

# 1 Aprendizado de máquina

## 1.1 Definição

Aprendizado de máquina é uma subárea de Inteligência Artificial que agrupa conhecimentos sobre algoritmos e técnicas que permitam que um programa melhore sua performance a partir de dados. Mais formalmente:

Um programa aprende a partir de uma experiência  $E$ , com relação a uma classe de tarefas  $T$  e a uma medida de performance  $P$ , se sua performance em tarefas da classe  $T$ , medida por  $P$ , melhora com a experiência  $E$ . (MITCHELL, 1997, p.2, tradução nossa)

adicionar exemplos de aplicação

## 1.2 Modelagem

Para exemplificar a modelagem de um programa com uma abordagem de aprendizado de máquina, (MITCHELL, 1997) apresenta uma sequência de passos para desenvolver um programa que aprenda a jogar xadrez, a ser utilizado para disputar um campeonato mundial de xadrez, tendo sua performance medida pela frequência de partidas que vencer.

### Escolha da experiência

O primeiro passo consiste na escolha da experiência a partir da qual o programa irá aprender, denominada experiência de treinamento. (MITCHELL, 1997) apresenta três atributos da experiência de treinamento que devem ser levados em consideração por impactarem no sucesso do aprendizado: feedback, nível de controle sobre os exemplos de treinamento e quão bem a distribuição dos exemplos de treinamento representam a distribuição dos exemplos sobre os quais a performance final do programa será mensurada.

O atributo feedback representa quão direta é a informação fornecida pela experiência para o problema em questão, podendo assumir os valores feedback direto e feedback indireto. Por exemplo, a tupla estado do tabuleiro e o melhor movimento possível a partir desse estado é classificada como experiência de feedback direto: o programa irá atuar realizando movimentos e esse tipo de experiência informa diretamente qual o melhor movimento a ser executado. Já a tupla sequência de movimentos de uma partida e seu resultado final é classificada como experiência de feedback indireto: o resultado final da partida não fornece informação direta sobre a influência dos movimentos que nela foram executados. A atividade de determinar o grau de influência que elementos de uma experiência de feedback indireto têm sobre o resultado é denominada credit assignment.

Com relação ao nível de controle sobre os exemplos de treinamento, a experiência pode ser classificada como experiência selecionada por especialista, experiência sugerida pelo programa e analisada por um especialista e experiência selecionada e analisada pelo programa. Por exemplo, a experiência será do tipo selecionada por especialista se foi

é necessário traduzir credit assignment

selecionada por um jogador experiente de xadrez, que selecionou estados de tabuleiro e indicou que melhores movimentos poderiam ser feitos, a partir desses estados; será do tipo sugerida pelo programa e analisada por um especialista se o próprio programa selecionar estados de tabuleiro para serem analisadas por um jogador experiente de xadrez; será do tipo selecionada e analisada pelo programa se o programa utilizar o resultado de partidas que disputar consigo mesmo.

Com relação a quão bem a a distribuição dos exemplos de treinamento representam a distribuição dos exemplos sobre os quais a performance final do programa será mensurada, a experiência pode ser classificado como representativa, se a distribuição de seus exemplos representar a distribuição dos exemplos com os quais o programa efetivamente será utilizado, e não representativo, caso contrário. Por exemplo, a experiência é não representativa caso esteja limitada ao conjunto de partidas de apenas um jogador: considerando que o programa será utilizado em um campeonato mundial, do qual participam jogadores diversos, com estilos de jogo diversos, é esperado que o programa, treinado com essa experiência, depare com estados de tabuleiro que não encontrou no treinamento. (MITCHELL, 1997) ressalta que muito da teoria de aprendizado de máquina depende da assunção de que a experiência utilizada no treinamento é representativa.

Para o problema em análise, (MITCHELL, 1997) escolhe como experiência de treinamento um conjunto de partidas jogadas pelo programa contra ele mesmo.

## Escolha da função alvo

O próximo passo é a escolha do tipo de conhecimento que deverá ser aprendido, representado por uma função denominada função alvo, e como ele será utilizado pelo programa. Considerando que o programa irá atuar como um jogador de xadrez, uma possível função a ser considerada é uma cujo domínio seja o conjunto de estados de tabuleiro e que retorne o melhor movimento a partir do estado de tabuleiro informado. Esse tipo de conhecimento depende da assunção de que, dado um estado de tabuleiro, existe um melhor movimento a ser executado. O problema de aprendizado dessa função depende, portanto, do problema de determinar quão um movimento influencia no resultado final de uma partida.

discutir

qual a escolhida no Mitchell? Lembrar de apresentar o exemplo

Outro exemplo de função alvo, a ser utilizada no exemplo, é uma que tenha como domínio o conjunto de estados de tabuleiro e retorne um número real, indicando quão bom o estado de tabuleiro informado é. O programa irá jogar verificando qual estado de tabuleiro maximiza o valor da função, considerando o conjunto de estados de tabuleiro que podem ser alcançados a partir do estado atual do tabuleiro e de todas jogadas válidas.

## Escolha de uma representação para a função alvo

Após a escolha do tipo de conhecimento que deverá ser aprendido, é necessário definir como esse conhecimento será representado. Por exemplo, a função que associa um estado de tabuleiro a um número real pode assumir diversas formas: pode ser uma matriz contendo uma célula com um número real para cada estado de tabuleiro possível; um conjunto de regras que associe atributos do estado do tabuleiro a números reais;

uma função polinomial de atributos do estado do tabuleiro em números reais etc. Para dar continuidade ao detalhamento dos passos, (MITCHELL, 1997) escolhe a seguinte representação de função, denominada  $V$ :

$$V(t) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

Os coeficientes  $w_0$  a  $w_6$  são parâmetros do programa, a serem ajustados no aprendizado. As variáveis  $x_1$  a  $x_6$  possuem as seguintes definições:

- $x_1$ : número de peças pretas no tabuleiro
- $x_2$ : número de peças brancas no tabuleiro
- $x_3$ : número de reis pretos no tabuleiro
- $x_4$ : número de reis brancos no tabuleiro
- $x_5$ : número de peças pretas ameaçadas por peças brancas no tabuleiro
- $x_6$ : número de peças brancas ameaçadas por peças pretas no tabuleiro

Resumindo os passos até aqui realizados, temos:

- Tarefa: jogar xadrez
- Medida de performance: frequência de partidas do campeonato mundial de xadrez ganhas
- Experiência de treinamento: partidas disputadas pelo programa contra ele mesmo
- Função alvo:  $V : EstadosDeTabuleiro \rightarrow \mathbb{R}$
- Representação da função alvo:  $\bar{V}(t) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$

## Escolha de um algoritmo de aproximação

O próximo passo consiste na escolha de um algoritmo que irá realizar o treinamento da função, isto é, que, a partir da experiência de treinamento, irá ajustar seus coeficientes a fim de aproximá-la da função alvo. Para tanto, é necessário um conjunto de exemplos de treinamento composto por tuplas,  $\langle b, V_{treino}(b) \rangle$ , de estado de tabuleiro e o valor da função alvo para esse estado, respectivamente,  $b$  e  $V_{treino}(b)$ . Por exemplo:

$$\langle \langle x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0 \rangle, +100 \rangle$$

Para o desenvolvimento dos exemplos de treinamento, é necessário estimar um valor para cada estado de tabuleiro alcançado nas partidas disputadas pelo programa contra ele mesmo (experiência de treinamento escolhida), indicando quão bom aquele estado de tabuleiro é. A abordagem apresentada consiste em atribuir valores para os últimos estados de tabuleiro das partidas e ir propagando esses valores para os estados intermediários através da regra:

$$V_{treino}(b) \leftarrow \bar{V}(t)(Sucessor(b)) \quad (1.1)$$

Determinar os valores dos estados de tabuleiros utilizados no treinamento é tarefa fácil para os estados finais: pode-se definir dois valores,  $max < min$ , e atribuir aos estados finais de vitória o valor  $max$  e aos de derrota o valor  $min$ . Já a determinação dos valores dos tabuleiros intermediários não é tão simples. O fato de uma partida ter sido ganha não implica necessariamente que todos os estados de tabuleiro nela percorridos devem receber um valor alto. Para definir tais valores pode-se utilizar uma regra de estimação. Por exemplo:

Essa regra atribui a um estado de tabuleiro de treinamento o valor que a função alvo estimada retorna para o estado de tabuleiro após a próxima jogada do oponente. Apesar de parecer estranho fazer uso de  $V$ , a função que se está estimando, para determinar os valores a serem utilizados para refiná-la, intuitivamente essa abordagem parece fazer sentido, por atribuir a um estado de tabuleiro um valor que é função de um estado que foi possível alcançar a partir dele.

Após a determinação dos valores dos estados de tabuleiros contidos na experiência de treinamento, o conjunto de dados de treinamento pode ser utilizado. Para estimar a função alvo a partir dos dados de treinamento, dada a representação da função escolhida, é necessário agora determinarmos seus pesos,  $w_0$  a  $w_6$ . Para tanto, primeiro definimos como mensurar quão bem a função estimada se adequa aos dados de treinamento. Uma medida comum é o erro quadrático, assim definido:

$$E = \sum_{(b, V_{treino}(b)) \in \text{dados de treinamento}} (V_{treino}(b) - V(b))^2$$

O problema de estimar a função alvo pode ser modelado então como o problema de encontrar os pesos  $w_0$  a  $w_6$  que minimizem o erro quadrático sobre os dados de treinamento. Um dos algoritmos que incrementalmente ajusta os pesos aos dados de treinamento, minimizando o erro quadrático é o Método dos Mínimos Quadrados. Esse algoritmo funciona ajustando, para cada dado de treinamento, os pesos da função na direção que minimiza o erro quadrático. Para tanto, atualiza iterativamente, para cada dado de treinamento, os pesos da função, incrementando com um valor proporcional a  $(V_{treino}(b) - V(b))$ : se o valor da função aplicado ao dado de treinamento,  $V(b)$ , é igual ao valor do dado de treinamento,  $V_{treino}(b)$ , o incremento será nulo; se o valor da função é maior que o do dado de treinamento, o incremento será negativo, fazendo com que, para o dado de treinamento em questão, o valor da função diminua; se o valor da função é menor que o do dado de treinamento, o incremento será positivo, fazendo com que, para o dado de treinamento em questão, o valor da função aumente.

## Conclusão

Nessa sequência de passos podem ser identificados quatro elementos de um programa que aprende:

- Sistema de performance: elemento responsável pela utilização do conhecimento aprendido para resolver uma tarefa. No exemplo, será responsável por determinar qual a próxima jogada, dado um estado de tabuleiro, utilizando a função que foi aprendida.
- Crítico: elemento responsável por receber como entrada um dado de treinamento e informar a que valor deve ser associado. No exemplo, o crítico é representado pela equação 1.1.

- Generalizador: elemento responsável por receber como entrada um conjunto de dados de treinamento e retornar uma função estimativa de uma função alvo. No exemplo, o generalizador é o Método dos Mínimos Quadrados.
- Representação da função alvo: elemento que define a estrutura da função que será utilizada como estimativa da função alvo. No exemplo, foi utilizada como representação uma combinação linear.

Outras configurações para esses elementos foram desenvolvidas. Por exemplo, como representação da função alvo pode-se utilizar um grafo em estrutura de árvore, denominado árvore de decisão. Cada nó seu que não seja folha possui uma regra que associa um dado a um de seus nós filhos. Os nós folhas são associados a um valor. Seu funcionamento consiste em apresentar um dado à regra de um nó, inicialmente o nó raiz, e recursivamente aplicar esse procedimento ao nó filho ao qual a regra associa o dado, até que seja alcançado um nó folha, cujo resultado associado é então retornado como o valor da função.

## 1.3 Modelos

definir o que quer dizer por modelo

representation + evaluation + optimization ([DOMINGOS, 2012](#))

## 1.4 Aplicações

melhorar

De acordo com (??), o conhecimento sobre aprendizado de máquina pode ser aplicado a diversas áreas, como, por exemplo, a reconhecimento de voz; a visão computacional, sendo utilizado no desenvolvimento de sistemas de reconhecimento facial; a controle de robôs.



## Referências

DOMINGOS, P. A few useful things to know about machine learning. *Communications of the ACM*, ACM, v. 55, n. 10, p. 78–87, 2012.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.