"Uma Brevíssima Introdução à Teoria da Aprendizagem Computacional"

Teoria de Aprendizagem Computacional é o ramo de estudo que objetiva aplicar Teoria da Complexidade para problemas de aprendizado. Nesta breve introdução, iremos apresentar, de forma intuitiva, o Modelo de Aprendizado-PAC (*Probably Approximately Correct*) de Leslie Valiant (1984) que é o trabalho seminal da área.

12 de março de 2019; rev. 3 de maio de 2019

Fred Guth

1 O que é Apreensível?

O que é apreensível? Essa é uma questão anterior à Ciência da Computação. No século 18, o filósofo escocês David Hume se perguntou se é possível gerar conhecimento a partir da indução (**O problema da Indução**)[3]. O que é justamente a base do aprendizado supervisionado, a área da Inteligência Artificial mais pesquisada no momento.

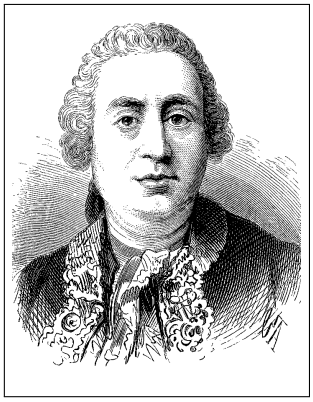

Figura 1: David Hume (1711-1776).

Para Hume, não há justificativa lógica para generalizações baseadas em indução.

> "O pão que comi anteriormente alimentou-me, (...) mas segue-se porventura disso outro pão deva igualmente alimentar-me em outra ocasião? Essa ocasião não parece de nenhum modo necessária" [4].

Para ele nós vemos causalidade onde há apenas a conjunção constante entre dois acontecimentos distintos. Mas se isso é verdade, é possível obter conhecimento através da indução? E se não é possível, o aprendizado indutivo supervisionado não tem justificativa? Pior, o método científico não pode ser logicamente justificado?

A abordagem de Valiant traz uma resposta ao mesmo tempo prática e formal para esse problema. Não é possível dizer que "esse pão que ainda não comi vai me alimentar", sem comê-lo. Entretanto, é possível dizer, com certo nível de confiança que a hipótese "vai me alimentar"está aproximadamente correta. Em outras palavras, embora seja possível que o pão não me alimente (vai que está estragado?), na maioria das vezes ele me alimenta e é possível mensurar a probabilidade desta hipótese estar aproximadamente correta.

Assim como a Teoria da Computação apresenta o conceito de computabilidade, Teoria da Aprendizagem apresenta o conceito de apreensibilidade, o que é e o que não é apreensível.

2 Teoria da Aprendizagem Computacional

Teoria da Complexidade Computacional, um dos alicerces da Ciência da Computação, tem por objetivo examinar e classificar problemas de acordo com suas dificuldades de solução. Ao estudar complexidade de algoritmos, entusiastas de aprendizado de máquina podem naturalmente se perguntar:

> Será que não seria possível analisar qual seria a quantidade de amostras necessárias para um algoritmo aprender uma tarefa?

A resposta é, naturalmente, sim. Em 1984, Leslie Valiant publicou o modelo de aprendizado **Provavelmente Aproximadamente Correto** (PAC, *Probably Approximately Correct*) [7], justamente com o objetivo de incitar pesquisadores que estudavam complexidade de algoritmos a pensar problemas de aprendizado.

Ele introduziu a ideia de problemas de aprendizado que são apreensíveis em tempo polinomial, **PAC apreensíveis**, em analogia com a classe dos problemas P. Pode-se dizer que foi bem sucedido: diversos pesquisadores estenderam ou propuseram novas teorias, o que originou o ramo de estudo chamado de **Teoria de Aprendizado Computacional**, que tem até a sua própria conferência anual, COLT (*Conference on Learning Theory*) [1].

Nesse trabalho, o objetivo é apresentar de forma intuitiva, reduzindo bastante os jargões, o Modelo de Aprendizado-PAC de Leslie Valiant que é o trabalho seminal da área.

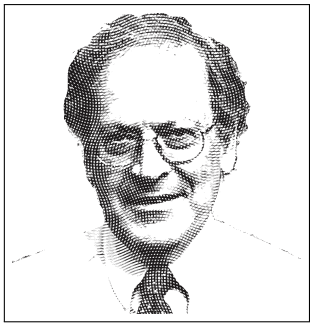

Figura 2: Leslie Valiant, Prêmio Turing 2010.

3 Modelo de Aprendizado-PAC

3.1 Definição informal

$$\underbrace{\text{Provavelmente}}_{\text{confiança}>(1-\delta)}\ \underbrace{\text{Aproximadamente}}_{\text{tolerância}\leq\epsilon}\ \underbrace{\text{Correto}}_{\text{Erro Absoluto}=0}$$

No **modelo PAC**, uma tarefa é apreensível se existe um algoritmo capaz de aprendê-la de forma confiável em um número razoável (polinomial) de passos e amostras de treinamento.

3.2 Definições e Notações

Genericamente, podemos pensar em um algoritmo como uma função que transforma uma instância do espaço de entrada, $\mathcal{X}$, em uma instância do espaço de soluções da tarefa que queremos realizar, $\mathcal{Y}$ (figura 3). Em geral, essa função é um conjunto de regras programadas.

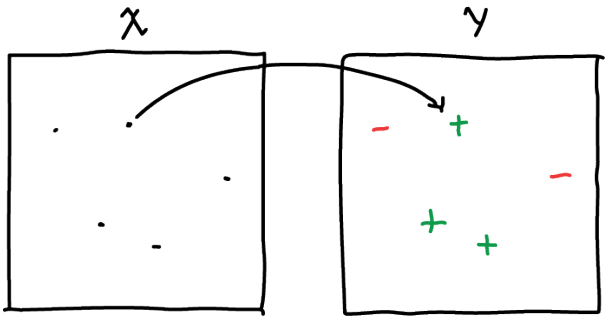

Figura 3: Um algoritmo genérico: $\mathcal{X} \rightarrow \mathcal{Y}$.

No contexto de aprendizado de máquina, não sabemos expressar essa função alvo, portanto teremos que inferi-la, aprendê-la. Chamaremos essa função ideal de **conceito** (figura 4). É o conceito que queremos aprender. No modelo PAC, um conceito é uma função $c : \mathcal{X} \rightarrow \{+, -\}$, ou seja, uma função que mapeia o espaço das instâncias em um valor *booleano*.

A função realmente obtida, a regra, a heurística, chamamos de **hipótese**(figura 5).

O aprendiz (figura 6), portanto, escolhe uma hipótese h restrita às generalizações permitidas e preferidas pelo espaço de hipóteses $\mathcal{H}$ (figura 7). O objetivo do aprendiz $\mathcal{A}$ é generalizar bem, ou seja, escolher uma hipótese com menor erro na distribuição desconhecida $\mathcal{D} = P(\mathcal{X})$, menor **erro absoluto** ($erro_{\mathcal{D}}$). Entretanto, $\mathcal{A}$ só tem conhecimento do **erro de treinamento**, $erro_S(h)$ (figura 8). Podemos formular o erro absoluto e o erro de treinamento da seguinte forma [5]:

$$erro_{\mathcal{D}}(h) \equiv P_{x\sim\mathcal{D}}[c(x) \neq h(x)] \tag{1}$$

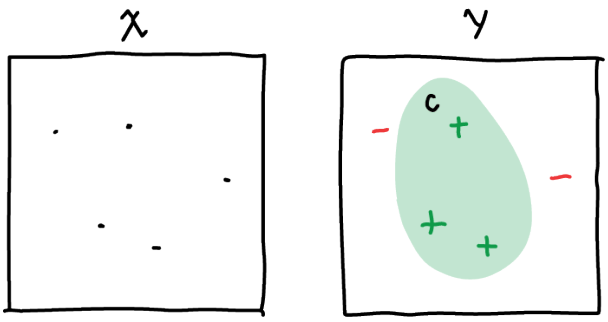

Figura 4: Um conceito c.

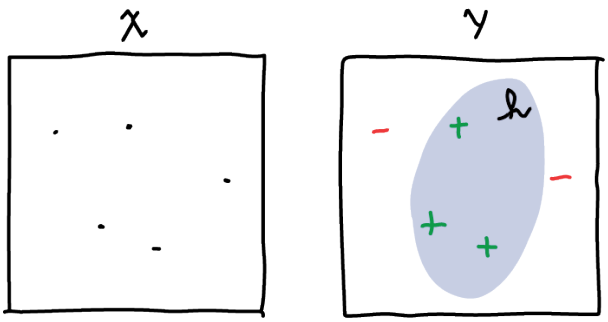

Figura 5: Uma hipótese $h \in \mathcal{H}$.

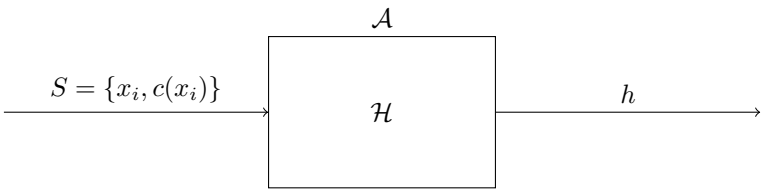

Figura 6: Um aprendiz genérico $\mathcal{A}$.

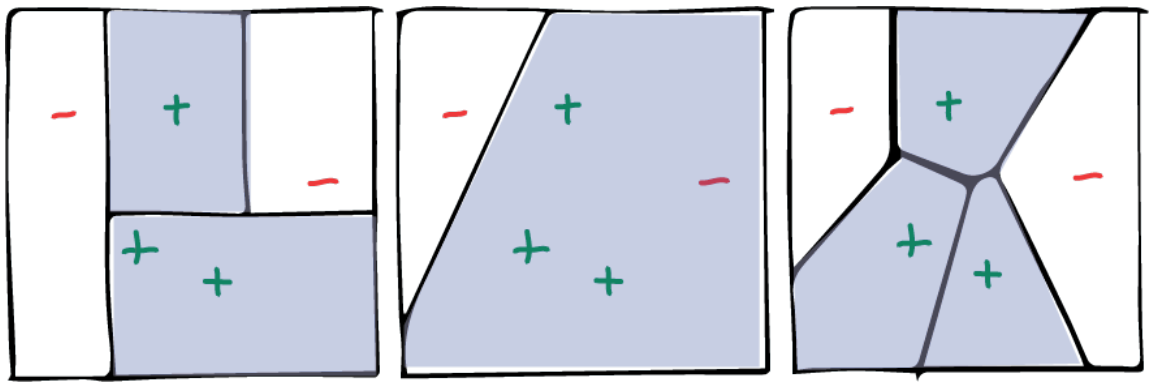

Figura 7: hipóteses pertencentes a diferentes espaços de hipóteses $\mathcal{H}$.

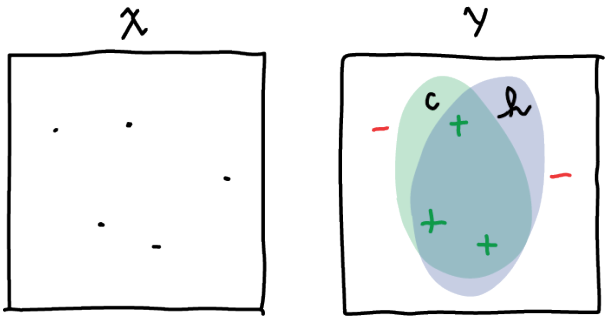

Figura 8: O erro, $c(x) \neq h(x)$

.

$$erro_S(h) \equiv P_{x \sim S}[c(x) \neq h(x)] \tag{2}$$

3.3 Modelo PAC: Definição formal

$\mathcal{C}$ é **PAC-apreensível** por $\mathcal{A}$ se e somente se, com probabilidade $1-\delta$, $\mathcal{A}$ gera uma hipótese $h \in \mathcal{H}$ com $erro_{\mathcal{D}}(h) \leq \epsilon$ com número de amostras polinomial em função de $1/\delta$, $1/\epsilon$, $n = |x|$ e $|c|$, para $0 < \epsilon \leq \frac{1}{2}$ e $0 < \delta \leq \frac{1}{2}$.

$\mathcal{C}$ é **eficientemente PAC-apreensível** por $\mathcal{A}$ se e somente se $\mathcal{A}$ é **PAC-apreensível** e gera hipótese em tempo polinomial em função de $1/\delta$, $1/\epsilon$, $|x|$ e $|c|$.

3.4 Teorema de Haussler (1988): Limites para $\mathcal{H}$ finito.

Para espaços de hipóteses finito e que contém o conceito que se quer aprender, o teorema de Haussler[2] dá um limite inferior ao erro absoluto esperado dado um número de amostras m ou o limite mínimo de amostras necessárias m para um aprendiz consistente aprender a classe de conceitos $\mathcal{C}$ com níveis aceitáveis de precisão e confiança.

Teorema: Seja $\mathcal{H}$ finito. Seja $\mathcal{A}$ um aprendiz que para qualquer conceito alvo c e distribuição desconhecida $\mathcal{D}$ retorna uma hipótese consistente $h : erro_S(h) = 0$. Seja $|S| = m, m \geq 1$, então
$P[\exists h \in \mathcal{H} : error_{\mathcal{D}}(h) > \epsilon] \leq |H|e^{-\epsilon m}$

Demonstração. Sejam $h_i (i = 1, ..., k)$ todas as hipóteses em $\mathcal{H}$ tais que $erro_{\mathcal{D}}(h_i) > \epsilon$, temos:

$$P_{x_j \sim \mathcal{S}}[(c(x_j) \neq h_i(x_j)) = 0] \leq (1-\epsilon) \tag{3}$$
$$P[\forall h \in |\mathcal{H}| : (erro_S(h) = 0 \wedge erro_{\mathcal{D}}(h) > \epsilon)] \leq (1-\epsilon)^m \tag{4}$$
$$P[\exists h \in H : erro_{\mathcal{D}}(h) > \epsilon] \leq k(1-\epsilon)^m \tag{5}$$
$$P[\exists h \in H : erro_{\mathcal{D}}(h) > \epsilon] \leq |\mathcal{H}|(1-\epsilon)^m \tag{6}$$
$$(1-x) \leq e^{-x}, 0 \leq x \leq 1 \implies P[\exists h \in \mathcal{H} : error_{\mathcal{D}}(h) > \epsilon] \leq |H|e^{-\epsilon m} \tag{7}$$

□

A probabilidade de uma hipótese "ruim"($erro_{\mathcal{D}}(h_i) > \epsilon$) prever corretamente uma determinada amostra é $\leq (1-\epsilon)$ (eq. 3). Portanto, a probabilidade de uma hipótese "ruim"prever corretamente todas as amostras de treinamento é $\leq (1-\epsilon)^m$ (eq. 4). Há hipóteses "boas"e "ruins", imaginando que há k hipóteses "ruins", a probabilidade de alguma destas k hipóteses "ruins"prever corretamente todo o conjunto de treinamento é $\leq k(1-\epsilon)^m$ (eq. 5). Como essas hipóteses "ruins"pertencem a $\mathcal{H}$, k é necessariamente menor que $|\mathcal{H}|$, portanto chegamos ao limite do erro absoluto de h dado uma precisão ϵ e um número de amostras m (eq. 6).

Como dito anteriormente, uma outra forma de ver o Teorema de Haussler é que para uma determinada tolerância ao erro ϵ e confiança $(1-\delta)$, o teorema dá o limite inferior no número de amostras de treinamento, $m \geq \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta})$, para se aprender a classe de conceitos $\mathcal{C}$.

$\|H\|e^{-\epsilon m} = \delta \implies$ aprendiz $\mathcal{A}$ aprende $\mathcal{C}$ em $m = \frac{1}{\epsilon}(\ln\|\mathcal{H}\| + \ln\frac{1}{\delta})$ amostras de treinamento.

Demonstração. Dado que o Teorema de Haussler demonstra que $|\mathcal{H}|e^{-\epsilon m}$ é um limite inferior para a probabilidade do **erro absoluto** ultrapassar a tolerância ao erro (ϵ) e que o modelo de aprendizado-PAC tem como limite inferior desta mesma probabilidade o valor δ, podemos supor $|\mathcal{H}|e^{-\epsilon m} = \delta$, consequentemente:

$$|\mathcal{H}|e^{-\epsilon m} = \delta \implies e^{-\epsilon m} = \frac{\delta}{|\mathcal{H}|} \tag{8}$$

$$-\epsilon m = (\ln\delta - \ln|\mathcal{H}|) \tag{9}$$

$$\epsilon m = (\ln|\mathcal{H}| - \ln\delta) \tag{10}$$

$$m = \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta}) \tag{11}$$

□

Interessante notar que esse limite inferior no número de amostras é idependente de $\mathcal{C}$ ou $\mathcal{D}$ e logarítmico em relação ao tamanho do espaço $\mathcal{H}$ [2].

4 Exemplos de aplicação do modelo de aprendizado-PAC

4.1 A classe universal de conceitos é PAC-apreensível?

Seja o universo das instâncias $\mathcal{X} = \{0,1\}^n$, o espaço de vetores booleanos de tamanho n. A classe de conceitos $\mathcal{U}_n$ é formada por todos os subconjuntos de $\mathcal{X}$, portanto, pode ser considerada como uma classe universal de conceitos já que contém todas as possíveis classificações para o espaço das instâncias dado.

$$|\mathcal{U}_n| = 2^{|\mathcal{X}|} = 2^{(2^m)} \tag{12}$$

$$|\mathcal{H}| \geq |\mathcal{U}_n| \tag{13}$$

$$|\mathcal{H}| \geq 2^{(2^m)} \tag{14}$$

Com a eq. 11, temos:

$$m \geq \frac{1}{\epsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta}) \tag{15}$$

$$m \geq \frac{1}{\epsilon}((2^m)\ln 2 + \ln\frac{1}{\delta}) \tag{16}$$

$$\therefore m \in O(2^m, \frac{1}{\epsilon}, ln\frac{1}{\delta}) \tag{17}$$

Logo, $\mathcal{U}_n$ **não é PAC-apreensível**.

There is no free-lunch.

4.2 A classe de conjunções de n literais é PAC-apreensível?

Seja o universo das instâncias $\mathcal{X} = \{x_1, \ldots, x_n\}$, n literais, ou seja, $x_i \vee \bar{x_i} = 1$.

A classe de conceitos representáveis por uma conjunção de até n literais, C_n é PAC-apreensível?

$$|C_n| = 3^n \tag{18}$$

uma vez que cada literal pode ser verdadeiro, falso ou não estar em C_n. Com a eq. 11, temos:

$$m \geq \frac{1}{\epsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta}) \tag{19}$$

$$m \geq \frac{1}{\epsilon}(n \ln 3 + \ln \frac{1}{\delta}) \tag{20}$$

$$\therefore m \in O(n, \frac{1}{\epsilon}, ln\frac{1}{\delta}) \tag{21}$$

Logo, $\mathcal{U}_n$ **é PAC-apreensível**. Além disso, é fácil provar que C_n **eficientemente PAC-apreensível**, já que o seguinte algoritmo polinomial em função do número de literais pode ser usado:

Seja $B_j = b_1, \ldots, b_n$ uma amostra para aprender uma expressão $c \in C_n$.

Para todos os 3^n possíveis valores de B_j:

$(B_j = 1 \wedge b_i = 0) \implies x_i \notin C_n$;

$(B_j = 1 \wedge b_i = 1) \implies \bar{x_i} \notin C_n$;

$(B_j = 0) \implies$ não há nada a aprender com esta amostra.

Quando todos os exemplos positivos de B_j tiverem sido processados, retorne a conjunção de $x_i \in C_n$.

Como C_n é PAC-apreensível e há um algoritmo polinomial em n e $|C_n|$ para obter h, C_n **é eficientemente PAC-apreensível**.

4.3 A classe de fórmulas k-CNF é PAC-apreensível?

Uma fórmula k-CNF é uma conjunção em que cada termo tem até k literais, $\bigwedge_1^n(\bigvee_1^k)$. Portanto, seja $\mathcal{C}_{k-CNF}$ a classe de todos os conceitos que podem ser representados por uma fórmula k-CNF, $\mathcal{C}_{k-CNF}$ é PAC-apreensível?

$$|\mathcal{C}_{k-CNF}| = (3^k)^n = 3^{k\dot{n}} \tag{22}$$

Com a eq. 11, temos:

$$m \geq \frac{1}{\epsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta}) \tag{23}$$

$$m \geq \frac{1}{\epsilon}(\not{k}n \ln 3 + \ln \frac{1}{\delta}) \tag{24}$$

$$\therefore m \in O(n, \frac{1}{\epsilon}, ln\frac{1}{\delta}) \tag{25}$$

Logo, $\mathcal{C}_{k-CNF}$ **é PAC-apreensível**.

Como C_{k-CNF} é reduzível a C_n

$$C_{k-DNF} \leq_m C_n \tag{26}$$

[6, pg. 18] , $\mathcal{C}_{k-CNF}$ **é eficientemente PAC-apreensível**.

4.4 A classe de fórmulas k-DNF é PAC-apreensível?

Uma fórmula k-DNF é uma conjunção em que cada termo tem até k literais, $\bigvee_1^n(\bigwedge_1^k)$. Portanto, seja $\mathcal{C}_{k-CNF}$ a classe de todos os conceitos que podem ser representados por uma fórmula k-CNF, $\mathcal{C}_{k-CNF}$ é PAC-apreensível?

É fácil ver que $|\mathcal{C}_{k-DNF}| = 3^{k\dot{n}}$ pelo mesmo raciocínio da eq. 22.

$$|\mathcal{C}_{k-DNF}| = (3^k)^n = 3^{k\dot{n}} \tag{27}$$

Logo, pelos mesmos motivos de C_{k-DNF}, $\mathcal{C}_{k-CNF}$ **é PAC-apreensível**.

Entretanto, $\mathcal{C}_{k-DNF}$ **não é eficientemente PAC-apreensível**, uma vez que é possível reduzir o problema 3-coloring de grafos para o 3-DNF:

$$\text{3-coloring} \leq C_{k-DNF} \tag{28}$$

$$\therefore C_{k-DNF} \in NP \tag{29}$$

[6, pg. 17-18]

5 Conclusão

Teoria de Aprendizagem Computacional traz um arcabouço teórico que nos permite estimar o número de amostras de treinamento necessárias para um algoritmo convergir, com alta probabilidade, para uma boa generalização.

Em particular no caso de $\mathcal{H}$ finito, para convergir com determinado nível de confiança e tolerância a erro, o limite inferior no tamanho do conjunto de treinamento é independente da classe de conceitos alvo ou da distribuição das instâncias de entrada e depende apenas logaritmicamente do tamanho do espaço de hipóteses.

Na prática, no entanto, esses limites são muito "frouxos"[5] e dão a falsa impressão que modelos mais simples (menor espaço de hipóteses) são sempre melhores, o que não corresponde com a realidade em que modelos complexos e grandes como redes neurais profundas apresentam melhores generalizações para um grande número de aplicações.

Um outro ponto interessante apresentado é o fato da classe de conceitos k-CNF ser eficientemente PAC-apreensível, enquanto k-DNF não é. Apesar dessas formas serem conversíveis uma a outra, a conversão é NP. Esse caso demonstra a importância da representação dos conceitos ao evidenciar que conceitos equivalentes apresentam aprendizados com complexidades diferentes.

Referências

[1] ACL Association for Computational Learning. learningtheory.org, 2019.

[2] David Haussler. Quantifying inductive bias: AI learning algorithms and valiants learning framework. *Artificial Intelligence*, 36(2):177–221, sep 1988.

[3] D. Hume. *Tratado da natureza humana - 2a Edição*. Ed. UNESP, 2009.

[4] David Hume. Investigações sobre o entendimento humano e sobre os princípios da moral. *São Paulo: Editora UNESP*, 2004.

[5] Tom M. Mitchel. Machine learning 10-701 - lecture 12, February 2011.

[6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[7] L. G. Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing - STOC '84*. ACM Press, 1984.