximar funções alvo real valorizadas ou discretas, assumindo que as instâncias correspondem a pontos num espaço Euclidiano n-dimensional. O valor da função alvo para uma nova consulta é estimado a partir dos valores conhecidos dos exemplos de treino k mais próximos.

Regressão ponderada localmente são uma generalização do k-NEAREST NEIGHBOR em que é construída uma aproximação local explícita à função alvo para cada instância de consulta. A aproximação local à função alvo pode ser baseada numa variedade de formas funcionais, tais como funções constantes, lineares ou quadráticas, ou em funções de núcleo espacialmente localizadas.

As redes de função de base radial (RBF) são um tipo de rede neural artificial construída a partir de funções de kernel espacialmente localizadas. Estas podem ser vistas como uma mistura de abordagens baseadas na instância (influência espacialmente localizada de cada função de kernel) e rede neural abordagens (forma-se uma aproximação global à função alvo no momento da formação em vez de uma aproximação local no momento da consulta). As redes de funções de base radial têm sido utilizadas com sucesso em aplicações tais como a interpretação de cenas visuais, nas quais a assunção de influências espacialmente locais é bem justificada.

REFORÇO DA APRENDIZAGEM

O reforço da aprendizagem aborda a questão de como um agente autónomo que sente e atua no seu ambiente pode aprender a escolher ações óptimas para alcançar os seus objectivos. Este problema muito genérico abrange tarefas tais como aprender a controlar um robô móvel, aprender a otimizar operações em fábricas, e aprender a jogar jogos de tabuleiro. Cada vez que o agente executa uma acção no seu ambiente, um treinador pode fornecer uma recompensa ou penalização para indicar a desejabilidade do estado resultante. Por exemplo, ao treinar um agente para jogar um jogo, o treinador pode fornecer uma recompensa positiva quando o jogo é ganho, uma recompensa negativa quando é perdido, e uma recompensa zero em todos os outros estados. A tarefa do agente é aprender com esta recompensa indireta, retardada, a escolher sequências de ações que produzam a maior recompensa cumulativa. Este capítulo centra-se num algoritmo chamado aprendizagem Q que pode adquirir estratégias de controlo óptimas a partir de recompensas atrasadas, mesmo quando o agente não tem conhecimento prévio dos efeitos das suas acções sobre o ambiente. Os algoritmos de aprendizagem de reforço estão relacionados com algoritmos de programação dinâmica frequentemente utilizados para resolver problemas de optimização.

Considere a construção de um robô de aprendizagem. O robô, ou agente, tem um conjunto de sensores para observar o estado do seu ambiente, e um conjunto de acções que pode realizar para alterar este estado. Por exemplo, um robô móvel pode ter sensores tais como uma câmara e sonares, e ações tais como "avançar" e "virar". A sua tarefa é aprender uma estratégia de controlo, ou política, para escolher acções que atinjam os seus objectivos. Por exemplo, o robô pode ter o objetivo de acoplar o seu carregador de bateria sempre que o seu nível de bateria estiver baixo.

Isto inclui, por exemplo, problemas de optimização do fabrico em que uma sequência de acções de fabrico deve ser escolhida, e a recompensa a ser maximizada é o valor dos bens produzidos menos os custos envolvidos. Inclui problemas de programação sequencial como a escolha dos táxis a enviar para os passageiros numa grande cidade, onde a recompensa a ser maximizada é uma função do tempo de espera dos passageiros e dos custos totais de combustível da frota de táxis. Em geral, estamos interessados em qualquer tipo de agente que deve aprender a escolher acções que alteram o estado do seu ambiente e onde uma função de recompensa cumulativa é utilizada para definir a qualidade de uma dada sequência de acções. Dentro desta classe de problemas, iremos considerar cenários específicos, incluindo cenários em que as acções têm resultados determinísticos ou não determinísticos, e cenários em que o agente tem ou não tem conhecimento prévio sobre os efeitos das suas ações sobre a ambiente.

Diagram

Description automatically generated