Inteligência Artificial

Raoni F. S. Teixeira

Aula 5 - Aprendizado - Regressão Linear

1 Introdução

Essa parte do curso aborda agentes que aprendem a partir de dados, sem depender de instruções explícitas em sua programação Nesse caso, a função do agente é um modelo que captura padrões dos dados e os utiliza para classificar objetos e fazer previsões. Dependendo de como o algoritmo utiliza os dados, o aprendizado é chamado de supervisionado, não-supervisionado ou por reforço. O tema dessa aula é a técnica de aprendizado supervisionado regressão linear.

2 Aprendizado Supervisionado

Um algoritmo de aprendizado supervisionado recebe um conjunto de dados de treinamento com uma coleção de pares de entrada e saída e *aprende* uma função h que prevê saídas para entradas ainda não observadas. A Figura 1 mostra um esquema de criação de um modelo de aprendizado. Os componentes principais são o conjunto de treinamento, o algoritmo de aprendizagem e o modelo (função h).

Formalmente, sejam $\mathcal{X}$ e $\mathcal{Y}$ conjuntos de entrada e saída, respectivamente, e $\mathcal{R}$ a relação que os associa. O *conjunto de treinamento* $\mathcal{T} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(M)}, y^{(M)})\}$ é um subconjunto de $\mathcal{R}$ e a função $h : \mathcal{X} \mapsto \mathcal{Y}$ modela os padrões de $\mathcal{R}$, aproximando $y^{(i)}$ a partir de $x^{(i)}$.

Cada par $(x^{(i)}, y^{(i)})$ do conjunto de treinamento é chamado de *exemplo de treinamento* e as variáveis $x^{(i)}$ e $y^{(i)}$ são chamadas entrada e saída. O índice (i) sobrescrito indica a posição da variável na listagem do conjunto e não é um sinal de exponenciação. x é um vetor de tamanho n. Quando a variável de saída y é contínua, o problema de aprendizado é uma regressão (e.g. previsão do preço de casas ilustrado na Figura 2). Quando y é discreta, o problema é uma classificação (tema das próximas aulas).

A Figura 2 mostra um exemplo de relação entre área e preço de imóveis em uma cidade. Os pontos indicam o conjunto de treinamento e a reta é o gráfico da função h aprendida levando

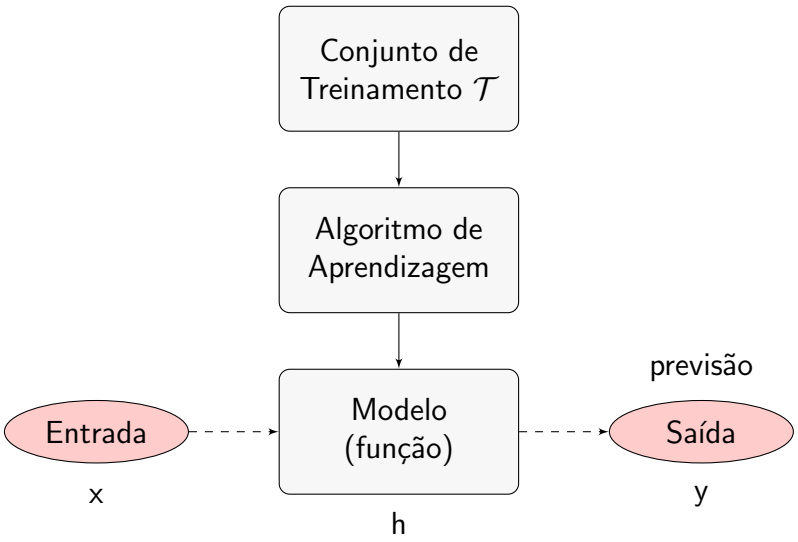


Figura 1: Aprendizado Supervisionado.

em conta esse conjunto. A reta é uma aproximação do preço com um *erro*. Quanto menor o erro, melhor é a previsão.

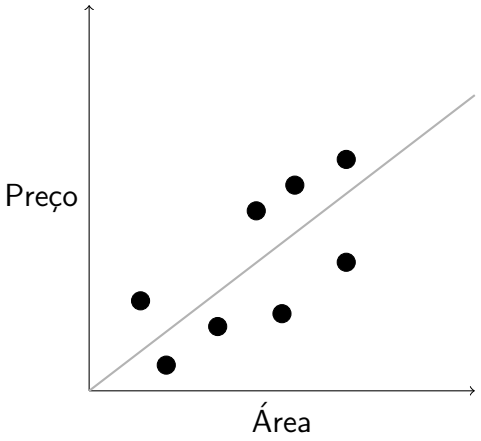


Figura 2: Relação entre área e preço de imóveis em uma cidade.

3 Regressão Linear

O esquema de aprendizado supervisionado da Figura 1 depende de uma função h. Nessa aula, h é uma função linear de x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n, \tag{1}$$

em que $\theta_0, \theta_1, \ldots \theta_n$ são *parâmetros* (também chamados de *pesos*) que controlam o mapeamento linear de $\mathcal{X}$ para $\mathcal{Y}$.

Se tomarmos $x_0 = 1$, podemos reescrever a função h como o produto vetorial a seguir:

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x. \tag{2}$$

O objetivo da regressão linear é escolher os valores de $\theta_0, \theta_1, \ldots, \theta_n$ de modo que $h_\theta(x^{(i)})$ se aproxime ao máximo de $y^{(i)}$ para cada amostra de treinamento $(x^{(i)}, y^{(i)})$.

A função de custo $J(\theta)$ mede a proximidade entre $h_\theta(x)$ e y em cada amostra de treinamento:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2. \tag{3}$$

Um valor menor de $J(\theta)$ indica um modelo melhor. Note que o valor de $J(\theta)$ é não negativo e será zero apenas se o modelo $h_\theta(x^{(i)})$ prever exatamente $y^{(i)}$ para todas as amostras do conjunto de treinamento.

A Figura 3 mostra duas retas para o conjunto de treinamento $\mathcal{T} = \{(1,1), (2,2), (3,3)\}$ e função $h_\theta(x) = \theta_0 + \theta_1 x_1$. Valores distintos de θ geram retas com custos variados. As retas das Figura 3a e Figura 3b tem custo 1.75 e 0.0. O objetivo do algoritmo é encontrar o θ para qual $J(\theta)$ é mínimo ($J(\theta) = 0$ nesse caso).

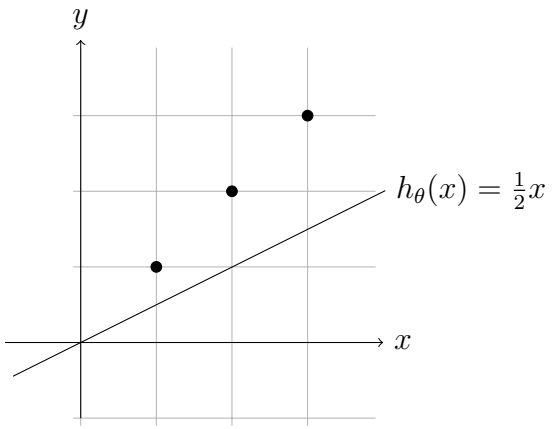
Assim, o algoritmo de aprendizado deve resolver a seguinte equação:

$$\underset{\theta}{\text{argmin}}\ J(\theta). \tag{4}$$

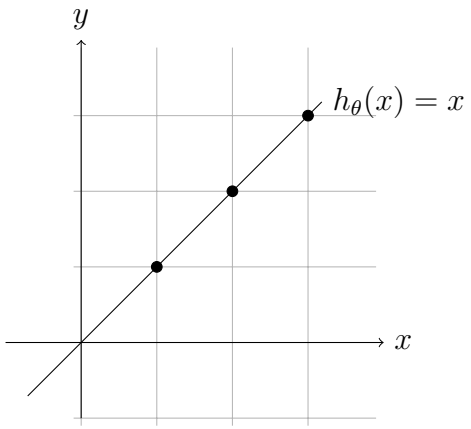
4 Método do Gradiente

O método do gradiente otimiza os parâmetros $\theta_j (j = 0, \ldots, n)$ iniciando com valores aleatórios e ajustando-os iterativamente com base no gradiente da função de custo:

$$\begin{aligned} &\theta_j^0 \leftarrow \text{RANDOM}() && \\ &\theta_j^k \leftarrow \theta_j^{k-1} - \alpha \frac{\partial J(\theta)}{\partial \theta_j}, && \forall \text{ iteração } k, k \in 1, \ldots, K, \end{aligned} \tag{5}$$



(a) $\theta_0 = 0$, $\theta_1 = \frac{1}{2}$ e $J(\theta) = 1.75$



(b) $\theta_0 = 0$, $\theta_1 = 1$ e $J(\theta) = 0$

Figura 3: Diferentes parâmetros θ e com diferentes custos.

em que α é chamado de *taxa de aprendizado*, k é o índice da iteração e o termo mais à direita é a derivada parcial de J.

A derivada parcial de J calcula como o erro de previsão afeta o parâmetro θ_j:

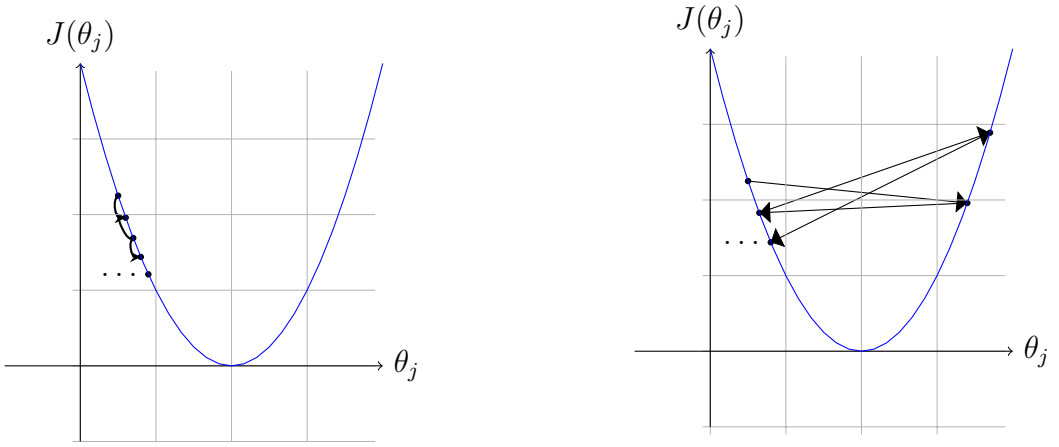
$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2 \cdot \frac{1}{2}(h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j}(h_\theta(x) - y) && \text{(regra da cadeia)} \\
&= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \sum_{i=0}^{n} \Big(\theta_i x_i - y\Big) \\
&= (h_\theta(x) - y) x_j.
\end{aligned}$$

Com base nessa derivada, ajustamos θ_j para cada amostra do treinamento $(x^{(i)}, y^{(i)})$, o parâmetro θ_j:

$$\theta_j^k \leftarrow \theta_j^{k-1} + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}. \tag{6}$$

Esse ajuste depende do erro atual. Previsões mais precisas levam a mudanças menores em θ_j.

A taxa de aprendizagem α controla a intensidade da alteração. Quanto maior o α, maior será alteração. A Figura 4 mostra a diferença entre o valores de α. Se α for muito pequeno (Figura 4a), então o algoritmo irá fazer muitos passos até encontrar o mínimo. Por outro lado, se o valor de α for muito alto (Figura 4b), o algoritmo pode *pular* e não encontrar o mínimo. Na prática, α é uma variável de configuração (hiper-parâmetro) que deve ser definida experimentalmente.



(a) Quando α é pequeno, o algoritmo é lento. (b) Quando α é grande, o algoritmo não converge.

Figura 4: Impacto do tamanho do passo (α) na descida do gradiente.

Há duas maneiras de aplicar a regra da Equação 6 em um conjunto com mais de uma amostra: a) considerar, para cada atualização, o conjunto inteiro — atualização em lote e b) atualizar o θ para cada amostra do treinamento — atualização estocástica. Os algoritmos 1 e 2 apresentam os pseudo-códigos dessas duas abordagens.

repeat
  for $j \in \{0, \ldots, n\}$ **do**
    $\theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$
until *convergence*;

Algoritmo 1: Método do Gradiente em lote

for $i \in \{1, \ldots, m\}$ **do**
  for $j \in \{0, \ldots, n\}$ **do**
    $\theta_j \leftarrow \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$

Algoritmo 2: Método do Gradiente estocástico

Como a abordagem em lote precisa percorrer todo o conjunto de treinamento antes de atualizar uma única vez os parâmetros, a descida gradiente estocástica (Algoritmo 2) é abordagem preferida quando o conjunto de treinamento é grande.

5 Solução Analítica

Embora o método do gradiente seja amplamente utilizado para otimizar essa função, existe uma solução analítica dada por:

$$\theta = (X^T X)^{-1} X^T y,$$

em que X^T é a transposta de X, e $(X^T X)^{-1}$ é o inverso da matriz $X^T X$.

Essa expressão fornece diretamente os valores de θ que minimizam o erro da previsão. No entanto, a aplicação prática dessa solução é limitada pela necessidade de calcular o inverso da matriz, o que é computacionalmente intensivo para grandes conjuntos de dados. Essa abordagem é útil para conjuntos de dados pequenos e onde o cálculo matricial seja viável, evitando a necessidade de iterações como no método do gradiente.

Exercícios

1. Escreva a função de custo da Equação 3 na forma vetorial.
2. Mostre que $\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$ é o gradiente $\frac{\partial J(\theta)}{\partial \theta_j}$.
3. Dado um conjunto de dados X e um vetor de saídas y, derive a expressão da solução analítica para a regressão linear. Explique as condições sob as quais essa solução é aplicável e por que o cálculo do inverso da matriz $X^T X$ pode ser problemático em conjuntos de dados grandes ou mal condicionados.

   Dica: A solução analítica é obtida resolvendo a derivada da função de custo em relação a θ e encontrando o ponto de mínimo global.