Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 8, n. 1, 2021.

Trabalho apresentado no XL CNMAC, Evento Virtual - Co-organizado pela Universidade do Mato Grosso do Sul (UFMS).

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Aprendizado PAC e problemas de classificação

Lucas F. F. Ignacio[1]
ICEx/UFF, Volta Redonda, RJ
Marina S. D. de Freitas[2]
ICEx/UFF, Volta Redonda, RJ
Alan P. de Paula[3]
ICEx/UFF, Volta Redonda, RJ

O presente trabalho tem como objetivo formalizar matematicamente a noção de aprendizado para problemas de classificação, um subcampo da teoria do aprendizado supervisionado. Para isso, será feita uma exposição da teoria conhecida como aprendizado PAC.

Um problema de aprendizado supervisionado é composto por um conjunto de entrada $\mathcal{X}$ e um conjunto de saída $\mathcal{Y}$, onde seus dados são pares $(x, y) \in \mathcal{X} \times \mathcal{Y}$. O algoritmo de aprendizado recebe como entrada uma amostra de treino $S = \{(x_1, y_1), ..., (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$. Assumimos que todas as instâncias da amostra são obtidas de forma independente e identicamente distribuídas de acordo com uma distribuição de probabilidade conjunta D sobre $\mathcal{X} \times \mathcal{Y}$. A partir disto, o algoritmo deve retornar uma função, ou hipótese, $h : \mathcal{X} \to \mathcal{Y}$ capaz de rotular pontos do domínio que estejam fora de S.

Dizemos que um problema de aprendizado supervisionado é um problema de classificação quando o conjunto $\mathcal{Y}$ é um conjunto discreto. Neste caso, a partir do processo de treinamento, dado um vetor em $\mathcal{X}$, busca-se predizer a qual classe em $\mathcal{Y}$ ele pertence. Dada uma função de perda l, que a cada hipótese h e $z_i = (x_i, y_i)$ associa o erro de predição, temos duas maneiras de quantificar a eficiência de um algoritmo de aprendizado: o risco empírico

$$L_S(h) := \frac{1}{m} \sum_{i=1}^{m} l(h, z_i),$$

sendo o erro médio sobre uma amostra S e o risco, ou erro de generalização

$$L_D(h) := \mathop{\mathbb{E}}_{Z \sim D}[l(h, Z)],$$

que é o erro esperado baseado na distribuição D.

No campo de aprendizado de máquina, estamos interessados em hipóteses que possuam baixo risco. Como o algoritmo não tem acesso à distribuição de probabilidade D, que gera as amostras do problema, L_D não pode ser calculado diretamente. No entanto, pela Lei Forte dos Grandes Números, pode-se utilizar o risco empírico como sua aproximação. É importante destacar que uma hipótese que possui baixo risco empírico não necessariamente possui um risco pequeno. Assim, ao invés de deixar que o algoritmo selecione qualquer hipótese, a ele é fornecida uma classe de hipóteses, o que é chamado de princípio de minimização do risco empírico com viés indutivo, e fundamenta a definição de aprendizado PAC.

O termo PAC é abreviatura em inglês para Probably Aproximately Correctly. Fixados um parâmetro ϵ, que representa a qualidade da predição (acurácia) do algoritmo, e um parâmetro δ,

[1]lucas_ignacio@id.uff.br.
[2]msdias@id.uff.br.
[3]alanprata@gmail.com .

que representa a confiança na amostra, sua definição nos garante que, se rodarmos o algoritmo em um número suficiente de dados de treino, o que chamamos de complexidade de amostra, com probabilidade maior ou igual a $1-\delta$, o risco da hipótese retornada pelo algoritmo, $L_D(h_S)$, não irá diferir mais do que ϵ do risco da melhor hipótese contida na classe selecionada, isto é

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\left[L_D(h_{\mathcal{S}}) \leq \min_{h'\in\mathcal{H}} L_D(h') + \epsilon\right] \geq 1-\delta.$$

Utilizando a teoria de aprendizado PAC, estudamos a formulação do algoritmo Perceptron desenvolvido por Rosemblatt. Este algoritmo é construído a partir de classe de hipóteses de semiespaços, que em $\mathbb{R}^d$ é dada por

$$HS_d = sign \circ L_d = \{\mathbf{x} \to sign(h_{\mathbf{w},b}(\mathbf{x})) \ : \ h_{\mathbf{w},b} \in L_d\}.$$

O algoritmo de Rosemblatt definido em [1] é um algoritimo iterativo que constrói uma sequência de vetores $(\mathbf{w}^{(n)})_{n\in\mathbb{N}}$ descrita a seguir. Inicialmente, o vetor $\mathbf{w}^{(1)}$ é definido como o vetor nulo. Na iteração t, o algoritmo encontra um exemplo i que é classificado de forma incorreta por $\mathbf{w}^{(t)}$, isto é, um exemplo tal que $sign(\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle) \neq y_i$, e utiliza a seguinte regra de atualização: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i\mathbf{x}_i$.

O objetivo é obter um vetor $\mathbf{w}$ tal que $y_i\langle \mathbf{w}, \mathbf{x}_i\rangle > 0$ para todo $i \in [m]$, e

$$y_i\langle \mathbf{w}^{(t+1)}, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)} + y_i\mathbf{x}_i, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle + \|\mathbf{x}_i\|^2.$$

Assim, sua regra de atualização guia a solução a minimizar o risco empírico a cada iteração, de modo que ele é baseado no princípio de minimização do risco empírico. O teorema a seguir nos garante que, em certo caso, o algoritmo classifica corretamente todos os dados de treino.

Teorema 0.1. *Assuma que $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)$ é linearmente separável, ou seja, que existe um hiperplano capaz de dividir corretamente todas os dados da amostra em suas classes correspondentes. Sejam $B = \min\{\|\boldsymbol{w}\| \ : \ \forall i \in [m],\ y_i\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle \geq 1\}$ e $R = \max_i \|\boldsymbol{x}_i\|$. Então, o algoritmo perceptron para em no máximo $(RB)^2$ iterações, e quando ele para, temos que $\forall i \in [m],\ y_i\langle \boldsymbol{w}^{(t)}, \boldsymbol{x}_i\rangle > 0$.*

Finalmente, aplicamos o algoritmo Perceptron, utilizando a linguagem de programação Python munida com a biblioteca scikit-learn, em dois problemas de classificação reais, a fim de estudarmos sua eficiência.

O primeiro problema se refere ao reconhecimento, através de imagens, de dígitos de 0 a 9 escritos à mão. Para possibilitar o treinamento, cada imagem foi transformada em uma matriz contendo informações acerca da cor dos *pixels* que constituem a imagem. Nesta caso, a pontuação de predição obtida foi de 0.9. Ou seja, o modelo acertou o dígito de 90% dos pontos aos quais ele não teve acesso durante o processo de treinamento.

O segundo problema de classificação compreende identificar se um tumor presente na mama de uma paciente é maligno ou benigno, tendo acesso a trinta informações distintas acerca de biópisia. Para este problema, a pontuação de predição foi de aproximadamente $0,96$.

Referências

[1] Rosenblatt, F. *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, v. 65 6, p. 386–408, 1958

[2] Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*. USA: The MIT Press, 2012.

[3] Shalev-shwartz, S.; Ben-david, s. *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.