Lucas França Ferreira Ignacio

Aprendizado de máquina: da teoria à aplicação

Volta Redonda, RJ

2021

Lucas França Ferreira Ignacio

Aprendizado de máquina: da teoria à aplicação

Trabalho de Conclusão de Curso submetido ao Curso de Matemática com ênfase em Matemática Computacional da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Matemática.

Universidade Federal Fluminense

Instituto de Ciências Exatas

Curso de Matemática

Orientador: Marina Sequeiros Dias de Freitas

Coorientador: Alan Prata de Paula

Volta Redonda, RJ

2021

Ficha catalográfica automática - SDC/BAVR
Gerada com informações fornecidas pelo autor

I24a Ignacio, Lucas França Ferreira
Aprendizado de máquina: da teoria à aplicação / Lucas França Ferreira Ignacio ; Marina Sequeiros Dias de Freitas, orientadora ; Alan Prata de Paula, coorientador. Volta Redonda, 2021.
80 f. : il.

Trabalho de Conclusão de Curso (Graduação em Matemática)- Universidade Federal Fluminense, Instituto de Ciências Exatas, Volta Redonda, 2021.

1. Aprendizado de máquina. 2. Aprendizado PAC. 3. Dimensão VC. 4. Produção intelectual. I. Freitas, Marina Sequeiros Dias de, orientadora. II. Paula, Alan Prata de, coorientador. III. Universidade Federal Fluminense. Instituto de Ciências Exatas. IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Lucas França Ferreira Ignacio

Aprendizado de máquina: da teoria à aplicação

Trabalho de Conclusão de Curso submetido ao Curso de Matemática com ênfase em Matemática Computacional da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Matemática.

Trabalho aprovado. Volta Redonda, RJ, 07 de maio de 2021:

Profa. Dra. Marina Sequeiros Dias de Freitas – UFF
Orientador

Prof. Dr. Alan Prata de Paula – UFF
Coorientador

Prof. Dr. Leandro Gines Egea – UFF

Profa. Dra. Jessica Quintanilha Kubrusly – UFF

Volta Redonda, RJ
2021

Agradecimentos

Agradeço aos meus familiares e a todos os professores do Instituto de Ciências Exatas da Universidade Federal Fluminense pelo apoio, sem o qual este trabalho não seria possível.

Em especial, agradeço aos meus pais e orientadores, que acompanharam de perto meu desenvolvimento, sempre me incentivando nesta caminhada.

Resumo

Neste trabalho, é apresentada a teoria de aprendizado PAC, uma definição matemática embasada na teoria de Probabilidade, utilizada para justificar, de maneira formal, intuições que permeiam o campo de aprendizado de máquina. São abordadas tanto a definição desta noção de aprendizado e suas generalizações como os conceitos e resultados que a fundamentam, sendo a ideia de dimensão VC um exemplo.

Ademais, baseado no princípio de minimização do risco empírico (ERM), são discutidas as formulações de dois algoritmos de aprendizado, o algoritmo Perceptron e a Regressão Linear. Por fim, estes dois algoritmos são utilizados para resolver problemas clássicos presentes na literatura.

Palavras-chave: 1.Aprendizado de máquina 2.Aprendizado PAC 3.Dimensão VC

Abstract

In this work, the PAC learning theory is presented, a mathematical definition based on Probability theory, used to justify, in a formal way, intuitions that permeate the field of machine learning. Both the definition of this notion of learning and its generalizations are addressed, as well as the concepts and results that underlie it, the idea of VC dimension being one example.

Furthermore, based on the principle of empirical risk minimization (ERM), the formulations of two learning algorithms, the Perceptron algorithm and the Linear Regression, are discussed. Finally, these two algorithms are used to solve classical problems present in the literature.

Keywords: 1.Machine Learning 2.PAC Learning 3.VC dimension

Sumário

1 Introdução

O termo Inteligência Artificial foi cunhado em uma conferência na Dartmouth University no ano de 1956. Desde então, este campo abrangente do conhecimento vem passando por um rápido desenvolvimento em suas técnicas e relevância, estando presente em diversos aspectos do nosso dia a dia. Conforme discutido em [1] e [2], as aplicações desta área do conhecimento variam desde respostas automáticas e sumarização do conteúdo de e-mails à reconhecimento de faces e fala, bem como análise do mercado financeiro e diagnóstico de câncer.

Neste trabalho, faremos um recorte para o campo do aprendizado de máquina, cuja principal característica é sua dependência de um conjunto de observações de um problema que se busca resolver (ver [3]). Baseado nesse conjunto de dados, os algoritmos de aprendizado são submetidos a um processo de treinamento, cujo objetivo é que tais algoritmos possam realizar tomada de decisão sem a necessidade de interferência humana.

Um problema de aprendizado pode ser classificado de diferentes maneiras, conforme será exposto no Capítulo 2, cada um contando com suas abordagens e técnicas. A primeira distinção entre os diversos tipos de aprendizado pode ser feita em relação aos dados que temos disponibilidade. Se os dados de treino contêm tanto os valores de entrada, como os valores que queremos estimar, dizemos que o aprendizado é supervisionado. Caso contrário, temos um problema de aprendizado não supervisionado. Ademais, se estamos interessados em predizer um valor, temos um problema de regressão. Quando estamos interessados em separar os dados em diferentes classes, temos um problema de classificação. O enfoque deste estudo será dado à classe de problemas supervisionados. É importante ressaltar que estudo de aprendizado de máquina envolve diferentes áreas do conhecimento tais como Matemática, Estatística, Computação e Otimização, sendo assim um campo multidisciplinar.

Ao longo dos próximos capítulos será mostrado que um algoritmo de aprendizado, munido de uma classe de funções, nos dá como saída uma função que minimiza a diferença entre seu erro de predição na amostra (dados de treino) e seu erro de predição na população (dados aos quais ele nunca teve acesso), a isto chamaremos de capacidade de generalização. Tal resultado será obtido através de resultados estatísticos, de forma que esta noção de aprendizado é chamada de aprendizado estatístico.

O principal objetivo deste trabalho é, portanto, expor o ferramental matemático que nos auxilie a ter uma clara definição do que consiste essa noção de aprendizado, e sob quais circunstâncias somos capazes de garantir que de fato ela é efetiva. Além disto, também serão discutidos alguns algoritmos de aprendizado, os quais serão implementados

em problemas clássicos da literatura.

No Capítulo 2 são expostas algumas noções gerais sobre a teoria de aprendizado de máquina, enquanto que o Capítulo 3 introduz matematicamente duas definições de aprendizado e um dos paradigmas sobre o qual os algoritmos de aprendizado podem ser baseados. O Capítulo 4 discute alguns aspectos relacionados a escolha de um algoritmo para a resolução de um problema, e o que deve ser considerado para embasar esta escolha a fim de garantir um resultado satisfatório. No Capítulo 5 é definido o conceito de dimensão VC, que é utilizado para verificar se classes de funções infinitas são aprendíveis segundo o paradigma PAC. Finalmente, no Capítulo 6 são apresentados dois algoritmos de aprendizado baseados em funções lineares, e no Capítulo 7 estes algoritmos são utilizados para resolver problemas concretos.

Ademais, no Apêndice A são relembradas algumas definições e resultados da teoria de probabilidade, que serão utilizados ao longo deste trabalho, e no Apêndice B estão as demonstrações de alguns resultados que foram retiradas do texto para garantir uma maior coesão de ideias e melhor legibilidade.

Notação

Antes de discutirmos os modelos de aprendizado que serão apresentados, será necessário introduzirmos algumas notações que serão utilizadas ao longo do texto.

As letras maiúsculas X, Y e Z são reservadas para variáveis aleatórias e as letras minúsculas x, y e z são elementos de um conjunto, que, dependendo do contexto, são realizações das variáveis X, Y e Z. Desta forma, $\mathcal{S} = \Big((X_1, Y_1), ..., (X_m, Y_m)\Big)$ representa um vetor aleatório de dimensão m e $S = \{(x_1, y_1), ..., (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ é um conjunto contendo m pares ordenados de $\mathcal{X} \times \mathcal{Y}$, que, dependendo do contexto, são m realizações (independentes) do vetor aleatório (X, Y).

Na prática, os conjuntos $\mathcal{X}$ e $\mathcal{Y}$ representam espaços $\mathbb{R}^m$ e $\mathbb{R}^n$, uma vez que, ao lidarmos com aplicações concretas de aprendizado de máquina, os valores que o problema assume são números reais ou vetores de números reais.

Ademais, dada uma função $g : \mathcal{X} \to \mathcal{Y}$ e uma variável aleatória X, $g(X) = g \circ X$. Ou seja, um ponto $x \in \mathcal{X}$ é obtido de maneira aleatória e então aplicado em uma função que tem $\mathcal{X}$ como domínio. Desta maneira, $g(X)$ também é uma variável aleatória sobre o mesmo espaço de probabilidade o qual X está definida.

A probabilidade e a esperança de um evento ou variável aleatória serão denotadas, respectivamente, por $\mathbb{P}$ e $\mathbb{E}$. Suas definições se encontram no Apêndice *A*.

Dizemos que a variável aleatória X tem distribuição D se $\underset{X \sim D}{\mathbb{P}}[X \in A] = D(\{x \ : \ x \in A\})$, isto é, a probabilidade do resultado de uma variável aleatória X estar contida em um conjunto A é dada pela medida de probabilidade D.

Em alguns momentos, por uma questão de simplicidade, será utilizada a abreviatura iid, para indicar que um conjunto é independente e identicamente distribuído.

A notação $[n]$ representa o conjunto dos números naturais menores ou iguais a n, ou seja, $[n] = \{1, 2, ..., n\}$.

Dados dois vetores $u = (u_1, ..., u_m), \ v = (v_1, ..., v_m) \in \mathbb{R}^m$, denotaremos seu produto interno, dado pela função $\langle \cdot, \cdot \rangle : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, por

$$\langle u, v \rangle = \sum_{i=1}^{m} u_i \cdot v_i.$$

Além disto, definimos a função sinal $sign : \mathbb{R} \to \mathbb{R}$ como

$$sign(x) = \begin{cases} -1, & \text{se } x < 0; \\ 1, & \text{se } x > 0. \end{cases}$$

Finalmente, dado um conjunto A e um domínio $\mathcal{X}$, definimos a função indicadora de A, $\mathbb{1}_A : \mathcal{X} \to \{0, 1\}$, como

$$\mathbb{1}_A(x) = \begin{cases} 0, & \text{se } x \notin A; \\ 1, & \text{se } x \in A. \end{cases}$$

2 Noções gerais

Neste capítulo são apresentados, de forma breve, os diferentes paradigmas e tipos de problemas de aprendizado.

2.1 O modelo e a estrutura de um problema de aprendizado

A fim de entender no que consiste um problema de aprendizado supervisionado, consideremos um exemplo.

Imagine que um banco deseja desenvolver um algoritmo capaz de aprovar ou rejeitar o pedido de crédito de um cliente. Neste caso, podemos interpretar a informação de cada cliente como sendo um vetor $x \in \mathbb{R}^d$ onde cada coordenada deste vetor representa uma informação individual.

Assim, o problema pode ser visto como a busca por uma função $f : \mathbb{R}^d \to \{-1, 1\}$ que tem como entrada os dados do cliente, e como saída -1, significando que o crédito deve ser aprovado, ou 1 caso o crédito deva ser rejeitado. Ressaltamos que estamos interessados apenas em funções que cometem poucos erros, ou seja, que não aprovam clientes que eventualmente não paguem suas dívidas, e que não rejeitam clientes que podem vir a trazer lucro.

Como exemplo, podemos tomar a composição $sign \circ h_{w,b}$, sendo $h_{w,b}(x) = \langle w, x\rangle + b$ onde w é um vetor de $\mathbb{R}^d$ e $b \in \mathbb{R}$. A função $h_{w,b}$ pode ser interpretada como o cálculo do *score* bancário do cliente, isto é, uma pontuação que indica as chances de determinado perfil pagar as contas em dia nos próximos 12 meses.

No entanto, note que, para cada par de parâmetros w e b, teremos uma função de decisão diferente. Sendo assim, estamos lidando com uma classe de funções $\{f \mid f : \mathbb{R}^d \to \{-1, 1\}\}$. Como estamos interessados em obter a função que comete menos erros, nosso próximo passo é, portanto, submeter o algoritmo a um processo de treinamento, a fim de obter os parâmetros w e b ótimos, ou seja, que minimizam os erros baseado numa amostra de treino, a fim de que, ao aplicarmos o algoritmo já treinado, ele possa ser capaz de tomar decisões corretas sem a necessidade de interferência humana.

2.2 Tipos de aprendizado

O campo de aprendizado de máquina possui diversos ramos de estudo, cada um com suas metodologias e aplicações. Uma das maneiras de fazer uma distinção entre estas

metodologias diz respeito ao tipo de dados disponíveis e aos objetivo a serem alcançados com o aprendizado.

2.2.1 Aprendizado supervisionado

Dizemos que o aprendizado é realizado de maneira supervisionada quando temos acesso a uma amostra do problema com informações sobre determinado objeto de estudo, os chamados dados de entrada, e o resultado obtido com essas informações, chamado dado de saída. Em outras palavras, há a presença de uma variável resposta para guiar o aprendizado, ou seja , o conjunto de treino é da forma $S = \{(x_1, y_1), ..., (x_m, y_m)\}$, onde para cada ponto de entrada $x \in \mathcal{X}$, temos o valor de interesse $y \in \mathcal{Y}$.

Nestes casos, no processo de treinamento, o algoritmo recebe os valores de entrada, e caso ele não apresente a saída correta, podemos corrigi-lo para que não cometa mais este erro. Assim, ao longo deste processo, esperamos que o algoritmo cometa cada vez menos erros sobre a amostra de treino disponível, e que isto se reflita em pontos aos quais ele nunca teve acesso.

Retornando ao exemplo da análise de crédito, poderíamos utilizar as informações de clientes antigos, o que denotaremos por S, dividindo-as em dois conjuntos: dados de treino e dados de teste. O algoritmo seria então executado utilizando o conjunto de dados de treino com a finalidade de obter os parâmetros do modelo estudado. Após esta etapa, para testar a capacidade de generalização, ou seja, se ele é realmente eficaz em fazer a tomada de decisão, testamos sua acurácia em relação aos dados de teste. É importante ressaltar que, durante o treinamento, o algoritmo não deve ter acesso aos dados de teste, uma vez que isto iria comprometer nossa avaliação sobre o seu desempenho.

Portanto, ao realizar aprendizado de maneira supervisionada, utilizamos um conjunto de dados de treino $S \in (\mathcal{X} \times \mathcal{Y})^m$ a fim de treinar um algoritmo para ser capaz de entender a relação entre os conjuntos $\mathcal{X}$ e $\mathcal{Y}$, de modo que, ao aplicarmos o algoritmo já treinado em um ponto $x \in \mathcal{X}$ ao qual ele nunca teve acesso, ele nos retorne o ponto $y \in \mathcal{Y}$ esperado.

Outro exemplo é o conjunto de dados Iris, um problema clássico da literatura. A partir de informações sobre uma amostra de vegetação, tem-se como objetivo classificar corretamente esta amostra em três espécies distintas. Neste problema, utilizamos aprendizado supervisionado, uma vez que além de termos acesso a informações de cada amostra, também temos a informação sobre a espécie a qual ela realmente pertence, e o objetivo do aprendizado é classificar os tipos de vegetação.

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
7.7	3.8	6.7	2.2	virginica
7.7	2.6	6.9	2.3	virginica

Figura 1 – Conjunto de dados iris como um exemplo de aprendizado supervisionado.

2.2.2 Aprendizado não supervisionado

Por outro lado, o aprendizado é dito não supervisionado quando não temos acesso a um conjunto de amostras contendo a relação entre pontos de entrada e de saída, e o objetivo do aprendizado consiste em extrair padrões dos dados disponíveis.

Nesse caso, existe uma estrutura do espaço de entrada tal que certos padrões ocorrem mais frequentemente do que outros, e queremos entender esse padrões. Em Estatística, isso é chamado de estimação da densidade. Um dos métodos de estimação de densidade mais conhecido é a análise de conglomerados, em inglês *clustering*, cujo objetivo é encontrar agrupamentos das entradas. À vista disso, busca-se observar algumas similaridades entre os objetos e incluí-los em grupos apropriados.

No exemplo da análise de crédito, o banco tem informações de seus clientes, tais como informações demográficas e transações com o banco e daí pode usar esses dados para ver a distribuição do perfil de seus clientes e analisar os tipos de clientes que frequentemente fazem parte do banco. Desse modo, a análise de conglomerados aloca os clientes com características similares dentro de um mesmo grupo, evidenciando agrupamentos naturais de seu público. Esse tipo de informação pode ser útil para o banco definir serviços e produtos específicos para grupos diferentes. Além disso, nessa análise também pode ser identificado alguns pontos discrepantes, isto é, clientes que são diferentes dos outros consumidores e isto pode representar um nicho de mercado que pode ser explorado pelo banco, através de estratégias para atrair clientes de outros perfis.

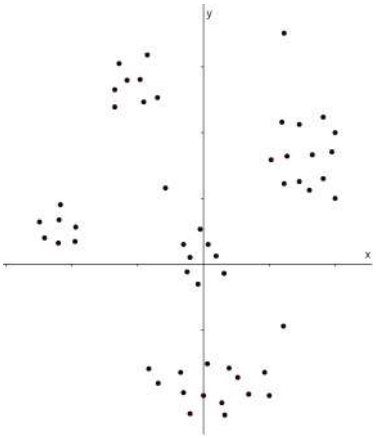

Figura 2 – Aprendizado não supervisionado.

A Figura 2 ilustra esse caso. Não temos acesso a nenhuma informação adicional sobre os pontos individuais, mas podemos perceber que eles parecem estar distribuídos em grupos distintos. Também podemos analisar a distância de um ponto aos demais pontos próximos a ele, a fim de tentar caracterizá-lo.

2.2.3 Aprendizado por reforço

O aprendizado por reforço é um paradigma de aprendizado baseado no processo de decisão de Markov. Esta técnica consiste em ensinar um agente a interagir com um meio, através de um conjunto finito de ações, a fim de que ele atinja um objetivo definido. Neste contexto, o processo de treinamento constitui-se em aplicar um valor de punição para cada ação realizada pelo agente, de modo que o agente busca encontrar as ações que minimizam a punição recebida.

Uma das aplicações mais usuais desta metodologia é ensinar algoritmos a jogarem jogos que possuam regras e objetivos bem definidos. Por exemplo, tanto em jogos de vídeo game como no xadrez, os jogadores, se valendo de um número finito de ações possíveis, visam obter a vitória, evitando, para isto, tomar decisões desfavoráveis a seus objetivos.

Embora esta técnica seja relativamente mais recente do que as discutidas anteriormente, atualmente, computadores munidos com algoritmos treinados por reforço já venceram campeões mundiais em diversos jogos complexos, como o Go, e a cada dia vem se tornando mais próximos de conquistarem desempenho sobre-humano em outras áreas.

2.3 Problemas de aprendizado supervisionado

Os dados de um problema de aprendizado podem ser caracterizados como quantitativos ou qualitativos. Variáveis quantitativas são aquelas que assumem valores numéricos, como idade e altura, enquanto que variáveis qualitativas (ou categóricas) assumem seu

valor em uma classe dentre um conjunto de classes disponível, como por exemplo o sexo de uma pessoa e a marca de um produto.

Assim, um problema de aprendizado supervisionado pode ser dividido em categorias, dependendo da natureza daquilo que se está tentando predizer. Em outras palavras, dependendo se os valores do contradomínio $\mathcal{Y}$ são quantitativos ou qualitativos.

Embora esta distinção seja útil para explicar o campo de aprendizado de máquina e suas diferentes abordagens, ressaltamos que tal distinção não é sempre tão nítida. A regressão linear de mínimos quadrados é usada com uma resposta quantitativa, enquanto a regressão logística é tipicamente usada com uma resposta qualitativa (duas classes ou binária). Desse modo, é frequentemente usada como método de classificação. Mas, uma vez que estima as probabilidades da classe, também pode ser considerada um método de regressão. Alguns métodos estatísticos, como K-vizinhos mais próximos e *boosting* podem ser usados no caso de respostas quantitativas ou qualitativas [4].

2.3.1 Problemas de classificação

Em geral, dizemos que um problema é de classificação se a variável de resposta é qualitativa. 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.

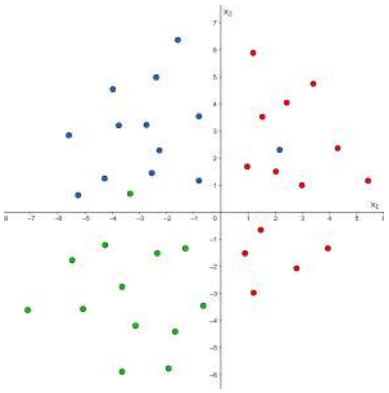

Figura 3 – Problema de classificação supervisionado.

A Figura 3 retrata este tipo problema. Retornando ao exemplo do conjunto de dados Iris, cada cor representa uma classe diferente, ou seja, uma espécie diferente de vegetação.

Outra aplicação que podemos citar é o de reconhecimento de imagens. Por exemplo, fazer a distinção entre cachorros e gatos, baseado em um conjunto de fotos.

2.3.2 Problemas de regressão

Por outro lado, se estamos interessados em predizer um valor quantitativo, temos um problema de regressão. Nesse contexto, para um vetor em $\mathcal{X}$, queremos estimar seu valor de saída $y \in \mathcal{Y}$. Sendo assim, podemos interpretar esta classe de problemas como a tentativa de estimar uma função contínua desconhecida. Aplicações incluem a estimação do valor de ativos em bolsas de valores.

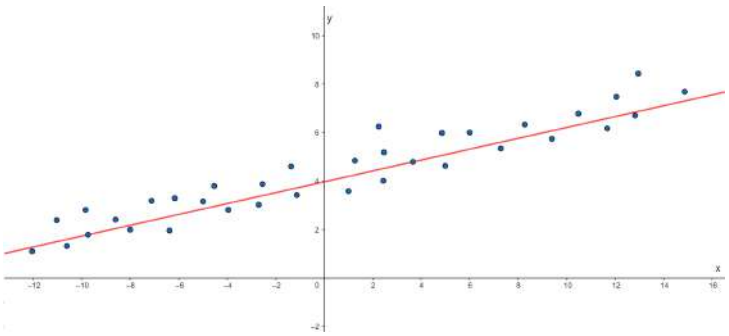

Figura 4 – Problema de regressão linear.

A Imagem 4 nos demonstra a ideia de estimar uma função desconhecida. A partir da amostra de treino, representada pelos pontos em azul, o algoritmo decide que os dados do problema tem um comportamento similar à função em vermelho, por ele retornada.

Como a imagem também sugere, erros na estimação de valores também devem ser levados em conta. Enquanto que, neste capítulo, apresentamos uma visão superficial destes conceitos, o desenvolvimento deste trabalho é voltado para sua definição formal, utilizando a teoria de Probabilidade como ferramenta para tal.

3 Aprendizado PAC

Neste capítulo, são apresentados os conceitos que serão utilizados para definir de forma concisa uma noção de aprendizado baseada em minimizar o erro cometido pelo algoritmo na amostra de treino a ele fornecida.

3.1 Um modelo simples de aprendizado

Inicialmente, supomos que um problema de aprendizado é composto por um conjunto de entrada $\mathcal{X}$ e um conjunto de saída $\mathcal{Y}$, onde seus dados são pares $(x, y) \in \mathcal{X} \times \mathcal{Y}$, sendo $y = f(x)$ onde $f : \mathcal{X} \to \mathcal{Y}$ é fixa e denominada função alvo.

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 desconhecida D em $\mathcal{X}$.

A partir disto, o algoritmo deve retornar uma função, ou hipótese, $h : \mathcal{X} \to \mathcal{Y}$ capaz de rotular, assim como f, pontos do domínio que estejam fora de S. Ou seja, baseado nos dados de treino, o algoritmo deve obter uma boa estimação para f.

Observe que S é composto pela realização de m eventos aleatórios independentes, onde cada um destes eventos resulta na obtenção de uma amostra $x \in \mathcal{X}$ através de uma distribuição de probabilidade D sobre $\mathcal{X}$. Conforme comentado na seção de notação, isto significa dizer que a obtenção de um ponto do domínio é uma variável aleatória X, cuja probabilidade de realização, a probabilidade no contra-domínio, é dada pela medida de probabilidade D.

Ademais, em alguns momentos, principalmente na demonstração de resultados, estaremos interessados na esperança ou probabilidade da obtenção de uma amostra, e não estaremos mais nos referindo a uma realização específica $S = \{(x_1, y_1), ..., (x_m, y_m)\}$. Nestes casos, estaremos lidando com um vetor aleatório $\mathcal{S} = ((X_1, Y_1), ..., (X_m, Y_m))$.

Durante as próximas seções, iremos considerar apenas problemas de aprendizado em que $\mathcal{Y} = \{0, 1\}$, ou seja, problemas de classificação binária. Tal escolha foi feita visando uma simplificação da exposição dos resultados e suas demonstrações.

3.2 Medindo a acurácia: risco e risco empírico

Evidentemente, precisamos de maneiras de quantificar a eficiência de um algoritmo de aprendizado. Para tanto, são definidas duas noções de erro, uma computada sobre

uma amostra de treino e outra sobre qualquer ponto do domínio $\mathcal{X}$ tomado de maneira aleatória.

Definição 3.1. O risco empírico de uma hipótese h em relação a uma função alvo f e a uma amostra de treino $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ é a discrepância em relação a função alvo f nas instâncias de S. É dado por

$$L_S(h) := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[h(x_i).\neq f(x_i)]}.$$

Definição 3.2. O risco de uma hipótese h em relação a uma função alvo f e a uma distribuição de probabilidade D é a probabilidade de tomarmos um ponto x qualquer do domínio, de acordo com uma distribuição de probabilidade D, tal que tenhamos $h(x) \neq f(x)$. Ou seja

$$L_{(D,f)}(h) := \mathop{\mathbb{P}}_{X \sim D}[h(X) \neq f(X)] = D(\{x \mid h(x) \neq f(x)\}) = \mathop{\mathbb{E}}_{X \sim D}\left[\mathbb{1}_{h(X) \neq f(X)}\right].$$

Portanto, o risco empírico de h é o erro médio sobre uma amostra S, enquanto que o risco, ou erro de generalização é o seu erro esperado baseado na distribuição D, isto ocorre porque para uma variável aleatória Z que toma seus valores em $\{0, 1\}$, $\mathop{\mathbb{E}}_{Z \sim D}[Z] = \mathop{\mathbb{P}}_{Z \sim D}[Z = 1]$.

É fácil perceber que estamos interessados em hipóteses que possuam baixo risco, no sentido da Definição 3.2, uma vez que isto implica que o algoritmo comete poucos erros de predição. No entanto, como o algoritmo não tem acesso à distribuição de probabilidade D que gera as amostras do problema, e nem a função alvo f, $L_{(D,f)}$ não pode ser calculado diretamente. No entanto, pode-se utilizar o risco empírico como sua aproximação.

A fim de garantir que o risco empírico é realmente uma boa aproximação para o risco, deve-se assegurar que a distribuição de probabilidade utilizada para obter a amostra de treino é a mesma utilizada para obter pontos fora dela, pois, neste caso, $\mathop{\mathbb{E}}_{\mathcal{S} \sim D}[L_{\mathcal{S}}(h)] = L_{(D,f)}(h)$. Além disso, no caso de uma amostra independente e identicamente distribuída com distribuição D, pela Lei Forte dos Grandes Números, obtemos que, com probabilidade 1, $\lim_{m \to \infty} L_{\mathcal{S}}(h) = L_{(D,f)}(h)$, o que sugere que o risco empírico é uma boa aproximação para o risco. Conforme será visto mais adiante, este fato é fundamental para garantir o aprendizado, segundo o princípio que será exposto na próxima seção.

3.3 Minimização do risco empírico

Conforme discutido anteriormente, o algoritmo de aprendizado, a partir de uma amostra de treino S, deve retornar uma hipótese capaz de relacionar corretamente pares (x, y) que não estejam na amostra de treino.

O caminho mais natural para contornar a impossibilidade do cálculo do risco é que o algoritmo retorne uma hipótese que possua baixo risco empírico, uma vez que este pode ser facilmente computado.

Entretanto, veremos no exemplo a seguir que uma hipótese que possui baixo risco empírico não necessariamente possui um risco pequeno.

Exemplo 3.3. *Seja $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{0, 1\}$, D uma distribuição de probabilidade sobre $\mathcal{X}$ tal que todas as suas instâncias estejam uniformemente distribuídas no quadrado $ABCD$ que possui área 2 e seja $f : \mathcal{X} \to \mathcal{Y}$ uma função cujo valor é 1 se x pertence ao quadrado $EFGH$, que possui área 1, e 0 caso contrário. Tal situação é representada abaixo: para os pontos em vermelho, tem-se $f(x) = 0$, e para os pontos em azul, $f(x) = 1$.*

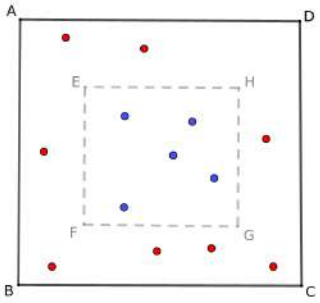

Figura 5 – Representação da distribuição dos pontos do domínio.

Desta maneira, temos que

$$\mathop{\mathbb{P}}_{X\sim D}[f(X) = 1] = \mathop{\mathbb{P}}_{X\sim D}[X \in EFGH] = \frac{\text{área}(EFGH)}{\text{área}(ABCD)}.$$

Além disto, dada uma amostra $\{(x_1, y_1), ..., (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ obtida de forma independente e identicamente distribuída de acordo com D, considere a hipótese

$$h_S(x) = \begin{cases} y_i, & \text{se } \exists i \in \{1, ..., m\} \ : \ x_i = x; \\ 0, & \text{caso contrário.} \end{cases}$$

Pela definição acima, temos que, para qualquer amostra S, $L_S(h_S) = 0$. Por outro lado, note que o evento $\{x \in \mathcal{X} \ : \ f(x) \neq h_S(x)\}$ pode ser escrito como $\{x \in \mathcal{X} \ : \ x \notin S \wedge f(x) = 1\}$. No entanto, como D é uniforme no quadrado $ABCD$ e S é um conjunto finito de tamanho m, $\mathop{\mathbb{P}}_{X\sim D}[X \in S] = 0$, de modo que

$$L_{(D,f)}(h_S) = \mathop{\mathbb{P}}_{X\sim D}[h_S(X) \neq f(X)] = \mathop{\mathbb{P}}_{X\sim D}[f(X) = 1] = \frac{\text{área}(EFGH)}{\text{área}(ABCD)} = \frac{1}{2}.$$

Portanto, embora h_S possua risco empírico zero, ela possui risco elevado. Tal comportamento é chamado de overfitting e será discutido detalhadamente no próximo capítulo. □

Assim, ao invés de deixar que o algoritmo selecione qualquer hipótese, a ele é fornecida uma classe de hipóteses, isto é, um conjunto $\mathcal{H} = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$, de modo que sua escolha ficará restrita a esse conjunto. Tal ideia é o paradigma utilizado para formular a noção de aprendizado PAC, e é definida abaixo.

Definição 3.4 (Minimização do Risco Empírico com viés indutivo)**.** Dada uma classe de hipóteses $\mathcal{H} = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$ e uma amostra de treino $S = \{(x_1, y_1), ..., (x_m, y_m)\}$, $\underset{\mathcal{H}}{\text{ERM}}(S)$ [1] é uma hipótese $h_S \in \mathcal{H}$ que minimiza o risco empírico em relação a S, isto é,

$$\underset{\mathcal{H}}{\text{ERM}}(S) \in \underset{h \in \mathcal{H}}{\operatorname{argmin}}\, L_S(h).$$

O termo viés indutivo é utilizado para ressaltar o fato de que ao selecionar uma classe de hipóteses específica, impõe-se um viés sobre o algoritmo. Em outras palavras, assume-se que a escolha de uma dada classe de hipóteses pode acarretar em uma perda de performance do algoritmo. Este é um preço a se pagar uma vez que não se pode rodar o algoritmo sobre todas as hipóteses concebíveis.

3.4 Aprendizado PAC

Conforme discutido na seção anterior, não se pode garantir que qualquer hipótese que minimize o risco empírico possuirá, de fato, boa capacidade de generalização.

Neste capítulo será introduzido o conceito de aprendizado PAC, cuja definição nos esclarecerá quais hipóteses são necessárias para garantir que o princípio de minimização do risco empírico com viés indutivo retorne uma hipótese que possua baixo risco. Por fim, será mostrado que classes de hipóteses finitas são PAC aprendíveis.

Retomando o que já foi visto, dados dois conjuntos $\mathcal{X}$ e $\mathcal{Y}$, uma distribuição de probabilidade D, uma função $f : \mathcal{X} \to \mathcal{Y}$, uma amostra $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ e uma classe de hipóteses $\mathcal{H} = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$ um algoritmo de aprendizado que é embasado no princípio ERM, tendo acesso a S, retorna uma hipótese $h_S \in \mathcal{H}$ que possui risco empírico mínimo em relação as outras hipóteses de $\mathcal{H}$, ou seja, h_S não difere muito de f nos pontos pertencentes a S.

Queremos, agora, descobrir quais exigências devem ser impostas sobre os objetos que compõem o problema de aprendizado para garantir que a hipótese retornada pelo algoritmo apresente uma margem de erro aceitável em pontos $x \in \mathcal{X}$ aos quais o algoritmo nunca teve acesso, em outras palavras, que esta hipótese retornada de fato é uma boa aproximação para f.

No entanto, como tanto a distribuição de probabilidade D e a função f são desconhecidas, não podemos fazer nenhuma suposição sobre elas. Ademais, como estamos

[1] A sigla ERM é uma abreviatura em inglês para empirical risk minimization.

definindo um modelo de aprendizado generalizado, não faria sentido impor alguma restrição a priori sobre a classe de hipóteses que será utilizada pelo algoritmo. Portanto, na definição de aprendizado PAC, utiliza-se apenas informações relativas à amostra de treino disponível.

Com isto em mente, temos a seguinte definição:

Definição 3.5 (Aprendizado PAC)**.** Uma classe de hipóteses $\mathcal{H}$ é dita PAC aprendível se existe uma função $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ e um algoritmo de aprendizado com a seguinte propriedade: para cada $\epsilon, \delta \in (0,1)$, para cada distribuição D sobre $\mathcal{X}$ e para cada função $f : \mathcal{X} \to \mathcal{Y} = \{0,1\}$, ao rodar o algoritmo em $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ realizações independentes e identicamente distribuídas geradas por D e que satisfaçam $y = f(x)$, o algoritmo retorna um hipótese h tal que,

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}[L_{(D,f)}(h) \leq \epsilon] \geq 1 - \delta.$$

O termo PAC é abreviatura em inglês para Probably Aproximately Correctly, que em português pode ser traduzido como Provavelmente Aproximadamente Correto. O parâmetro ϵ representa a qualidade da predição (acurácia) do algoritmo, ele determina o quão longe o classificador de saída pode estar do ótimo, é de onde vem a expressão "aproximadamente correto". Já δ é a confiança na amostra, ou em outras palavras, a probabilidade de obter uma amostra de treino que não reflita bem as características da distribuição de probabilidade que a gerou, de modo que $(1-\delta)$ é o parâmetro de confiança de nossa predição e está relacionado ao "provavelmente" da abreviatura PAC. Ambos devem ser prefixados.

A função $m_{\mathcal{H}}$ é chamada de complexidade de amostra. Escolhidos ϵ e δ, ela nos indica o tamanho da amostra de treino S necessário para que o algoritmo tenha a acurácia e confiança desejadas.

A seguir, veremos um exemplo clássico e muito usado para ilustrar o aprendizado PAC. Mais detalhes podem ser obtidos em [5] e [6].

Exemplo 3.6. *Considere retângulos em $\mathbb{R}^2$, onde cada retângulo mapeia um ponto do plano $\boldsymbol{x} = (u, v) \in \mathbb{R}^2$, em 1 se ele está no retângulo, e -1, caso contrário.*

Um retângulo no plano pode ser expresso como uma função $h_{l,r,b,t} : \mathbb{R}^2 \to \{-1, 1\}$ dada por

$$h_{l,r,b,t}(\boldsymbol{x}) = \begin{cases} 1, & \text{se } u \in [l,r] \text{ e } v \in [b,t]; \\ -1, & \text{caso contrário.} \end{cases}$$

Seja $\mathcal{H} = \{h_{l,r,b,t} \mid l < r,\ b < t\}$, vamos provar que esta classe é PAC aprendível, supondo que a função alvo pertence a essa classe. Para tanto, consideraremos um algoritmo de aprendizado que, dada uma amostra rotulada S, retorna o retângulo alinhado ao eixo "mais estreito" contendo pontos rotulados com 1. Ele pode ser representado da seguinte maneira:

AprendeRetângulo

Entrada: *m pontos* $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m) \in \mathbb{R}^2 \times \{-1, 1\}$*;*

Defina:

$$l' = \max_{i\,:\,y_i=1} u_i; \qquad r' = \min_{i\,:\,y_i=1} u_i;$$
$$b' = \max_{i\,:\,y_i=1} v_i; \qquad t' = \min_{i\,:\,y_i=1} v_i.$$

Retorna: $h_{l',r',b',t'} = h_S \in \mathcal{H}$.

Fixe uma distribuição D sobre $\mathbb{R}^2$, uma função $f : \mathbb{R}^2 \to \{-1, 1\}$ e ϵ, $\delta \in [0, 1]$. Seja $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ obtida de maneira independente e identicamente distribuída de acordo com a distribuição D.

Supondo que a função alvo f pertence à classe de hipóteses $\mathcal{H}$, temos $f = f_{l^\star,r^\star,b^\star,t^\star}$. Sendo assim, definimos

$$R_f = [l^\star, r^\star] \times [b^\star, t^\star]$$

como sendo o retângulo formado pela função alvo f e

$$R_{h_S} = [l', r'] \times [b', t']$$

como o retângulo formado pela hipótese h_S, que é retornada pelo algoritmo AprendeRetângulo.

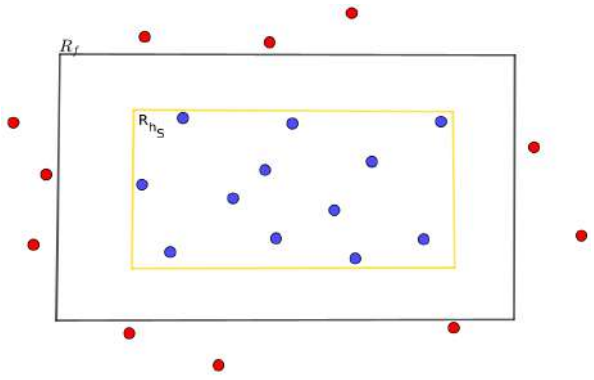

Figura 6 – Representação das regiões retangulares R_f e R_{h_S}.

Na figura acima, temos uma representação gráfica das regiões retangulares definidas e dos pontos pertencentes à amostra de treino. Os pontos em vermelho são classificados como 0, e os pontos em azul como 1.

Note que, devido a estas definições, a região formada pelo retângulo da função alvo subtraída do retângulo da hipótese, isto é $R_L = \{x : x \in R_f \wedge x \notin R_{h_S}\}$*, é tal que*

$$\mathop{\mathbb{P}}_{X\sim D}[X \in R_L] = L_{(D,f)}(h_S)$$

onde $\mathop{\mathbb{P}}_{X\sim D}[X \in R_L]$ *é a massa de probabilidade da região* R_L*.*

Portanto, queremos encontrar um tamanho m *para a amostra de treino tal que*

$$\mathop{\mathbb{P}}_{S\sim D^m}\left[L_{(D,f)}(h_S) \geq \epsilon\right] \leq \delta.$$

Para isto, definimos

$$z_1 = \sup\left\{z \ : \ \mathop{\mathbb{P}}_{X\sim D}\Big[X \in [l,r]\times[z,t]\Big] \geq \frac{\epsilon}{4}\right\}$$
$$z_2 = \sup\left\{z \ : \ \mathop{\mathbb{P}}_{X\sim D}\Big[X \in [z,r]\times[b,t]\Big] \geq \frac{\epsilon}{4}\right\}$$
$$z_3 = \inf\left\{z \ : \ \mathop{\mathbb{P}}_{X\sim D}\Big[X \in [l,r]\times[b,z]\Big] \geq \frac{\epsilon}{4}\right\}$$
$$z_4 = \inf\left\{z \ : \ \mathop{\mathbb{P}}_{X\sim D}\Big[X \in [l,z]\times[b,t]\Big] \geq \frac{\epsilon}{4}\right\}.$$

Sendo assim, temos 4 regiões retangulares pertencentes a R_f *cuja massa de probabilidade é no mínimo* $\epsilon/4$*, a saber*

$$R_1 = [l,r]\times[z_1,t]$$
$$R_2 = [z_2,r]\times[b,t]$$
$$R_3 = [l,r]\times[b,z_3]$$
$$R_4 = [l,z_4]\times[b,t].$$

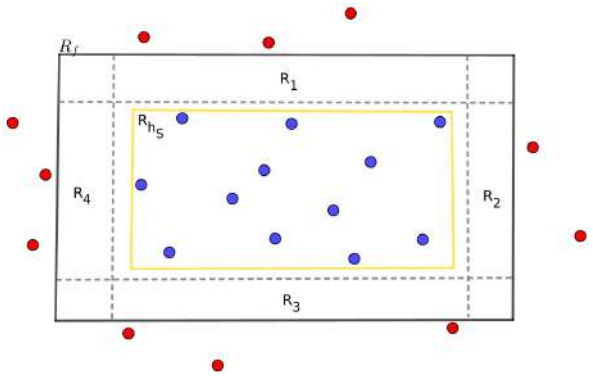

Figura 7 – Representação das regiões retangulares R_1, R_2, R_3 e R_4.

Uma vez que as propriedades de probabilidade nos garantem que dados dois conjuntos A e B quaisquer, se $A \subset B$ então $\mathbb{P}(A) \leq \mathbb{P}(B)$, as regiões

$$\begin{aligned}\overline{R}_1 &= [l, r] \times (z_1, t] \\ \overline{R}_2 &= (z_2, r] \times [b, t] \\ \overline{R}_3 &= [l, r] \times [b, z_3) \\ \overline{R}_4 &= [l, z_4) \times [b, t]\end{aligned}$$

possuem massa de probabilidade de no máximo $\epsilon/4$.

Sendo assim, se R_{h_S} intersecta R_i para todo $i \in [4]$, por ser um retângulo, R_{h_S} terá um lado em cada uma dessas quatro regiões. Desse modo, R_L deve estar contido em $\bigcup_{i=1}^{4} \overline{R}_i$ de forma que

$$L_{(D,f)}(h_S) = \mathop{\mathbb{P}}_{X \sim D}[X \in R_L] \leq \mathop{\mathbb{P}}_{X \sim D}\left[X \in \bigcup_{i=1}^{4} \overline{R}_i\right] \leq \sum_{i=1}^{4} \mathop{\mathbb{P}}_{X \sim D}[X \in \overline{R}_i] < \sum_{i=1}^{4} \frac{\epsilon}{4} = \epsilon. \quad (3.1)$$

Finalmente, pela contra-positiva, se $L_{(D,f)}(h_S) > \epsilon$ então R_L não intersecta R_i para algum $i \in [4]$ de modo que

$$\begin{aligned}\mathop{\mathbb{P}}_{S \sim D^m}[L_{(D,f)}(h_S) \geq \epsilon] &\leq \mathop{\mathbb{P}}_{S \sim D^m}[\exists i \in [4] \ : \ R_{h_S} \cap R_i = \emptyset] \\ &= \mathop{\mathbb{P}}_{S \sim D^m}\left[\bigcup_{i=1}^{4} \{R_{h_S} \cap R_i = \emptyset\}\right] \\ &\leq \sum_{i=1}^{4} \mathop{\mathbb{P}}_{S \sim D^m}[R_{h_S} \cap R_i = \emptyset] \quad (3.2) \\ &\leq \sum_{i=1}^{4} \left[\prod_{j=1}^{m} \left(1 - \frac{\epsilon}{4}\right)\right] \quad (3.3) \\ &= 4\left(1 - \frac{\epsilon}{4}\right)^m.\end{aligned}$$

A desigualdade (3.2) segue da cota da união de eventos de probabilidade, enquanto que a desigualdade (3.3) segue de $\mathop{\mathbb{P}}_{X \sim D}[X \in R_i] \leq \epsilon/4$ e da independência da obtenção de cada elemento da amostra. Utilizando o fato de que $1 - x \leq e^{-x}$ temos que

$$\mathop{\mathbb{P}}_{S \sim D^m}[L_{(D,f)}(h_S) \geq \epsilon] \leq 4e^{-m\epsilon/4}.$$

Assim, se $m \geq \frac{4}{\epsilon} \ln \frac{4}{\delta}$, então

$$\mathop{\mathbb{P}}_{S \sim D^m}[L_{(D,f)}(h_S) \geq \epsilon] \leq \delta.$$

Ou seja, mostramos que, dada uma distribuição de probabilidade D sobre $\mathcal{X}$, uma função $f : \mathcal{X} \to \mathcal{Y}$ e $\epsilon, \delta \in [0, 1]$, existe uma complexidade de amostra e um algoritmo de aprendizado, denominado AprendeRetângulo, tal que ao rodarmos este algoritmo em pelo menos $\frac{4}{\epsilon} \ln \frac{4}{\delta}$

amostras de treino obtidas de maneira independente e identicamente distribuída de acordo com D, com probabilidade maior ou igual a $1-\delta$, a hipótese $h_S \in \{h_{l,r,b,t} \mid l < r,\ b < t\}$ retornada pelo algoritmo satisfaz $L_{(D,f)}(h_S) \leq \epsilon$. Portanto, $\mathcal{H} = \{h_{l,r,b,t} \mid l < r,\ b < t\}$ é PAC aprendível, desde que a função alvo f pertença a $\mathcal{H}$. □

O objetivo final deste capítulo é demonstrar que toda classe de hipóteses finita é PAC aprendível, sendo que a complexidade de amostra depende do seu número de elementos, ou seja, de $|\mathcal{H}|$.

3.4.1 A hipótese de consistência

No exemplo anterior, é suposto que a função alvo faça parte da classe de hipóteses. Como veremos a seguir, esta exigência define o conceito de consistência, sob o qual o aprendizado PAC é fundamentado.

Definição 3.7 (Hipótese de consistência)**.** Dado uma distribuição de probabilidade D e uma função $f : \mathcal{X} \to \mathcal{Y}$, uma classe de hipóteses $\mathcal{H} = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$ é dita consistente se

$$\exists\, h^* \in \mathcal{H} \text{ tal que } L_{(D,f)}(h^*) = 0.$$

Embora a definição de consistência seja clara e objetiva, devemos lembrar que, com esta formulação, não temos como verificar se uma dada classe de hipóteses é consistente, uma vez que não somos capazes de calcular o risco de seus elementos. No entanto, se uma hipótese possui risco igual a zero, então ela também terá risco empírico zero com probabilidade um.

Sendo assim, dada uma classe de hipóteses $\mathcal{H}$ consistente, para qualquer amostra de treino S obtida de forma independente e identicamente distribuída de acordo com a distribuição de probabilidade D para qual a classe é consistente, existe pelo menos uma hipótese $h \in \mathcal{H}$ cujo risco empírico é igual a zero. Ademais, como a noção de aprendizado PAC segue o paradigma de minimização do risco empírico (ERM), no caso consistente, toda hipótese retornada por um algoritmo de aprendizado necessariamente tem risco empírico igual a zero.

Finalmente, supondo que a hipótese de consistência é satisfeita, vamos mostrar que toda classe de hipóteses finita é PAC aprendível, obtendo a complexidade de amostra necessária. Para isto, obteremos uma cota para a probabilidade de uma hipótese consistente não possuir capacidade de generalização. Como não sabemos qual hipótese será retornada pelo algoritmo, precisaremos de uma cota uniforme, ou seja, que se mantenha válida para todas as hipóteses $h \in \mathcal{H}$ tais que $L_{(D,f)}(h) = 0$.

Proposição 3.8. *Toda classe de hipóteses finita que satisfaz a definição da hipótese de consistência (Definição 3.7) é PAC aprendível, com complexidade de amostra*

$$m_{\mathcal{H}} \leq \left\lceil \frac{log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil.^{2}$$

Demonstração. Fixe $\epsilon > 0$. Para mostrar que a classe de hipóteses é PAC aprendível, devemos encontrar um tamanho mínimo m para a amostra de treino S, tal que, com probabilidade de pelo menos $1 - \delta$ sobre a obtenção de tal amostra de acordo com uma distribuição de probabilidade D, o risco da hipótese retornada por qualquer algoritmo de aprendizado seja menor ou igual a ϵ. Para isto, consideraremos seu evento complementar $\{L_{(D,f)}(h_S) > \epsilon\}$.

No entanto, sabemos, pela hipótese de consistência, que a hipótese retornada deverá ter risco empírico igual a zero. Assim vamos cotar a probabilidade de existir $h \in \mathcal{H}$ tal que

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\Big[L_S(h) = 0 \ \wedge \ L_{(D,f)}(h) > \epsilon\Big].$$

Defina $\mathcal{H}_\epsilon = \{h \in \mathcal{H} \ : \ L_{(D,f)}(h) > \epsilon\} \subseteq \mathcal{H}$. A probabilidade de uma hipótese $h \in \mathcal{H}_\epsilon$ possuir risco empírico zero pode ser cotada como

$$\begin{aligned}
\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}[L_{\mathcal{S}}(h) = 0] &= \mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}[\forall i \in [m], \ h(X_i) = f(X_i)] \\
&= \prod_{i=1}^{m} \mathop{\mathbb{P}}_{X\sim D}[h(X) = f(X)] \\
&= \prod_{i=1}^{m} \Big(1 - L_{(D,f)}(h)\Big) \\
&\leq \prod_{i=1}^{m} 1 - \epsilon \\
&= (1-\epsilon)^m. \qquad (3.4)
\end{aligned}$$

Portanto,

$$\begin{aligned}
\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\Big[\exists\, h \in \mathcal{H}_\epsilon \ : \ L_{\mathcal{S}}(h) = 0\Big] &= \mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\left[\bigcup_{h\in\mathcal{H}_\epsilon} \{L_{\mathcal{S}}(h) = 0\}\right] \\
&\leq \sum_{h\in\mathcal{H}_\epsilon} \mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\Big[L_{\mathcal{S}}(h) = 0\Big]. \qquad (3.5)
\end{aligned}$$

Onde a desigualdade (3.5) segue da cota da união de eventos de probabilidade. Finalmente, aplicando (3.4) em (3.5), temos

$$\sum_{h\in\mathcal{H}_\epsilon} \mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\Big[L_{\mathcal{S}}(h) = 0\Big] \leq \sum_{h\in\mathcal{H}_\epsilon} (1-\epsilon)^m = |\mathcal{H}_\epsilon|(1-\epsilon)^m \leq |\,\mathcal{H}\,|(1-\epsilon)^m. \qquad (3.6)$$

A complexidade de amostra é obtida utilizando o fato de que $1 - \epsilon \leq e^{-\epsilon}$ e resolvendo a equação (3.6) para m. ■

[2] Neste contexto, $\lceil x \rceil$ indica a função teto. O teto de um número real x é o primeiro inteiro maior ou igual a x. Ademais, o logaritmo está sendo tomado na base e.

Este resultado pode ser escrito de maneira equivalente, sendo resolvido para ϵ, como: se $\mathcal{H}$ é uma classe de hipóteses finita, $\forall\ \epsilon, \delta > 0$, com probabilidade de pelo menos $1 - \delta$,

$$L_{(D,f)}(h_S) \leq \frac{1}{m}\left(log|\,\mathcal{H}\,| \ +\ log\frac{1}{\delta}\right).$$

Conforme nossa intuição poderia esperar, o termo $1/m$ indica que os algoritmos de aprendizado se beneficiam da disponibilidade de grandes quantidades de dados de treino. O termo $log(\mathcal{H})$, que difere de $log_2(\mathcal{H})$ apenas por uma constante, pode ser interpretado como o número de bits necessário para representar $\mathcal{H}$ computacionalmente.

Embora a hipótese de consistência seja útil na teoria, na maioria dos casos reais ela pode ser forte demais para ser satisfeita. Isto pode acontecer se a classe de hipóteses não for complexa o suficiente para conseguir obter uma aproximação da função alvo ou o problema abordado pode simplesmente ser muito difícil de se resolver. Sendo assim, nos próximos capítulos, será abordada uma noção mais generalizada de aprendizado.

3.5 Generalizando o modelo de aprendizado

No início deste capítulo, foi descrito um modelo de aprendizado que tem como suposição a existência de uma função f que é utilizada para relacionar pontos do domínio, obtidos de acordo com uma distribuição de probabilidade, aos pontos do conjunto de saída do problema. Em outras palavras, neste modelo, os dados do problema são pares $(x, y) \in \mathcal{X} \times \mathcal{Y}$ onde $y = f(x)$.

No entanto, esta formulação impede que pontos distintos do domínio que possuam mesmo valor de entrada tenham um valor de saída diferente, o que não reflete uma gama de problemas reais. Por exemplo, imagine que estamos interessados em prever o sexo de uma pessoa baseado em sua altura e peso. Neste caso, poderia acontecer de duas pessoas possuírem mesma altura e peso mas serem de sexos diferentes. Sendo assim, podemos estender nosso modelo de aprendizado, para que ele possa abranger diferentes problemas.

Neste caso mais geral, a função f é substituída por uma noção mais flexível onde dados um conjunto de entrada $\mathcal{X}$ e um conjunto de saída $\mathcal{Y}$, que compõem o problema a ser estudado, temos uma distribuição de probabilidade conjunta D sobre $\mathcal{X} \times \mathcal{Y}$. Esta distribuição conjunta pode ser entendida como sendo composta por duas partes [3]:

- Uma distribuição marginal D_x sobre pontos $x \in \mathcal{X}$, que nos dá a probabilidade de obtermos um ponto de entrada com certas características. No nosso exemplo, ela nos retornaria a probabilidade de obtermos uma pessoa com peso e altura em uma determinada faixa de valores.

- Uma distribuição condicional sobre os valores $y \in \mathcal{Y}$ para cada ponto do domínio $\mathcal{X}$, $D((x, y)|x)$. Ou seja, a probabilidade de uma pessoa com uma dada altura e peso ser do sexo feminino, por exemplo.

Chamamos este modelo de aprendizado de estocástico, e o discutido no início deste capítulo de determinístico, devido às suas formulações particulares.

De maneira similar ao modelo determinístico, o algoritmo recebe como entrada uma amostra de treino S, que é obtida de acordo com a distribuição de probabilidade D, e utilizando o paradigma de minimização do risco empírico com viés indutivo, deve retornar uma hipótese $h : \mathcal{X} \to \mathcal{Y}$ pertencente a uma classe $\mathcal{H}$, que possua baixo risco empírico sobre esta amostra. Porém, as Definições 3.1 e 3.2 apresentadas anteriormente para o risco e o risco empírico de uma hipótese devem ser reescritas para o caso estocástico.

Definição 3.9. Dada uma distribuição de probabilidade conjunta D sobre $\mathcal{X} \times \mathcal{Y}$ e uma hipótese $h : \mathcal{X} \to \mathcal{Y}$, definimos o risco empírico de h em relação a uma amostra $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ de tamanho m obtida de maneira independente e identicamente distribuída de acordo com a distribuição D por

$$L_S(h) := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[h(x_i) \neq y_i]}.$$

Além disto, o risco de h em relação a D é dado por

$$L_D(h) := \mathop{\mathbb{P}}_{(X,Y)\sim D}[h(X) \neq Y] = D(\{(x, y) \mid h(x) \neq y\}) = \mathop{\mathbb{E}}_{(X,Y)\sim D}[\mathbb{1}_{h(X)\neq Y}]$$

Ademais, outro fato que deve ser observado é que, até agora, estávamos apenas considerando problemas de classificação binária, ou seja, problemas em que $\mathcal{Y} = \{0, 1\}$. Para que o modelo de aprendizado também possa ser utilizado em outras classes de problemas, precisaremos, novamente, rever as definições do erro de uma dada hipótese.

Foi determinado que o risco empírico de uma hipótese em relação a uma amostra de treino é dado pela proporção de pontos da amostra aos quais a hipótese erra na predição, o que é representado pelo termo $\mathbb{1}_{[h(x_i)\neq y_i]}$. Todavia, em um problema de regressão, como o conjunto $\mathcal{Y}$ é contínuo, faria muito mais sentido que a noção de erro para um par (x, y) fosse dada em função da diferença entre $h(x)$ e y. Sendo assim, em diferentes problemas de aprendizado utilizamos diferentes maneiras de penalizar o erro de uma hipótese.

Definição 3.10. Dada uma classe de hipóteses $\mathcal{H}$ e algum conjunto $\mathcal{Z}$, uma função de perda é uma função $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ responsável por fazer a penalização dos erros cometidos pelas hipóteses de $\mathcal{H}$.

[3] Uma definição formal destes conceitos pode ser encontrada no apêndice A.

Exemplo 3.11. *A função de perda*

$$l_{0-1}(h,(x,y)) := \begin{cases} 0, & \text{se } h(x) = y \\ 1, & \text{se } h(x) \neq y \end{cases}$$

é utilizada em problemas de classificação binária. □

Exemplo 3.12. *A função de perda de erro quadrático*

$$l_{sq}(h,(x,y)) := (h(x) - y)^2$$

é utilizada em problemas de regressão. □

Desta forma, redefine-se o risco e o risco empírico de maneira a considerar a função de perda específica que será utilizada em cada problema.

Definição 3.13. Dada uma distribuição de probabilidade conjunta D sobre um conjunto $\mathcal{Z}$, uma hipótese h definida em $\mathcal{Z}$ e uma função de perda $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$, definimos o risco empírico de h em relação a uma amostra $S = \{z_1, ..., z_m\}$ de tamanho m obtida de maneira independente e identicamente distribuída de acordo com a distribuição D por

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

Além disto, o risco de h em relação a D e l é dado por

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

Assim, é fácil percebermos a correlação entre as Definições 3.9 , onde estávamos lidando especificamente com a função de perda l_{0-1}, e 3.13. Para isto note que, para uma variável aleatória Z que toma seus valores em $\{0, 1\}$, $\mathop{\mathbb{E}}_{Z \sim D}[Z] = \mathop{\mathbb{P}}_{Z \sim D}[Z = 1]$.

O risco empírico é então formulado como sendo a média da função de perda que uma hipótese comete sobre uma dada amostra de treino, e o risco como sendo a esperança da função de perda em relação a hipótese considerada.

3.6 Aprendizado PAC agnóstico

Interessado em estender o modelo de aprendizado PAC para ser capaz de abranger o modelo estocástico com função de perda generalizada, David Haussler (ver [7]), baseado na teoria de decisão estatística, propôs uma outra formulação para o problema de aprendizado, a qual seguiremos.

Dados um conjunto de entrada $\mathcal{X}$, um conjunto de saída $\mathcal{Y}$, uma classe de distribuições de probabilidade conjunta $\mathcal{D}$ sobre $\mathcal{X} \times \mathcal{Y}$ e um conjunto A, o qual chamaremos de espaço de decisão, o algoritmo de aprendizado recebe uma amostra de treino

$S = \{(x_1, y_1), ..., (x_m, y_m)\}$ de tamanho m obtida de maneira independente e identicamente distribuída de acordo com uma distribuição $D \in \mathcal{D}$. Após o processo de treinamento, o algoritmo recebe outros exemplos, pares $(x, y) \in \mathcal{X} \times \mathcal{Y}$, obtidos de acordo com a mesma distribuição D, de forma que ele só tenha acesso a informação de entrada $x \in \mathcal{X}$. Então, o algoritmo seleciona uma ação $a \in A$ e o valor de saída $y \in \mathcal{Y}$ é a ele revelado, sendo que este depende apenas da entrada correspondente x, e não da ação a escolhida. Note que nenhuma suposição é feita sobre os conjuntos $\mathcal{X}$, $\mathcal{Y}$ e A, de forma que eles são arbitrários.

Para cada ação a e saída y, o algoritmo sofre uma penalidade que é computada por uma função de perda $l : \mathcal{Y} \times A \to \mathbb{R}$ por ele conhecida. A forma que o algoritmo seleciona uma ação a baseado em um valor de entrada x é uma função h chamada de regra de decisão, que pertence a uma classe de funções $\mathcal{H} = \{h|\ h : \mathcal{X} \to A\}$ a qual denominamos espaço de decisão. Desta maneira, o algoritmo busca encontrar uma regra de decisão $h \in \mathcal{H}$ que minimiza o erro esperado de suas ações, tendo como base exemplos obtidos através de uma distribuição de probabilidade a ele desconhecida.

Embora esta formulação possa parecer muito diferente do que foi apresentado nas seções anteriores, é fácil verificar que ela compreende o modelo determinístico de um problema de classificação binária, utilizado para a definição de aprendizado PAC. Para isto, basta notar que, naquele caso, tendo $\mathcal{Y} = \{0, 1\}$ e apenas a entrada x de um exemplo (x, y) sendo obtida de acordo com uma distribuição de probabilidade, com o valor de saída y dado por uma função $f : \mathcal{X} \to \mathcal{Y}$ desconhecida, o espaço de decisão A é igual ao espaço de saída $\mathcal{Y}$, e uma ação a pode ser interpretada com uma predição de um valor de entrada. Sendo assim, uma regra de decisão $h \in \mathcal{H}$ é um mapa de $\mathcal{X}$ para $\mathcal{Y}$, bem como a função alvo f.

Ademais, outra forma de percebermos a similaridade entre os dois modelos se dá através da Definição 3.11. No modelo de Haussler, em geral, quando estamos interessados em estimar uma função desconhecida, a função de perda $l(y, a)$ mede a distância entre a predição a e o correto valor de saída y em alguma métrica. Em particular, o modelo PAC utiliza a métrica discreta: $l(y, a) = 0$ se $a = y$, senão, $l(y, a) = 1$, de forma que a perda esperada de uma regra de decisão (hipótese) seja a probabilidade de uma predição incorreta, que é justamente a definição que foi feita para o risco na Seção 3.2.

Esta independência da métrica utilizada pela função de perda é a característica que permite que o aprendizado PAC agnóstico seja utilizado em diversos problemas, onde o conjunto $\mathcal{Y}$ difere de $\{0, 1\}$.

De forma análoga à Definição 3.13, determina-se que o risco de uma regra de decisão h seja o valor médio de $l(y, h(x))$ quando (x, y) é obtido através de uma distribuição de probabilidade. Assim sendo, fixada uma distribuição D, desejamos encontrar uma regra de decisão $\hat{h}$ que não difira muito do risco mínimo de $\mathcal{H}$. Formalmente, definindo

$L_D^* = \min_{h \in \mathcal{H}} L_D(h)$, pode-se controlar a proximidade ao valor mínimo de risco através de $|L_D(\hat{h}) - L_D^*| \leq \epsilon$, para algum $\epsilon > 0$ pequeno, utilizando a métrica usual $d(x, y) = |x - y|$, ou utilizando-se outra métrica.

Uma vez estipulada de qual maneira será avaliada a proximidade ao valor ótimo, é necessário ainda especificar um critério para medir o sucesso de um algoritmo que seja baseado nesta formulação de aprendizado. Para tanto, utilizando a formulação apresentada, note que é possível interpretar a estrutura de um algoritmo de aprendizado como uma função de todas amostras que podem ser obtidas de $Z = \mathcal{X} \times \mathcal{Y}$ para a classe $\mathcal{H}$, ou seja $\mathcal{A} : \bigcup_{m \geq 1} Z^m \to \mathcal{H}$ de forma que, fixada uma distribuição $D \in \mathcal{D}$ utilizada para gerar as amostras, ao escolher uma decisão h o algoritmo sofre uma penalidade (arrependimento) $L(D, h)$ por não ter escolhido uma decisão ótima. Desta maneira, é definida a função de arrependimento $L : \mathcal{D} \times \mathcal{H} \to \mathbb{R}_+$ que pode ser obtida através de uma função de perda l e que mede o quanto o algoritmo $\mathcal{A}$ falhou em retornar uma decisão h próxima à decisão ótima, assumindo que a distribuição D é utilizada para obter os exemplos de treino. Assim sendo, Haussler define o objetivo do aprendizado como sendo a minimização do valor médio do arrependimento sofrido por um algoritmo $\mathcal{A}$ sobre todas as possíveis amostras de treino $z \in Z^m$ de tamanho m obtidas de acordo com a distribuição de probabilidade $D \in \mathcal{D}$ utilizada para gerá-las, isto é a minimização de

$$R_{L,\mathcal{A},m}(D) = \int_{z \in Z^m} L(D, \mathcal{A}(z)) dD^m(z).$$

Esta formulação é utilizada para definir a noção de aprendizado PAC agnóstico.

Definição 3.14 (Aprendizado PAC agnóstico)**.** Uma classe de hipóteses $\mathcal{H}$ é dita PAC agnóstica aprendível se $\exists\, m_{\mathcal{H}} : (0, 1)^2 \to \mathbb{N}$ e um algoritmo de aprendizado com a seguinte propriedade: para cada $\epsilon, \delta \in (0, 1)$ e para cada distribuição D sobre $\mathcal{X} \times \mathcal{Y}$, quando o algoritmo é executado em $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ exemplos iid gerados por D, o algoritmo retorna uma hipótese h tal que, com probabilidade $1 - \delta$ sobre a escolha de exemplos

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \epsilon$$

Esta definição nos garante que se rodarmos o algoritmo em um número suficiente de dados de treino, a hipótese retornada pelo algoritmo não será muito pior que a melhor hipótese contida na classe selecionada.

A relação com o modelo de Haussler pode ser percebida ao definir a função de arrependimento utilizando-se a métrica usual

$$L_\epsilon(D, h) = \begin{cases} 1, & \text{se } |L_D(h) - L_D^*| > \epsilon \\ 0, & \text{caso contrário.} \end{cases}$$

Deste modo, $R_{L,\mathcal{A},m}(D)$ mede a probabilidade do algoritmo $\mathcal{A}$ retornar uma regra de decisão h cujo risco difira mais do que ϵ do valor ótimo ao ter como entrada uma amostra de tamanho m obtida de acordo com D. Finalmente, impondo que $R_{L,\mathcal{A},m}(D)$ seja menor do que um parâmetro de confiança $\delta > 0$ temos que, com probabilidade de pelo menos $1 - \delta$, a hipótese h retornada por $\mathcal{A}$ satisfaz $L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \epsilon$.

Exemplo 3.15. *Dada uma distribuição de probabilidade D sobre $\mathcal{X} \times \{0, 1\}$, a melhor função que prevê a classificação de $\mathcal{X}$ em $\{0, 1\}$ é*

$$f_D(x) = \begin{cases} 1, & \text{se } \mathbb{P}[y = 1|x] \geq 1/2 \\ 0, & \text{caso contrário.} \end{cases}$$

Esta função sempre terá risco zero. No entanto, não podemos utilizá-la por não termos acesso a distribuição D. □

3.6.1 Convergência uniforme

Fixando uma função de perda l utilizada para compor a função de arrependimento L_ϵ, a fim de resolver um problema de aprendizado, um algoritmo deve encontrar uma hipótese que possui grande probabilidade de seu risco estar próximo ao valor ótimo. Devido ao fato de, assim como no caso do aprendizado PAC, a distribuição utilizada para gerar uma amostra de treino ser desconhecida, também será necessário que no caso agnóstico seja introduzida a definição de risco empírico, como apresentado em 3.13. Desta forma, o problema pode ser repensado como um problema de otimização, onde busca-se encontrar uma hipótese que minimize a diferença entre seu risco empírico e o menor risco empírico que pode ser obtido na classe de hipóteses, ou seja, o princípio de minimização do risco empírico com viés indutivo ainda é utilizado.

Embora resolver o problema de otimização não necessariamente resolva o problema de aprendizado, sabemos que o risco empírico é uma boa estimativa para o risco, uma vez que, assumindo que a função de perda é limitada, a Lei dos Grandes Números nos garante que, para cada $h \in \mathcal{H}$, quando $m \to \infty$, $L_S(h) \to L_D(h)$ com probabilidade 1. Portanto, nesta seção será introduzida a definição de convergência uniforme que juntamente com a desigualdade de Hoeffding será utilizada para demonstrar que toda classe de hipóteses finita é aprendível em relação ao modelo PAC agnóstico.

Definição 3.16. Um conjunto de treino S é dito ϵ-representativo em relação ao domínio $\mathcal{Z}$, a classe de hipóteses $\mathcal{H}$, a função de perda l e a distribuição D se

$$\forall\, h \in \mathcal{H},\ |L_S(h) - L_D(h)| \leq \epsilon.$$

Conforme discutido acima, a diferença $|L_S(h) - L_D(h)|$ é a maneira mais natural de medir a eficiência de um hipótese. O resultado a seguir motiva sua definição.

Lema 3.17. *Assuma que S é $\frac{\epsilon}{2}$-representativo em relação ao domínio Z, a classe de hipóteses $\mathcal{H}$, a função de perda l e a distribuição D. Então, qualquer retorno de $ERM_{\mathcal{H}}(s)$, i.e qualquer $h_S \in argmin_{h\in\mathcal{H}} L_S(h)$, satisfaz*

$$L_D(h_S) \leq \min_{h\in\mathcal{H}} L_D(h) + \epsilon.$$

Demonstração. $\forall\ h \in \mathcal{H}$,

$$\begin{aligned} L_D(h_S) &\leq L_S(h_S) + \frac{\epsilon}{2} \\ &\leq L_S(h) + \frac{\epsilon}{2} \\ &\leq L_D(h) + \frac{\epsilon}{2} + \frac{\epsilon}{2} \\ &= L_D(h) + \epsilon. \end{aligned}$$

Onde a primeira e a terceira desigualdades seguem da hipótese de S ser $\epsilon/2$-representativa e a segunda desigualdade pelo fato de h_S minimizar o risco empírico. ■

O lema implica que para garantir que a regra ERM é PAC agnóstica aprendível, é suficiente mostrar que com probabilidade de pelo menos $1-\delta$ sobre a escolha dos exemplos, ela será ϵ-representativa. A partir disto, definimos a convergência uniforme.

Definição 3.18 (Convergência uniforme)**.** Dizemos que uma classe de hipóteses $\mathcal{H}$ possui a propriedade de convergência uniforme, em relação a um domínio Z e a uma função de perda l, se existe uma função $m^{UC}_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ tal que para cada $\epsilon, \delta \in (0,1)$ e para cada distribuição de probabilidade D sobre Z, se S é uma amostra de $m \geq m^{UC}_{\mathcal{H}}(\epsilon, \delta)$ exemplos obtidos i.i.d de acordo com D, então, com probabilidade de pelo menos $1-\delta$, S é ϵ-representativa.

O termo uniforme se refere ao fato de termos um tamanho de amostra m fixado que funcione para todos os elementos de $\mathcal{H}$ e sobre todas as possíveis distribuições de probabilidade sobre o domínio.

Da definição de convergência uniforme e do lema anterior, segue:

Corolário 3.19. *Se a classe $\mathcal{H}$ possui a propriedade de convergência uniforme com função $m^{UC}_{\mathcal{H}}$, então a classe é PAC agnóstica aprendível com complexidade de amostra $m_{\mathcal{H}}(\epsilon, \delta) \leq m^{UC}_{\mathcal{H}}(\frac{\epsilon}{2}, \delta)$. Ademais, neste caso, o paradigma $ERM_{\mathcal{H}}$ é um algoritmo de aprendizagem PAC agnóstico bem sucedido para $\mathcal{H}$.*

Assim, podemos mostrar que classes finitas são PAC Agnósticas aprendíveis.

Proposição 3.20. *Seja $\mathcal{H}$ uma classe de hipóteses finita, Z um domínio e $l : \mathcal{H} \times Z \to [0,1]$ uma função de perda. Então, $\mathcal{H}$ possui a propriedade de convergência uniforme com complexidade de amostra $m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil$.*

Ademais, a classe é PAC agnóstica aprendível usando o algoritmo ERM com complexidade de amostra

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}\left(\frac{\epsilon}{2}, \delta\right) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil.$$

Demonstração. O resultado seguirá se mostrarmos a convergência uniforme. Assim, fixados ϵ e δ, precisamos encontrar um tamanho de amostra m que garanta que, para qualquer distribuição D, com probabilidade de pelo menos $1-\delta$ sobre a obtenção de $S = (z_i, ..., z_m)$ de maneira independente e identicamente distribuída através de D, temos que $\forall h \in \mathcal{H},\ |L_S(h) - L_D(h)| \leq \epsilon$. Isto é,

$$D^m(S\ :\ \forall\, h \in \mathcal{H},\ |L_S(h) - L_D(h)| \leq \epsilon) \geq 1 - \delta.$$

Equivalentemente, queremos mostrar que

$$D^m(S\ :\ \exists\, h \in \mathcal{H}\ com\ |L_S(h) - L_D(h)| > \epsilon) < \delta.$$

Escrevendo

$$\{S\ :\ \exists\, h \in \mathcal{H}\ com\ |L_S(h) - L_D(h)| > \epsilon\} = \bigcup_{h \in \mathcal{H}} \{S\ : |L_S(h) - L_D(h)| > \epsilon\}$$

e aplicando a cota da união, obtemos

$$D^m(\{S\ :\ \exists\, h \in \mathcal{H}\ \text{com}\ |L_S(h) - L_D(h)| > \epsilon\}) \leq \sum_{h \in \mathcal{H}} D^m(\{S\ : |L_S(h) - L_D(h)| > \epsilon\}. \tag{3.7}$$

Como cada z_i é obtido de forma independente e identicamente distribuída de acordo com D, $L_D(h) = \mathbb{E}_{Z \sim D}[l(h, Z_i)]$. Além disso, $L_S(h) = (1/m)\sum_{i=1}^m l(h, z_i)$ logo, pela linearidade da esperança, $\mathbb{E}[L_S(h)] = L_D(h)$. Assim, aplicando a desigualdade de Hoeffiding, obtemos

$$D^m(\{S\ : |L_S(h) - L_D(h)| > \epsilon\}) = \mathbb{P}[|L_S(h) - L_D(h)| > \epsilon] \leq 2e^{-2m\epsilon^2}. \tag{3.8}$$

Aplicando (3.8) em (3.7), segue

$$D^m(S\ :\ \exists\, h \in \mathcal{H}\ com\ |L_S(h) - L_D(h)| > \epsilon) \leq \sum_{h \in \mathcal{H}} 2e^{-2m\epsilon^2} = 2|\mathcal{H}|e^{-2m\epsilon^2}.$$

Portanto, se escolhermos $m \geq \frac{\log(\frac{2|\mathcal{H}|}{\delta})}{2\epsilon^2}$, temos

$$D^m(S\ :\ \exists\, h \in \mathcal{H}\ com\ |L_S(h) - L_D(h)| > \epsilon) \leq \delta.$$

■

Resolvendo a complexidade de amostra obtida para ϵ temos que, com probabilidade de pelo menos $1 - \delta$,

$$L_D(h) \leq L_S(h) + \sqrt{\frac{log|\,\mathcal{H}\,| + log\frac{2}{\delta}}{2m}}.$$

Além das observações já feitas no final da Seção 3.4.1, note que para aprender uma mesma classe de hipóteses com mesma acurácia ϵ, no aprendizado PAC agnóstico precisamos de uma maior complexidade de amostra do que no aprendizado PAC não agnóstico. Isto se deve à raiz quadrada presente nesta cota, que resulta do relaxamento da noção de aprendizado, isto é, da não satisfação da hipótese de consistência.

Assim sendo, fica evidente que quanto mais rica for uma classe de hipóteses, no sentido de possuir mais elementos, maior deverá ser o número de dados de treino disponíveis para aprendê-la. Entretanto, uma maior classe de hipóteses pode ajudar a diminuir o risco empírico, do qual esta cota é dependente. Esta troca entre reduzir o risco empírico versus controlar o número de dados de treino disponível será um dos assuntos do próximo capítulo.

4 Não existe almoço grátis

Conforme discutido anteriormente, um algoritmo de aprendizado $\mathcal{A}$, munido de uma classe de hipóteses $\mathcal{H}$ e um conjunto de dados de treino S, tem como objetivo selecionar uma hipótese $h_S \in \mathcal{H}$ que não só possua baixo risco empírico, mas também não tenha um risco elevado. Quando tal hipótese é encontrada, dizemos que o algoritmo possui capacidade de generalização, ou seja, sendo treinado sobre uma amostra do problema, ele é capaz de obter um desempenho satisfatório em pontos aos quais ele nunca teve acesso.

No entanto, a capacidade de generalização não é uma garantia. A teoria de aprendizado PAC nos mostra sob quais circunstâncias é esperado que isto aconteça. Mais especificamente, vimos que as cotas de aprendizado 3.8 e 3.20 apresentam uma dependência da classe de hipóteses escolhida e da amostra de treino disponível. Devido a esta dependência, também iremos nos referir às classes de hipóteses como modelos. Tal denominação evidencia o fato de que cada classe possui diferentes características.

Assim, fica evidente que a escolha e ajuste de um modelo é fundamental no processo de aprendizado. Este capítulo visa elucidar algumas questões a respeito desta escolha.

4.1 Superajuste e Subajuste

A primeira questão concerne à complexidade e à riqueza da classe de hipóteses escolhida. Neste contexto, o termo complexidade refere-se aos graus de liberdade das hipóteses pertencentes à classe selecionada. Por exemplo, funções polinomiais são mais complexas do que funções lineares. Já o termo riqueza refere-se ao tamanho, número de elementos, pertencentes a uma dada classe.

Embora as Proposições 3.8 e 3.20 pareçam favorecer classes de hipóteses com poucos elementos, é necessário garantir que o modelo escolhido é complexo o suficiente para ser capaz de explicar o problema, ou seja, não cometer muitos erros. Entretanto, se a classe de hipóteses for muito complexa, a hipótese selecionada pelo algoritmo pode apenas memorizar os dados de treino, fazendo com que seu risco seja demasiado elevado. Estas duas situações são denominadas subajuste (*underfitting*) e superajuste (*overfitting*).

Subajuste, ocorre quando o modelo não se adapta bem sequer aos dados com os quais ele foi treinado. Por exemplo, considere um problema de regressão em $\mathcal{X} = \mathbb{R}$ e um modelo utilizando funções lineares, isto é $\mathcal{H} = \{h_{a,b} \ : \ a, b \in \mathbb{R},\ h_{a,b} : \mathbb{R} \to \mathbb{R}\}$ onde $h_{a,b}(x) = ax + b$.

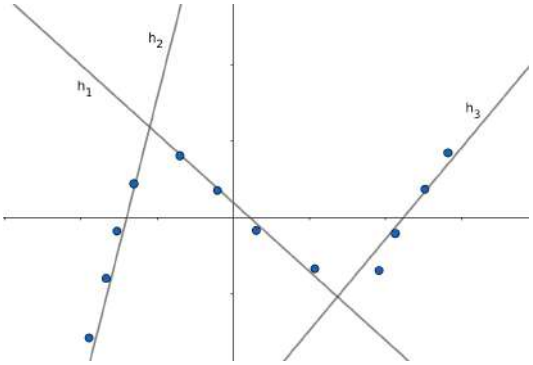

Figura 8 – Subajuste

Na Figura 8, temos a representação de uma amostra de treino. Ela nos sugere que, se os dados do problema não estiverem dispostos de maneira linear, não se pode esperar que mesmo a hipótese que minimiza o risco empírico possua de fato um baixo risco empírico. Neste caso, a escolha de um modelo mais flexível, com funções polinomiais por exemplo, seria mais adequado.

Superajuste, ocorre quando o modelo se adapta muito bem aos dados com os quais está sendo treinado, porém, não generaliza bem para novos dados, aos quais ele nunca teve acesso. O Exemplo 3.3 ilustra esta situação. Embora a hipótese tenha sempre risco empírico igual a zero, seu risco é elevado.

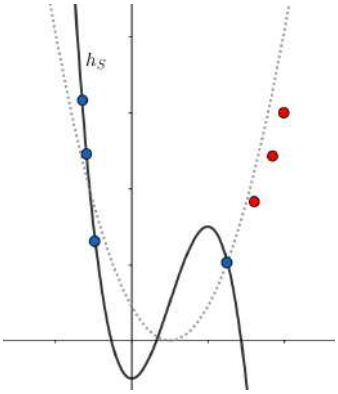

Figura 9 – Superajuste

Retornando ao problema de regressão unidimensional, a Figura 9 mostra os pontos de uma amostra, em azul, e pontos fora dela, em vermelho. A hipótese retornada pelo algoritmo, h_S, não comete nenhum erro sobre a amostra, no entanto, o algoritmo não possui boa capacidade de generalização. Neste caso, um polinômio de menor grau, e portanto com menos graus de liberdade, mesmo tendo um maior risco empírico, poderia se mostrar mais efetivo.

Uma formalização destas situações baseada na teoria da informação pode ser encontrada em [8].

4.2 A necessidade de escolher um modelo

A seção anterior discute a importância, e as consequências, da escolha de um modelo, ou seja, do ato de restringir a classe de hipóteses que será utilizada no problema. No entanto, podemos nos perguntar se tal escolha é realmente necessária, isto é, se existe um algoritmo de aprendizado que, munido com a classe de hipóteses contendo todas as funções existentes, é capaz de resolver qualquer problema de aprendizado.

O teorema que será demonstrado a seguir elucida que tal algoritmo não existe. Formalmente, ele nos diz que, para qualquer algoritmo de aprendizado, existe um problema em que ele falha, mesmo que tal problema possa ser resolvido por outro algoritmo.

Teorema 4.1 (Não existe almoço grátis)**.** *Seja $\mathcal{A}$ um algoritmo de aprendizado para a tarefa de classificação binária com função de perda l_{0-1} sobre um domínio $\mathcal{X}$. Seja m um número menor do que $|\mathcal{X}|/2$, representando o tamanho do conjunto de treino. Então, existe uma distribuição de probabilidade D sobre $\mathcal{X} \times \{0,1\}$ tal que:*

1. *existe uma função $f : \mathcal{X} \to \{0,1\}$ com $L_D(f) = 0$;*
2. *com probabilidade de pelo menos $1/7$ sobre a escolha de $S \sim D^m$, temos*

$$L_D(\mathcal{A}(S)) \geq 1/8.$$

Demonstração. Seja C um subconjunto de $\mathcal{X}$ com tamanho $2m$. Existem $T = 2^{2m}$ possíveis funções de C para $\{0,1\}$, que denotaremos por $f_1, ..., f_T$.

Para cada uma destas funções, seja D_i uma distribuição sobre $C \times \{0,1\}$ definida por

$$D_i(\{(x,y)\}) = \begin{cases} 1/|C|, & \text{se } y = f_i(x); \\ 0, & \text{caso contrário} \end{cases},$$

é fácil ver que $L_{D_i}(f_i) = 0$.

Vamos mostrar que, para todo algoritmo $\mathcal{A}$ que recebe um conjunto de dados de treino S, de tamanho m, e retorna uma função $\mathcal{A}(S) : C \to \{0,1\}$, vale que

$$\max_{i \in [T]} \mathop{\mathbb{E}}_{\mathcal{S} \sim D_i^m} [L_{D_i}(\mathcal{A}(\mathcal{S}))] \geq \frac{1}{4}. \tag{4.1}$$

Existem $k = (2m)^m$ sequências de C com m amostras, que denotaremos por $S_1, ..., S_k$. Dado $j \in [k]$, com $S_j = (x_1, ..., x_m)$ seja $S_j^i = ((x_1, f_i(x_1)), ..., (x_m, f_i(x_m)))$.

Fixado D_i, cada $S_1^i, ..., S_k^i$ que $\mathcal{A}$ pode receber possui a mesma probabilidade de ser obtido, logo

$$\mathop{\mathbb{E}}_{\mathcal{S} \sim D_i^m} [L_{D_i}(\mathcal{A}(\mathcal{S}))] = \frac{1}{k} \sum_{j=1}^{k} L_{D_i}(\mathcal{A}(S_j^i)). \tag{4.2}$$

Além disso,

$$\begin{aligned}\max_{i\in[T]} \frac{1}{k}\sum_{j=1}^{k} L_{D_i}(\mathcal{A}(S_j^i)) &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{k}\sum_{j=1}^{k} L_{D_i}(\mathcal{A}(S_j^i)) \\ &= \frac{1}{k}\sum_{j=1}^{k}\frac{1}{T}\sum_{i=1}^{T} L_{D_i}(\mathcal{A}(S_j^i)) \\ &\geq \min_{j\in[k]} \frac{1}{T}\sum_{i=1}^{T} L_{D_i}(\mathcal{A}(S_j^i)). \end{aligned} \tag{4.3}$$

Agora, fixe $j \in [k]$. Seja $v_1, ..., v_p$ os elementos de C que não estão em S_j, obviamente $p \geq m$. Assim, para cada função $h : C \to \{0, 1\}$ e para cada $i \in [T]$, temos

$$L_{D_i} = \frac{1}{2m}\sum_{x\in C} \mathbb{1}_{[h(x)\neq f_i(x)]} \geq \frac{1}{2m}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r)\neq f_i(v_r)]} \geq \frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r)\neq f_i(v_r)]}.$$

Logo,

$$\begin{aligned}\frac{1}{T}\sum_{i=1}^{T} L_{D_i}(\mathcal{A}(S_j^i)) &\geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[A(S_j^i)(v_r)\neq f_i(v_r)]} \\ &= \frac{1}{2p}\sum_{r=1}^{p}\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r)\neq f_i(v_r)]} \\ &\geq \frac{1}{2}.\min_{r\in[p]} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r)\neq f_i(v_r)]}. \end{aligned} \tag{4.4}$$

Agora, fixe $r \in [p]$. Podemos particionar as funções $f_1, ..., f_T$ em $T/2$ pares disjuntos, onde para um par $(f_i, f_{i'})$ teremos que, para cada $c \in C$, $f_i(c) \neq f_{i'}(c)$ se, e somente se, $c = v_r$. Como, para este par, devemos ter $S_j^i = S_j^{i'}$, segue que

$$\mathbb{1}_{[\mathcal{A}(S_j^i)(v_r)\neq f_i(v_r)]} + \mathbb{1}_{[\mathcal{A}(S_j^{i'})(v_r)\neq f_i(v_r)]} = 1$$

de modo que

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r)\neq f_i(v_r)]} = \frac{1}{2}. \tag{4.5}$$

Combinando (4.5), (4.4), (4.3) e (4.2), temos

$$\begin{aligned}\max_{i\in[T]} \mathop{\mathbb{E}}_{\mathcal{S}\sim D_i^m}[L_{D_i}(\mathcal{A}(\mathcal{S}))] &= \max_{i\in[T]} \frac{1}{k}\sum_{j=1}^{k} L_{D_i}(\mathcal{A}(S_j^i)) \\ &\geq \min_{j\in[k]} \frac{1}{T}\sum_{i=1}^{T} L_{D_i}(\mathcal{A}(S_j^i)) \\ &\geq \frac{1}{2}.\min_{r\in[p]} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r)\neq f_i(v_r)]} \\ &= \frac{1}{4}. \end{aligned}$$

Isto significa que para cada algoritmo $\mathcal{A}'$ que recebe um conjunto de treino S com m elementos, existe uma função $f : \mathcal{X} \to \{0, 1\}$ e uma distribuição D sobre $\mathcal{X} \times \{0, 1\}$ tal que $L_D(f) = 0$ e

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}[L_D(\mathcal{A}'(\mathcal{S}))] \geq 1/4$$

aplicando o Lema B.1, segue que

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D_i^m}[L_{D_i}(\mathcal{A}'(\mathcal{S})) \geq 1/8] \geq 1/7$$

■

A ideia usada na demonstração é a de que qualquer algoritmo de aprendizagem que observa apenas metade dos exemplos em C não tem informação sobre quais devem ser os rótulos das demais instâncias em C. Portanto, existe uma função alvo f que contradiz os rótulos que $A(S)$ prediz nos exemplos não observados em C.

Embora, num primeiro momento, o teorema anterior não pareça fatalista, como sua consequência, tem-se o seguinte resultado.

Corolário 4.2. *Seja $\mathcal{X}$ um domínio infinito e $\mathcal{H}$ o conjunto de todas as funções de $\mathcal{X}$ para $\{0,1\}$. Então, $\mathcal{H}$ não é PAC aprendível.*

Demonstração. Assuma, por contradição, que a classe é PAC aprendível.

Tome $\epsilon < 1/8$ *e* $\delta < 1/7$. Pela definição de aprendizado PAC, deve existir um algoritmo A e um inteiro $m = m(\epsilon, \delta)$, tal que para qualquer distribuição D sobre $\mathcal{X} \times \{0,1\}$, se para alguma função $f : \mathcal{X} \to \{0,1\}$ tivermos $L_D(f) = 0$, então com probabilidade maior do que $1-\delta$ quando A é aplicado sobre amostras S de tamanho m geradas i.i.d de acordo com D, temos $L_D(A(S)) \leq \epsilon$.

No entanto, aplicando o teorema, com $|\mathcal{X}| > 2m$, para cada algoritmo de aprendizado (em particular para A), existe uma distribuição D tal que com probabilidade maior do que $1/7 > \delta$, $L_D(A(S)) > 1/8 > \epsilon$. Contradição. ■

Portanto, conclui-se que, para cada problema de aprendizado que se deseja resolver, além de se possuir uma quantidade satisfatória de dados de treino, é necessário que se faça uma boa escolha do modelo que será utilizado. Desta maneira, quando submetido ao processo de treinamento, o algoritmo será capaz de encontrar parâmetros, referentes as hipóteses que constituem a classe escolhida que façam com que o risco e o risco empírico estejam, simultaneamente, dentro de uma margem estabelecida.

4.3 Viés indutivo versus complexidade

Naturalmente, segue a seguinte questão: como fazer uma boa escolha para a classe de hipóteses?

Uma das possíveis respostas pode ser encontrada na noção de risco. Dada uma hipótese $h_S \in \mathop{\mathrm{ERM}}_{\mathcal{H}}$, podemos escrever

$$L_D(h_S) = \epsilon_{APP} + \epsilon_{EST}$$

onde $\epsilon_{APP} = \min_{h \in \mathcal{H}} L_D(h)$ e $\epsilon_{EST} = L_D(h_S) - \epsilon_{APP}$. Sendo assim, o erro cometido por um algoritmo pode ser decomposto em duas categorias, com origens diferentes.

O erro de aproximação, ϵ_{APP}, é o risco mínimo alcançável por uma hipótese de uma classe de hipóteses. Desta maneira, mede o quanto de risco se tem devido a restrição da classe de hipóteses, ou seja, a quantidade de viés indutivo. Ele não depende do tamanho da amostra e é determinado pela classe de hipóteses escolhida, podendo ser diminuído ao aumentar-se sua complexidade.

Por outro lado, o erro de estimação, ϵ_{EST}, é a diferença entre o risco obtido pela hipótese que minimiza o risco empírico e o erro de aproximação. Ele resulta do fato do risco empírico ser apenas uma aproximação do risco verdadeiro. Sendo assim, depende do tamanho do conjunto de dados de treino e do tamanho da classe de hipóteses escolhida.

Assim, existe um troca, um conflito de escolha, entre o viés indutivo e a complexidade de $\mathcal{H}$. Escolhendo $\mathcal{H}$ com uma alta complexidade, diminui-se o erro de aproximação, mas ao mesmo tempo pode haver um aumento no erro de estimação, causando superajuste. Em contrapartida, escolhendo-se $\mathcal{H}$ com poucos elementos, reduz-se o erro de estimação, mas o erro de aproximação pode aumentar, causando subajuste.

A única maneira de contornar esta situação é ser capaz de obter conhecimento prévio acerca do problema a ser resolvido. Este conhecimento prévio é utilizado para selecionar uma classe de hipóteses que melhor se adapte ao problema apresentado. Em outras palavras, concluímos que diferentes modelos possuem suas vantagens e desvantagens, de modo que não existe uma resposta única e universal para a questão da seleção do modelo.

5 Dimensão VC

Até o momento, só foram consideradas classes de hipóteses finitas. No entanto, na maioria das aplicações realizadas no mundo real, são utilizadas classes de hipóteses infinitas.

Neste capítulo, será apresentado o conceito de dimensão VC, uma propriedade intrínseca a cada classe de hipóteses, que é utilizado para verificar se uma dada classe é ou não aprendível no modelo PAC. O termo VC faz referência a Vapinik e Chernoviks, os matemáticos que a desenvolveram.

Primeiramente, consideremos o exemplo a seguir.

Exemplo 5.1. *Seja* $\mathcal{H} = \{h_a \ : \ a \in \mathbb{R}\}$ *onde* $h_a : \mathbb{R} \to \{0,1\}$ *é dada por* $h_a(x) = \mathbb{1}_{[x<a]}$. *Embora* $\mathcal{H}$ *seja infinita, podemos mostrar que ela é PAC aprendível, usando a regra ERM, com complexidade de amostra*

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(2/\delta)}{\epsilon} \right\rceil.$$

Para isto, sejam $a^* \in \mathbb{R}$ *tal que* $h_{a^*}(x) = \mathbb{1}_{[x<a^*]}$ *possui* $L_D(h_{a^*}) = 0$[1], D_x *a distribuição marginal sobre* $\mathcal{X}$ *e* $a_0 < a^* < a_1$ *tais que*

$$\mathop{\mathbb{P}}_{X \sim D_x}[X \in (a_0, a^*)] = \mathop{\mathbb{P}}_{X \sim D_x}[X \in (a^*, a_1)] = \epsilon.$$

Se $D_x(-\infty, a^*) \leq \epsilon$ *fazemos* $a_0 = -\infty$, *similarmente para* a_1.

Dado um conjunto de treino S, *seja* $b_0 = \max\{x \ : \ (x,1) \in S\}$ *e* $b_1 = min\{x \ : \ (x,0) \in S\}$. *Se nenhum exemplo em* S *é positivo fazemos* $b_0 = -\infty$, *e se nenhum exemplo é negativo fazemos* $b_1 = \infty$.

Dado $b_S \in \mathbb{R}$ *associado à hipótese ERM* h_S, *temos* $b_S \in (b_0, b_1)$. *Logo, uma condição suficiente para que* $L_D(h_S) \leq \epsilon$ *é que* $b_0 \geq a_0$ *e* $b_1 \leq a_1$, *em outras palavras*

$$\mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[L_D(h_S) > \epsilon] \leq \mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[b_0 < a_0 \vee b_1 > a_1].$$

Mas, pela cota da união

$$\mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[L_D(h_S) > \epsilon] \leq \mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[b_0 < a_0] + \mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[b_1 > a_1]. \tag{5.1}$$

O evento $b_0 < a_0$ *acontece se, e somente se, todas as amostras em* S *não estão no intervalo* (a_0, a^*), *cuja massa de probabilidade definimos para ser* ϵ, *ou seja*

$$\mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[b_0 < a_0] = \mathop{\mathbb{P}}_{\mathcal{S} \sim D^m}[\forall (x,y) \in S, \ x \notin (a_0, a^*)] = (1-\epsilon)^m \leq e^{-\epsilon m}.$$

[1] Como queremos mostrar que a classe é aprendível segundo o paradigma PAC, e não PAC agnóstico, estamos assumindo que a hipótese de consistência é satisfeita.

Como assumimos que $m > \frac{log(2/\delta)}{\epsilon}$*, segue que a probabilidade é no máximo* $\delta/2$*. De modo análogo, é fácil ver que* $\mathbb{P}_{\mathcal{S}\sim D^m}[b_1 > a_1] \leq \delta/2$*.*

Assim, aplicando estas desigualdade em (5.1)*, segue que*

$$\mathbb{P}_{\mathcal{S}\sim D^m}[L_D(h_S) > \epsilon] \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta.$$

Portanto, $\mathcal{H} = \{h_a \ : \ a \in \mathbb{R}\}$ *é PAC aprendível.* □

O exemplo anterior nos mostra que, de fato, $\mathcal{H}$ ser finita é uma condição suficiente, mas não necessária para garantir o aprendizado.

5.1 Restrição e fragmentação

Antes do conceito de dimensão VC ser introduzido, precisamos de algumas definições que o fundamentam.

Definição 5.2 (Restrição de $\mathcal{H}$ para C)**.** Seja $\mathcal{H}$ uma classe de funções de $\mathcal{X}$ para $\{0,1\}$ e $C = \{c_1, ..., c_m\} \subset \mathcal{X}$. A restrição de $\mathcal{H}$ para C é o conjunto das funções de C para $\{0,1\}$ que podem ser derivadas de $\mathcal{H}$, isto é,

$$\mathcal{H}_C = \{(h(c_1), ..., h(c_m)) \ : \ h \in \mathcal{H}\}.$$

Assim, representamos cada função de C para $\{0,1\}$ como um vetor em $\{0,1\}^{|C|}$

Como veremos, as restrições da classe de hipóteses em relação aos subconjuntos do domínio do problema irão nos indicar se tal classe é capaz de resolver o problema de aprendizado com o domínio considerado. A definição a seguir corrobora com essa visão.

Definição 5.3. Uma classe de hipóteses $\mathcal{H}$ fragmenta um conjunto finito $C \subset \mathcal{X}$ se a restrição de $\mathcal{H}$ para C é o conjunto de todas as funções de C para $\{0,1\}$, isto é, $|\mathcal{H}_C| = 2^{|C|}$.

Exemplo 5.4. *Seja* $\mathcal{H} = \{h_a \ : \ a \in \mathbb{R}\}$ *onde* $h_a : \mathbb{R} \to \{0,1\}$ *é dada por* $h_a(x) = \mathbb{1}_{[x<a]}$*. Tome* $C = \{c_1\}$*.*

Fazendo $a = c_1 + 1$ *temos* $h_a(c_1) = 1$*, e tomando* $a = c_1 - 1$ *temos* $h_a(c_1) = 0$*. Portanto,* $\mathcal{H}_C$ *é o conjunto de todas as funções de C para* $\{0,1\}$*, e* $\mathcal{H}$ *fragmenta C.*

Porém, dado $C = \{c_1, c_2\}$*, com* $c_1 \leq c_2$*, nenhum* $h \in \mathcal{H}$ *é capaz de retornar* $(0,1)$*, pois qualquer* $a \in \mathbb{R}$ *que atribua 0 a* c_1 *deverá atribuir 0 a* c_2*. Portanto,* $\mathcal{H}$ *não fragmenta C.* □

Sabemos que $\mathcal{H} = \{h_a \ : \ a \in \mathbb{R}\}$, mesmo sendo infinita, é PAC aprendível. O ponto fundamental é que $\mathcal{H}$ não fragmenta subconjuntos com mais de um elemento.

Voltando à demonstração do Teorema 4.1, vemos que, se qualquer subconjunto $C \subset \mathcal{X}$ é fragmentado por $\mathcal{H}$, podemos construir uma distribuição sobre C baseada em qualquer função de C para $\{0, 1\}$, enquanto ainda mantemos a hipótese de consistência. A possibilidade de obtenção desta distribuição é o que impede que a classe seja PAC aprendível. Mais explicitamente, o teorema não existe almoço grátis possui o seguinte corolário:

Corolário 5.5. *Seja $\mathcal{H}$ uma classe de hipóteses de um domínio $\mathcal{X}$ para $\{0, 1\}$ e seja m o tamanho da amostra de treino. Assuma que existe um conjunto $C \subset \mathcal{X}$ de tamanho $2m$ que é fragmentado por $\mathcal{H}$. Então, para qualquer algoritmo de aprendizado A, existe uma distribuição D sobre $\mathcal{X} \times \{0, 1\}$ e uma hipótese $h \in \mathcal{H}$ tal que $L_D(h) = 0$ mas*

$$\mathop{\mathbb{P}}_{S \sim D^m}[L_D(A(S)) \geq 1/8] \geq 1/7.$$

O resultado anterior nos diz que, se $\mathcal{H}$ fragmenta algum conjunto C de tamanho 2m, então não podemos aprender $\mathcal{H}$ com uma amostra de tamanho m. Portanto, vemos que, para uma classe de hipóteses infinita ser PAC aprendível, ela não pode fragmentar subconjuntos infinitos[2]do domínio do problema.

5.2 Dimensão VC

A dimensão VC é utilizada para caracterizar classes de hipóteses infinitas, ao invés da complexidade de amostra. A partir do que foi apresentado na seção anterior, sabemos que ela é dada em função das restrições da classe de hipóteses com relação aos subconjuntos do domínio do problema considerado. Formalmente:

Definição 5.6 (Dimensão VC)**.** A dimensão VC de uma classe de hipóteses $\mathcal{H}$, $VCdim(\mathcal{H})$, é o tamanho máximo de um conjunto $C \subset \mathcal{X}$ que pode ser fragmentado por $\mathcal{H}$.

Se $\mathcal{H}$ pode fragmentar conjuntos de tamanho arbitrariamente grande, dizemos que $\mathcal{H}$ possui dimensão VC infinita.

Pela definição anterior, para mostrar que $VCdim(\mathcal{H}) = d$, precisamos mostrar que

1. existe um conjunto C de tamanho d que é fragmentado por $\mathcal{H}$;
2. todo conjunto C de tamanho d+1 não pode ser fragmentado por $\mathcal{H}$.

Vejamos um exemplo.

[2] No sentido de $\mathcal{H}$ ser capaz fragmentar conjuntos de tamanho arbitrariamente grande.

Exemplo 5.7. *Seja* $\mathcal{H} = \{h_{a,b} \ : \ a, b \in \mathbb{R}, \ a < b\}$ *onde* $h_{a,b} : \mathbb{R} \to \{0, 1\}$ *é dada por*

$$h_{a,b}(x) = \mathbb{1}_{[x \in (a,b)]}.$$

Dado $C = \{1, 2\}$*, é fácil ver que* $\mathcal{H}_C = \{(1, 1), (0, 0), (1, 0), (0, 1)\}$*, de modo que* $VCdim(\mathcal{H}) \geq 2$*.*

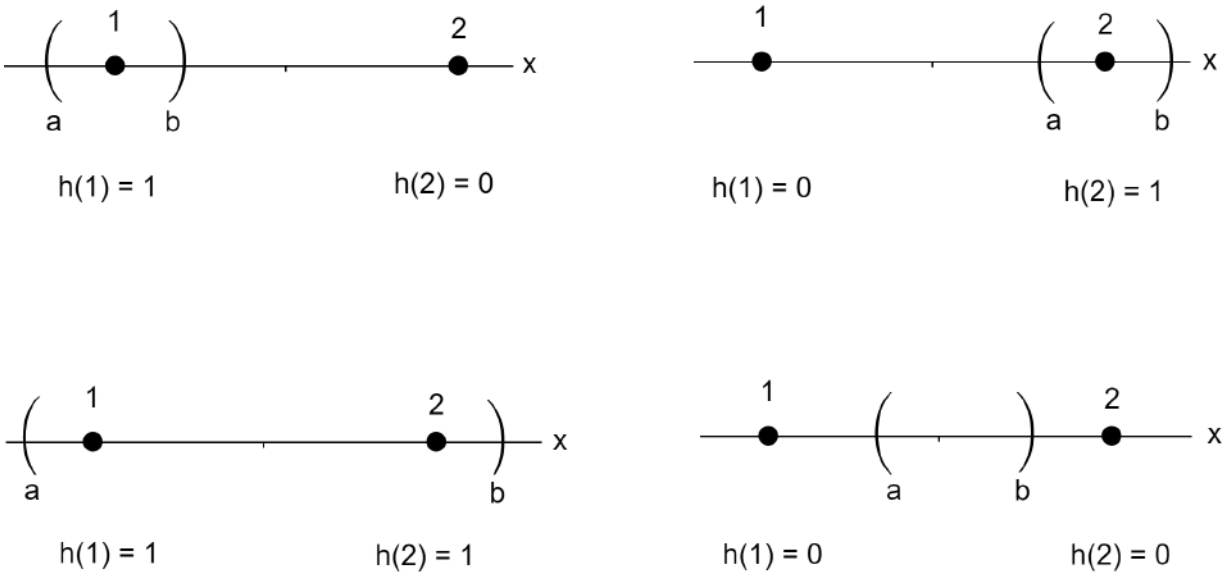

Figura 10 – C = {1,2} sendo fragmentado por $\mathcal{H}$, conforme definida no Exemplo 5.7.

No entanto, dado $C = \{c_1, c_2, c_3\}$ *arbitrário, assumindo, sem perda de generalidade, que* $c_1 \leq c_2 \leq c_3$ *não existe nenhuma função em* $\mathcal{H}$ *restrita a* C *que corresponda a* $(1, 0, 1)$*. Logo* $\mathcal{H}$ *não fragmenta* C*, e, portanto,* $VCdim(\mathcal{H}) = 2$*.* □

Exemplo 5.8. *Seja* $\mathcal{H}$ *uma classe de hipóteses finita. É fácil ver que, para qualquer conjunto* C*,* $|\mathcal{H}_C| \leq |\mathcal{H}|$*. Deste modo,* C *não pode ser fragmentado por* $\mathcal{H}$ *se* $|\mathcal{H}| < 2^{|C|}$*, ou seja,* $VCdim(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$*.*

Seguindo nossa argumentação, o próximo passo é mostrar que a dimensão VC ser finita é uma condição necessária para o aprendizado.

Teorema 5.9. *Seja* $\mathcal{H}$ *uma classe de dimensão VC infinita, então* $\mathcal{H}$ *não é PAC aprendível.*

Demonstração. Se a dimensão VC de $\mathcal{H}$ é infinita, para todo $m \in \mathbb{N}$, existe um subconjunto $C \subset \mathcal{X}$ de tamanho $2m$ que é fragmentado por $\mathcal{H}$. Do Corolário 5.5, segue que não existe uma complexidade de amostra que faça com que $\mathcal{H}$ seja PAC aprendível. ■

Embora a dimensão VC de uma classe de hipóteses ser finita garanta que a classe é PAC aprendível, este resultado é mais difícil de ser demonstrado. Este assunto será abordado na próxima seção.

5.3 Função de crescimento e lema de Sauer

Introduzida por Vapinik e Chernoviks (ver [9]), a função de crescimento é uma maneira de medir a complexidade de uma classe de hipóteses em função de suas restrições aos subconjuntos $C \subset \mathcal{X}$.

Definição 5.10. Seja $\mathcal{H}$ uma classe de hipóteses. Então a função de crescimento de $\mathcal{H}$, $\tau_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$, é definida como

$$\tau_{\mathcal{H}}(m) = \max_{C \subset \mathcal{X}:|C|=m} |\mathcal{H}_C|.^{3}$$

Em outras palavras, $\tau_{\mathcal{H}}(m)$ é o número de funções distintas de um conjunto C de tamanho m para $\{0, 1\}$ que podem ser obtidas restringindo $\mathcal{H}$ a C.

Note que, se $VCdim(\mathcal{H}) =$ d, então para qualquer $m \leq d$ temos $\tau_{\mathcal{H}}(m) = 2^m$. Nestes casos, $\mathcal{H}$ induz todas as funções de C para $\{0, 1\}$. O resultado a seguir nos fornece uma cota para a função de crescimento em função da dimensão VC.

Lema 5.11 (Sauer-Shelah-Perles)**.** *Seja $\mathcal{H}$ uma classe de hipóteses com $VCdim(\mathcal{H}) = d < \infty$. Então, para todo m*

$$\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}.$$

Em particular, se $m > d + 1$ então $\tau_{\mathcal{H}}(m) \leq (em/d)^d$.

Como podemos obter uma cota para a função de crescimento que depende da dimensão VC, o caminho natural para mostrar que a dimensão VC ser finita implica que a classe de hipóteses é PAC aprendível é utilizar esta nova medida de complexidade.

Uma outro resultado interessante que utiliza a função de crescimento é apresentado no teorema a seguir.

Teorema 5.12. *Seja $\mathcal{H}$ uma classe e $\tau_{\mathcal{H}}$ sua função de crescimento. Então, para cada distribuição D e $\delta \in (0, 1)$, com probabilidade de pelo menos $1 - \delta$ sobre $S \sim D^m$, temos*

$$|L_D(h) - L_S(h)| \leq \frac{4 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\delta\sqrt{2m}}.$$

Estes dois resultados, cujas demostrações se encontram no apêndice, serão utilizados para demonstrar a proposição desejada. No entanto, já é possível perceber que o termo $|L_D(h) - L_S(h)|$ indica uma relação com a propriedade de convergência uniforme.

[3] Estamos assumindo que $\tau_{\mathcal{H}}(m)$ é uma função mensurável.

Proposição 5.13. *Se $dimVC(\mathcal{H}) = d < \infty$, então $\mathcal{H}$ é PAC agnóstica aprendível.*

Demonstração. Devido ao que foi discutido no Capítulo 2, é suficiente mostrar que se a dimensão VC é finita, então a propriedade de convergência uniforme é garantida. Assim, vamos mostrar que

$$m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq 4\frac{16d}{(\delta\epsilon)^2}\log\left(\frac{16d}{(\delta\epsilon)^2}\right) + \frac{16d\log(2e/d)}{(\delta\epsilon)^2}.$$

Pelo lema de Sauer temos que, para $m > d$, $\tau_{\mathcal{H}}(2m) \leq (2em/d)^d$. Combinando isto com o Teorema 5.12, obtemos que, com probabilidade de pelo menos $1 - \delta$,

$$|L_D(h) - L_S(h)| \leq \frac{4 + \sqrt{d\log(2em/d)}}{\delta\sqrt{2m}}.$$

Como, em geral, temos um grande número de dados de treino, podemos assumir que $\sqrt{d\log(2em/d)} \geq 4$, de forma que

$$|L_D(h) - L_S(h)| \leq \frac{1}{\delta}\sqrt{\frac{2d\log(2em/d)}{m}}.$$

Para garantir que isto será no máximo ϵ, precisamos que

$$m \geq \frac{2d\log(m)}{(\delta\epsilon)^2} + \frac{2d\log(2e/d)}{(\delta\epsilon)^2}.$$

Utilizando o Lema B.6, segue que

$$m \geq 4\frac{2d}{(\delta\epsilon)^2}\log\left(\frac{2d}{(\delta\epsilon)^2}\right) + \frac{4d\log(2e/d)}{(\delta\epsilon)^2}.$$

■

Assim, mostramos que a dimensão VC é, de fato, uma maneira de caracterizar se uma classe de hipóteses é ou não PAC aprendível.

5.4 O teorema fundamental do aprendizado PAC

Como consequência de tudo que foi apresentado nos últimos capítulos, temos um dos resultados mais importantes da teoria de aprendizado. Ele sintetiza a relação dos diferentes conceitos que foram apresentados até agora.

Teorema 5.14 (Teorema fundamental do aprendizado PAC)**.** *Seja $\mathcal{H}$ uma classe de funções de um domínio $\mathcal{X}$ para $\{0, 1\}$ e considere a função de perda l_{0-1}. Então, são equivalentes*

1. *$\mathcal{H}$ possui a propriedade de convergência uniforme.*

2. *qualquer hipótese ERM é um algoritmo PAC agnóstico para* $\mathcal{H}$.

3. $\mathcal{H}$ *é PAC agnóstico aprendível.*

4. $\mathcal{H}$ *é PAC aprendível.*

5. *qualquer hipótese ERM é um algoritmo PAC para* $\mathcal{H}$.

6. $\mathcal{H}$ *tem dimensão VC finita.*

A implicação $1 \to 2$ foi demonstrada na Proposição 3.17 enquanto que as implicações $2 \to 3$, $3 \to 4$ e $4 \to 5$ seguem diretamente das definições de aprendizado PAC e aprendizado PAC agnóstico, que por sua vez são baseadas no princípio de minimização do risco empírico. A implicação $5 \to 6$ segue do Teorema 5.9. Finalmente, a implicação $6 \to 1$ foi demonstrada na Proposição 5.13. Assim, temos uma visão geral da equivalência das diferentes noções de aprendizado que foram definidas.

5.5 Demais medidas de complexidade

Devido a definição de fragmentação apresentada, a dimensão VC só pode ser usada como medida de complexidade para classes de hipóteses que assumem valores em um contra domínio contendo apenas dois elementos. Sendo assim, outras medidas de complexidade precisam ser utilizadas ao serem abordados problemas de classificação compostos por mais de duas classes e problemas de regressão. Deste modo, apresentaremos brevemente algumas definições e resultados importantes, a fim de fazer uma extensão dos assuntos abordados até o momento. Para uma abordagem mais formal, consultar [10], [11] e [12].

Em problemas de classificação em que $\mathcal{Y} = \{y_1, ..., y_k\}$, diversas situações que ainda não foram consideradas podem ocorrer. Por exemplo, uma determinada classe pode possuir um maior número de elementos no conjunto de dados de treino, de modo que, ao ser baseado no princípio ERM, um algoritmo pode selecionar uma hipótese que atribua maior chance de uma amostra a qual ele nunca teve acesso pertencer a classe mais representativa, o que pode não ser verdade para elementos fora da amostra de treino. Assim, outras funções de perda devem ser utilizadas a fim de prevenir tais situações, de modo que outras definições de fragmentação se fazem necessárias. Levando isto em consideração, uma possibilidade para o caso consistente é redefinir

$$L_S(h) = \sum_{i=1}^{m} \left(\sum_{l=1}^{k} \mathbb{1}_{[h(x_i)]_l \neq [f(x_i)]_l} \right)$$

e

$$L_D(h) = \mathop{\mathbb{E}}_{X \sim D} \left[\sum_{l=1}^{k} \mathbb{1}_{[h(x)]_l \neq [f(x)]_l} \right].$$

Assim, a dimensão de Natarajan, definida abaixo, pode ser utilizada para caracterizar a classe de hipóteses.

Definição 5.15. Seja $\mathcal{H}$ uma classe de hipóteses de $\mathcal{X}$ para $\mathcal{Y}$. Dado $C \subseteq \mathcal{X}$, dizemos que $\mathcal{H}$ N-fragmenta C se existem $f_1, f_2 : C \to \mathcal{Y}$ tais que, para todo $x \in C$ se tenha $f_1(x) \neq f_2(x)$, e para todo $D \subseteq C$ exista uma hipótese $h \in \mathcal{H}$ tal que

$$\forall x \in D,\ h(x) = f_1(x) \text{ e } \forall x \in C \backslash D, h(x) = f_2(x).$$

A dimensão de Natarajan de uma classe de hipóteses $\mathcal{H}$, $dimN(\mathcal{H})$, é a máxima cardinalidade de um conjunto que pode ser N-fragmentado por $\mathcal{H}$.

A utilização da dimensão proposta por Natarajan segue de um teorema proposto em [13], que nos garante que, dada uma classe de hipóteses, exitem constantes C_1 e C_2 tais que

$$C_1 \left(\frac{dimN(\mathcal{H}) + \log(\frac{1}{\delta})}{\epsilon} \right) \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \left(\frac{dimN(\mathcal{H}) \cdot \log(|\,\mathcal{Y}\,|) \cdot \log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})}{\epsilon} \right)$$

onde C_1 e C_2 seguem de um teorema provado por Vapink para a dimensão VC. Assim, temos que a dimensão de Natarajan de uma classe de hipóteses ser finita é uma condição suficiente e necessária para garantir o aprendizado, no caso consistente, com acurácia ϵ e confiança δ.

Já em problemas de regressão, devido ao fato de o contra domínio do problema ser contínuo, a dimensão de Natarajan não pode ser utilizada. Isto acontece principalmente porque a reformulação proposta a pouco para o risco e o risco empírico, como vimos na Seção 3.5, não é compatível com a função de perda $l_{sq}(h, (x, y)) := (h(x) - y)^2$. Neste caso, precisamos de uma formulação para o conceito de fragmentação associado a função de perda, e não às hipóteses que compõem a classe de hipóteses considerada.

Dada uma classe de hipóteses $\mathcal{H}$, definimos $\mathcal{L} = \{(x, y) \to l(h(x), y) \ : \ h \in \mathcal{H}\}$ como a família das funções de perda associadas a $\mathcal{H}$. Sobre esta nova família definida, o conceito de pseudodimensão desenvolvido por Pollard é aplicado.

Definição 5.16. Seja $\mathcal{L}$ uma família de funções de $\mathcal{X}$ para $\mathbb{R}$. Um subconjunto $C = \{x_1, ..., x_m\} \subseteq \mathcal{X}$ é P-fragmentado por $\mathcal{L}$ se existem $t_1, ..., t_m \in \mathbb{R}$ tais que

$$\left| \left\{ \begin{bmatrix} sign(l(x_1) - t_1) \\ \vdots \\ sign(l(x_m) - t_m) \end{bmatrix} \ : \ l \in \mathcal{L} \right\} \right| \ = \ 2^m.$$

A pseudodimensão de uma classe de hipóteses $\mathcal{L}$, $dimP(\mathcal{L})$, é a máxima cardinalidade de um conjunto que pode ser P-fragmentado por $\mathcal{L}$.

Para esta nova dimensão, também existe um resultado que garante que ela ser finita é uma condição suficiente e necessária para o aprendizado, a saber:

Proposição 5.17. *Seja $\mathcal{H}$ uma família de funções reais e $\mathcal{L} = \{(x, y) \to l(h(x), y) \ : \ h \in \mathcal{H}\}$ a família de funções de perda associadas a $\mathcal{H}$. Se $Pdim(\mathcal{L}) = d < \infty$ e l é não negativa e limitada por $M > 0$ então, para todo $\delta > 0$, com probabilidade de pelo menos $1 - \delta$ sobre a escolha de $\mathcal{S} \sim D^m$, para todo $h \in \mathcal{H}$,*

$$L_D(h) \leq L_S(h) + M\sqrt{\frac{2d\log(\frac{em}{d})}{m}} + M\sqrt{\frac{\log\ \frac{1}{\delta}}{2m}}.$$

Embora cada uma destas medidas de dimensão apresente uma definição diferente para fragmentação, elas são uma generalização para a dimensão VC. Se $|\,\mathcal{Y}\,| = 2$, temos que $VCdim(\mathcal{H}) = Ndim(\mathcal{H})$ e, se $\mathcal{H}$ é composta por funções de tipo $(x, y) \to \{0, 1\}$, denotando com $\mathcal{L}$ a família de funções de perda associadas a $\mathcal{H}$, temos $dimVC(\mathcal{H}) = dimP(\mathcal{L})$. Sendo assim, ressaltamos que essas ferramentas são utilizadas para que possamos obter uma cota da forma

$$L_D(h) \leq L_S(h) + \phi(|\,\mathcal{H}\,|, m, \delta).$$

Uma outra alternativa é utilizar a complexidade de Rademacher que, diferente das medidas de dimensão apresentadas, é baseada na definição de $\epsilon-$representatividade e pode ser utilizada tanto em problemas de regressão como em problemas de classificação. Mais detalhes podem ser vistos em [14].

6 Preditores Lineares

A fim de exemplificar a teoria de aprendizado anteriormente exposta, neste capítulo, serão apresentados dois algoritmos baseados no princípio de minimização do risco empírico.

Embora tais algoritmos possam ser aplicados em problemas reais, devido a sua simplicidade, outros algoritmos podem se mostrar mais efetivos. Isto ocorre porque, mesmo que o princípio ERM garanta capacidade de generalização, sua implementação computacional nem sempre é eficiente, e as cotas de aprendizado que foram obtidas nos capítulos anteriores podem ser melhoradas. A despeito disto, outros paradigmas para a construção de algoritmos de aprendizado podem ser utilizados, como o princípio de minimização do risco estrutural. Para mais detalhes, ver [15].

6.1 Classes de hipóteses lineares

Primeiramente, definimos a família de funções que será utilizada para compor a classe de hipóteses sobre a qual os algoritmos serão baseados. Em outras palavras, definimos o modelo que será utilizado.

Definição 6.1. A classe de funções afins, em $\mathbb{R}^d$, é definida como

$$L_d = \{h_{\mathbf{w},b} \ : \ \mathbf{w} \in \mathbb{R}^d, \ b \in \mathbb{R}\}$$

onde cada $h_{\mathbf{w},b} : \mathbb{R}^d \to \mathbb{R}$ é dada por

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left(\sum_{i=1}^{d} w_i x_i \right) + b.$$

Também é conveniente utilizar a notação

$$L_d = \{\mathbf{x} \to \langle \mathbf{w}, \mathbf{x} \rangle + b \ : \ \mathbf{w} \in \mathbb{R}^d, \ b \in \mathbb{R}\}.$$

Além disto, pode-se incorporar o valor da constante b como uma coordenada de $\mathbf{w}$, fazendo

$$\mathbf{w}' = (b, w_1, ..., w_d) \in \mathbb{R}^{d+1} \quad \text{e} \quad \mathbf{x}' = (1, x_1, ..., x_d)$$

de modo que

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle.$$

Assim, segue que cada função afim em $\mathbb{R}^d$ pode ser reescrita como um função linear homogênea em $\mathbb{R}^{d+1}$.

Note que diferentes classes de hipóteses lineares podem ser obtidas através da composição de uma função $\phi : \mathbb{R} \to \mathcal{Y}$ com as funções de L_d. A partir disto, em tarefas de

classificação binária, podemos escolher ϕ como sendo a função sinal, e para problemas de regressão, onde $\mathcal{Y} = \mathbb{R}$, ϕ pode ser simplesmente a função identidade.

As hipóteses $h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ são hiperplanos em $\mathbb{R}^d$, de modo que a composição com a função sinal pode ser interpretada como uma divisão de $\mathbb{R}^d$ em duas regiões, uma acima do hiperplano, e o outra abaixo. Formalmente, temos a definição abaixo.

Definição 6.2. Considerando L_d no caso homogêneo, a classe de semiespaços é definida como

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

Para ilustrar esta classe de hipóteses geometricamente, consideremos o caso $d = 2$. Cada hipótese forma um hiperplano perpendicular ao vetor $\mathbf{w}$ e que intersecta o eixo vertical no ponto $(0, -b/w_2)$.

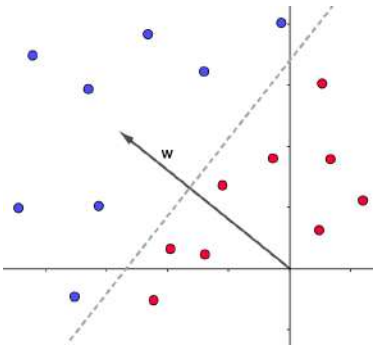

Figura 11 – Representação gráfica de classe de hiperplanos.

Os pontos em azul, que estão acima do hiperplano, são rotulados positivamente e os pontos em vermelho, que estão abaixo do hiperplano, são rotulados negativamente.

Devido ao fato desta classe possuir infinitos elementos, a fim de mostrar que ela é PAC aprendível, precisamos mostrar que ela possui dimensão VC finita.

Proposição 6.3. *A dimensão VC da classe de semiespaços, no caso homogêneo, em* $\mathbb{R}^d$ *é* d.

Demonstração. Primeiramente, considere o conjunto de vetores $\mathbf{e}_1, ..., \mathbf{e}_d$, onde para cada i o vetor $\mathbf{e}_i$ possui o valor 1 na i-ésima coordenada e zero em todas as outras. Este conjunto é fragmentado pela classe de semi espaços.

De fato, dados os rótulos correspondentes $y_1, ..., y_d$, defina $\mathbf{w} = (y_1, ..., y_d)$. Temos $\langle \mathbf{w}, \mathbf{e}_i \rangle = y_1$ para todo i.

Ademais, seja $\mathbf{x}_1, ..., \mathbf{x}_{d+1}$ um conjunto de d+1 vetores quaisquer em $\mathbb{R}^d$. Então, devem existir $a_1, ..., a_{d+1} \in \mathbb{R}$ escalares que não sejam todos iguais a zero, tais que $\sum_{i=1}^{d+1} a_i \mathbf{x}_i = 0$.

Sejam $I = \{i \ : \ a_i > 0\}$ e $J = \{j \ : \ a_j < 0\}$. Pelo menos um dos dois conjuntos deve ser não vazio.

Primeiramente, vamos assumir que os dois conjuntos são não vazios. Então

$$\sum_{i \in I} a_i \mathbf{x}_i = \sum_{j \in J} |a_j| \mathbf{x}_j$$

Agora, suponha que $\mathbf{x}_1, ..., \mathbf{x}_{d+1}$ é fragmentado pela classe homogênea. Então, deve existir um vetor $\mathbf{w}$ tal que $\langle \mathbf{w}, \mathbf{x}_i \rangle > 0$ para todo i, enquanto que $\langle \mathbf{w}, \mathbf{x}_j \rangle < 0$ para todo j. Isto implica que

$$0 < \sum_{i \in I} a_i \langle \mathbf{x}_i, \mathbf{w} \rangle = \left\langle \sum_{i \in I} a_i \mathbf{x}_i, \mathbf{w} \right\rangle = \left\langle \sum_{j \in J} |a_j| \mathbf{x}_j, \mathbf{w} \right\rangle = \sum_{j \in J} |a_j| \langle \mathbf{x}_j, \mathbf{w} \rangle < 0$$

o que é uma contradição.

E, se J (ou sem perda de generalidade, I) é vazio, então o termo mais a direita (esquerda) se torna uma igualdade, o que também é uma contradição.

■

A partir deste resultado, é fácil mostrar que a dimensão VC da classe não homogênea de semi espaços em $\mathbb{R}^d$ é d+1.

Em relação aos problemas de regressão e de classificação com mais de duas classes, em [13] e [11], é mostrado que a dimensão de Natarajan de HS_d e a pseudo dimensão de L_d são ambas finitas, o que mostra que estas classes de hipóteses, de fato, podem ser utilizadas para construir algoritmos de aprendizado a partir do principio de minimização do risco empírico.

6.2 Algoritmo Perceptron

O algoritmo perceptron de Rosemblatt definido em [16] é um algoritmo iterativo que constrói uma sequência de vetores $(\mathbf{w}^{(n)})_{n \in \mathbb{N}}$. Seu objetivo é encontrar um vetor $\mathbf{w}$, isto é uma hipótese pertencente a classe de semiespaços, capaz de realizar a correta classificação de uma amostra de treino.

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 $i \in [m]$, então note que

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

Logo, a regra de atualização guia a solução a minimizar o risco empírico a cada iteração. Um pseudocódigo é apresentado abaixo.

Algoritmo Perceptron

entrada: conjunto de treino $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$
inicialize: $\mathbf{w}^{(1)} = (0, ..., 0)$

for t = 1,2,...
    if $(\exists i \in [m] \ : \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$
        $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    else
        retorne $\mathbf{w}^{(t)}$

O teorema a seguir nos garante que, no caso consistente, o algoritmo classifica corretamente todos os dados de treino.

Teorema 6.4. *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* $B = \min\{\|\boldsymbol{w}\| \ : \ \forall i \in [m], \ y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq 1\}$[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* $\forall i \in [m], \ y_i \langle \boldsymbol{w}^{(t)}, \boldsymbol{x}_i \rangle > 0$.

Demonstração. Pela definição de condição de parada, se o algoritmo para, ele deve ter classificado corretamente todos os exemplos. Vamos mostrar que se o algoritmo roda por T iterações, então devemos ter $T \leq (RB)^2$.

Seja $\mathbf{w}^*$ o vetor que atinge o mínimo na definição de B, a ideia desta demostração é mostrar que após T iterações, o cosseno do ângulo entre $\mathbf{w}^*$ *e* $\mathbf{w}^{T+1}$ é pelo menos $\sqrt{T}/RB$. Ou seja, vamos mostrar que

$$\frac{\langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle}{\|\mathbf{w}^*\| \|\mathbf{w}^{(T+1)}\|} \geq \frac{\sqrt{T}}{RB}. \tag{6.1}$$

Pela desigualdade de Cauchy-Schwartz, o lado esquerdo da equação é no máximo 1. Logo, a equação (6.1) implicará que

$$1 \geq \frac{\sqrt{T}}{RB} \Rightarrow T \leq (RB)^2$$

o que concluirá a demostração. Para mostrar que a equação (6.1) é válida, primeiro mostraremos que

$$\langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle \geq T.$$

[1] Note que, como estamos supondo que hipótese de consistência é satisfeita, deve existir ao menos um vetor $\mathbf{w}$ que represente um hiperplano com risco, e portanto risco empírico, igual a zero, ou seja, que satisfaça $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1, \ \forall i \in [m]$. Desta maneira, B está bem definido.

De fato, é válido para $T = 0$, pois $\mathbf{w}^{(1)} = (0, .., 0)$ e portanto $\langle \mathbf{w}^*, \mathbf{w}^{(1)} \rangle = 0$. Na iteração t, se utilizarmos o exemplo $(\mathbf{x}_i, y_i)$, temos

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

Portanto, após $T > t$ iterações, temos

$$\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \sum_{t=1}^{T} (\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle - \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle) \geq T. \tag{6.2}$$

Agora, obtemos uma cota superior para $\|\mathbf{w}^{(T+1)}\|$. Para cada iteração t, temos que

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2 y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + y_i^2 \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R^2 \end{aligned} \tag{6.3}$$

onde a desigualdade vem do fato da amostra estar classificada incorretamente, isto é $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0$, e a norma de $\mathbf{x}_i$ ser no máximo R. Como $\|\mathbf{w}^{(1)}\|^2 = 0$, se usarmos a equação (6.3) de forma recursiva para T iterações, obtemos

$$\|\mathbf{w}^{(T+1)}\|^2 \leq TR^2 \Rightarrow \|\mathbf{w}^{(T+1)}\| \leq \sqrt{T} R. \tag{6.4}$$

Combinando as equações (6.2) e (6.4), e usando o fato de $\|\mathbf{w}^*\| = B$, temos

$$\frac{\langle \mathbf{w}^*, \mathbf{w}^{(T+1)} \rangle}{\|\mathbf{w}^*\| \|\mathbf{w}^{(T+1)}\|} \geq \frac{T}{B\sqrt{T}R} = \frac{\sqrt{T}}{RB}.$$

■

O perceptron é um algoritmo simples de ser implementado e sua convergência é garantida no caso consistente. No entanto, a taxa de convergência depende de B, que em algumas situações pode ser exponencialmente grande em d, fazendo com que ele não seja computacionalmente eficiente.

Nos casos em que a hipótese de consistência não é satisfeita, ou seja, em problemas onde não há a separabilidade linear dos dados, embora não tenhamos a garantia da convergência do algoritmo, esta ainda pode ocorrer. No entanto, nesta situação, havendo a convergência do algoritmo, o risco da hipótese retornada não será igual a zero.

6.3 Regressão linear

Idealizada por Francis Galton (ver [17]), a técnica de regressão linear visa prever um valor escalar, chamado de variável dependente, a partir de valores que acreditamos que ele está relacionado, as chamadas variáveis independentes. O termo linear se deve ao

fato de assumirmos que a relação entre a variável dependente e as variáveis independentes é dada por uma função linear. Neste caso, utilizaremos a classe de funções L_d, definidas em 6.1.

Sendo assim, utilizando o paradigma de minimização do risco empírico, dado um conjunto de treino S de tamanho m, queremos encontrar

$$\underset{\mathbf{w}}{\operatorname{argmin}} L_S(h_{\mathbf{w}}) = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle)^2.$$

Logo, queremos encontrar os vetores $\mathbf{w}$ que fazem com que o gradiente do risco empírico seja igual a zero. Ou seja, queremos resolver

$$\frac{2}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle) \cdot \mathbf{x}_i = 0.$$

No entanto, podemos reescrever este problema na forma $A\mathbf{w} = \mathbf{b}$ onde

$$A = \left(\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T \right) \text{ e } \mathbf{b} = \sum_{i=1}^{m} y_i \mathbf{x}_i.$$

Ou, na forma matricial

$$A = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix}^T$$

e

$$\mathbf{b} = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

Assim, se A é uma matriz invertível, a solução do problema ERM é dada por $\mathbf{w} = A^{-1}\mathbf{b}$. Caso contrário, outro método numérico pode ser utilizado para obter $\mathbf{w}$, como a decomposição LU ou o método do gradiente conjugado, por exemplo. Uma exposição sobre a teoria e a implementação sobre estes algoritmos pode ser encontrada em [18].

Além de ser utilizada para a predição da variável dependente, o modelo de Regressão Linear também pode ser utilizado para analisar a influência das variável independentes sobre o valor da variável dependentes, a partir da interpretação dos parâmetros do modelo [19].

No capítulo seguinte, serão apresentadas aplicações envolvendo estes dois algoritmos.

7 Aplicações

Neste capítulo, utilizaremos os modelos apresentados para resolver problemas clássicos da literatura. Para tal, a linguagem de programação Python foi utilizada, munida com a biblioteca scikit-learn, que possui diversas implementações de algoritmos de aprendizado e funcionalidades para auxiliar no processo de treinamento. A implementação do algoritmo Perceptron pode ser vista abaixo.

```
# Importa o conjunto de dados
X, y = datasets.load_digits(return_X_y = True)

# Divide os dados entre teste e treino
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size)

# Cria um perceptron
perceptron = Perceptron()

# Inicia o processo de treinamento
perceptron.fit(X_train, y_train)

# Calcula a pontuação de predição
score = perceptron.score(X_test, y_test)
```

Como podemos ver, a amostra disponível para cada problema foi obtida através da biblioteca scikit-learn, que já realiza o pré-processamento adequado, isto é, a padronização dos dados em um mesmo intervalo de valores, quando isto se faz necessário.

Ressaltamos que o objetivo deste capítulo é realizar uma breve exposição das possíveis aplicações da teoria matemática apresentada durante este trabalho, sendo os algoritmos Perceptron e Regressão Linear apenas um exemplo. Desta maneira, não foram utilizadas técnicas de seleção de modelo, uma vez que nos propusemos a implementar apenas os algoritmos discutidos na seção anterior. Além disto, não foram aplicadas técnicas de seleção de variáveis, como a análise de multicolinearidade, para verificar a influência das variáveis independentes dos problemas que serão abordados, visto que isto fugiria do escopo daquilo que está sendo abordado. Mais informações acerca destas técnicas, que são fundamentais para garantir a eficiência dos modelos de predição em situações reais, podem ser encontradas em [20], [21] e [22].

7.1 Problemas de classificação

Para os problemas de classificação, utilizaremos o algoritmo Perceptron.

A validação da capacidade de generalização é feita computando uma pontuação sobre o conjunto de teste, da seguinte maneira

$$\sum_{i=1}^{m} \left(\sum_{l=1}^{k} \mathbb{1}_{[h_S(x_i)]_l \neq [y_i]_l} \right)$$

onde h_S corresponde ao algoritmo perceptron treinado utilizando-se o conjunto de treino S e k o número de classes do problema.

Ressaltamos que, pela teoria exposta nos capítulos anteriores, já temos a garantia de que, com uma alta probabilidade, o risco do algoritmo não diverge muito do seu risco empírico. Portanto, tendo em vista a impossibilidade de calcular o risco, na prática, nossa única medida da eficiência de um algoritmo é dada por seu risco empírico, sendo isto assegurado somente se a dimensão da classe de hipóteses for finita.

7.1.1 Problema de exemplo: classificação

Uma vez que, na maioria dos casos reais, a dimensionalidade dos problemas de aprendizado é muito grande para permitir uma visualização, antes de discutirmos aplicações concretas iremos introduzir um problema exemplo.

O problema consiste em resolver uma classificação binária tendo acesso a 500 pontos gerados de maneira aleatória por uma funcionalidade da biblioteca scikit-learn.

```
# Gera um problema de classificação binária com 500 pontos
X, y = make_classification(n_samples = 500, n_features = 2,
        n_informative = 2, n_redundant = 0)
```

A distribuição destes pontos pode ser vista na imagem a seguir.

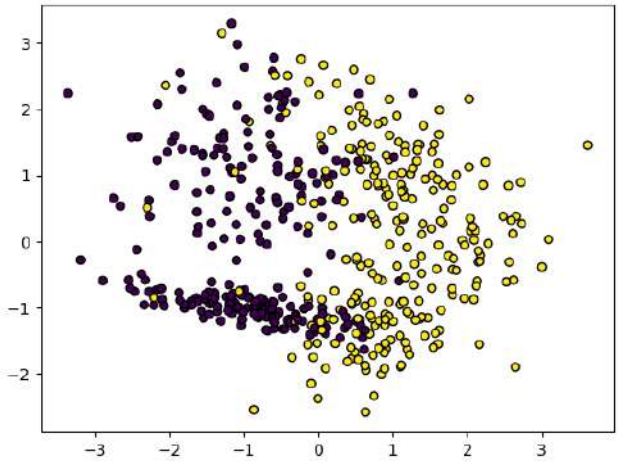

Figura 12 – Problema de classificação.

As cores diferentes indicam que os pontos pertencem a classes distintas. Aplicando o algoritmo sobre estes pontos, ele apresenta uma pontuação de predição igual a $0,9$ indicando que o Perceptron acerta a classe de 90% dos dados que pertencem ao conjunto de teste.

Devido a baixa dimensionalidade do problema, somos capazes de verificar a região de decisão, isto é, a região do plano que o modelo associa a cada classe.

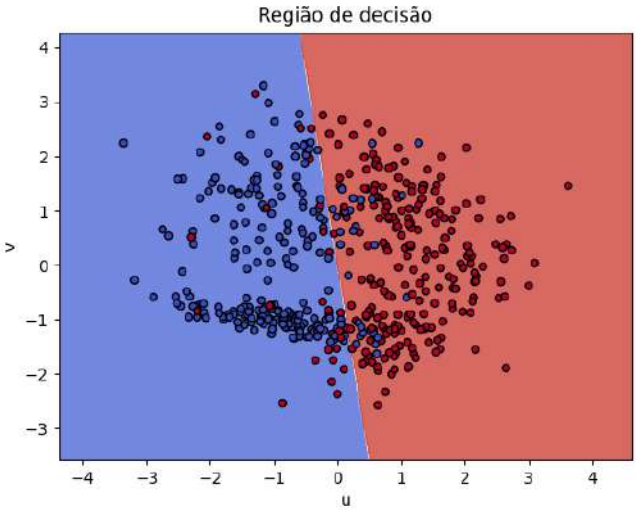

Figura 13 – Região de decisão.

Outra observação que pode ser feita é a não separabilidade linear do problema, de modo que não podemos esperar que a hipótese de consistência possa ser satisfeita.

7.1.2 Reconhecimento de dígitos

O primeiro problema concreto 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.

```
# Obtém os dados do problema de reconhecimento de dígitos
X, y = datasets.load_digits(return_X_y=True)
```

Deste modo, o domínio do problema é dado por $\mathcal{X} = \mathbb{R}^{64}$, e o conjunto de interesse é $\mathcal{Y} = \{0, 1, 2, ..., 9\}$. Aproximadamente, cada uma das dez classes possui 180 amostras, totalizando 1797 pontos.

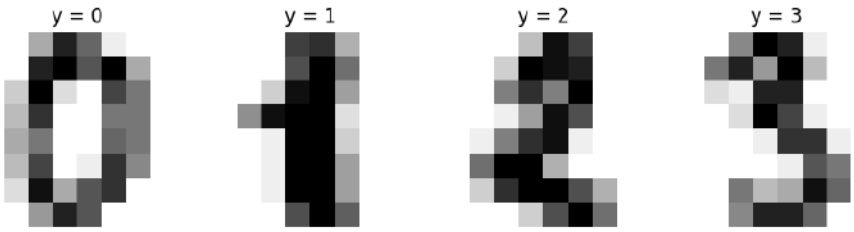

Figura 14 – Algumas imagens as quais o algoritmo tem acesso.

Nesta caso, a pontuação de predição obtida foi de 0.9. Isto significa que o modelo acertou o dígito de 90% dos pontos aos quais ele não teve acesso durante o processo de treinamento.

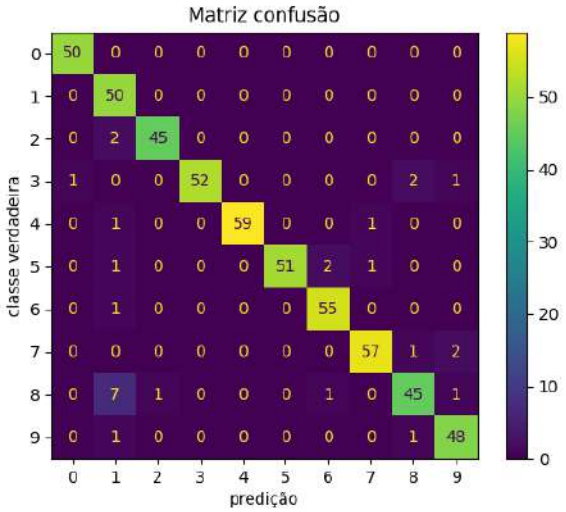

Figura 15 – Matriz confusão associada ao problema de reconhecimento de dígitos.

A Figura 15 apresenta a matriz confusão associada ao algoritmo para este problema, uma tabela que mostra as frequências de classificação para cada classe do modelo. Como podemos perceber, de fato o algoritmo foi capaz de predizer corretamente o dígito na maioria dos casos.

7.1.3 Diagnóstico de câncer de mama

O segundo problema de classificação compreende identificar se um tumor presente na mama de uma paciente é maligno (M) ou benigno (B), tendo acesso a trinta informações distintas acerca de biópisia tais quais o diâmetro do tumor, sua concavidade e textura. Ou seja, as 569 amostras do problema pertencem a $\mathbb{R}^{30} \times \{B, M\}$.

```
# Obtém os dados do problema de diagnóstico de câncer de mama
X, y = datasets.load_breast_cancer(return_X_y=True)
```

Para este problema, a pontuação de predição foi de aproximadamente $0,96$, indicando que o algoritmo Perceptron é eficaz em fazer a distinção entre um tumor benigno e um tumor maligno.

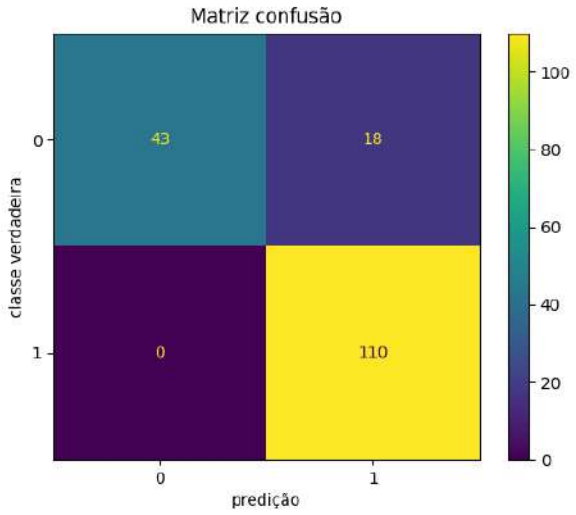

Figura 16 – Matriz confusão associada ao problema de diagnóstico de câncer de mama.

Na Figura 16, temos a matriz confusão associada ao algoritmo para este problema. Note que o conjunto $\{\text{benigno},\ \text{maligno}\}$ foi representado numericamente como $\{0,\ 1\}$. Através da matriz confusão, podemos perceber que o algoritmo não produz resultados falso negativos.

7.2 Problemas de regressão

Já para os problemas de regressão, foi utilizado o algoritmo de Regressão Linear, sendo sua pontuação de predição calculada como

$$1 - \frac{\sum_{i=1}^{m} \left(y_i - h_S(x_i)\right)^2}{\sum_{i=1}^{m} \left(y_i - \overline{y}\right)^2}$$

onde m é o tamanho do conjunto de testes, h_S o algoritmo de regressão linear treinado com o conjunto de treino S e $\overline{y}$ o valor médio das instâncias y_i pertencentes ao conjunto de teste, ou seja $\overline{y} = \sum_{i=1}^{m} \frac{y_i}{m}$.

Ressaltamos que, neste caso, a pontuação de predição difere do risco empírico, mas a utiliza em seu cálculo. Desta forma, o valor máximo de pontuação de predição que um algoritmo pode atingir é um, se ele acerta todas as predições. No entanto, esta pontuação não possui um limite inferior, indicando que um algoritmo pode ser arbitrariamente pior do que outro.

Esta pontuação é denominada Coeficiente de determinação, ou R^2, sendo muito utilizada em técnicas de regressão.

7.2.1 Problema de exemplo: regressão

Assim como foi feito para os problemas de classificação, apresentaremos primeiro uma caso de teste, a fim de permitir uma intuição visual.

Para tanto, os dados do problema de regressão foram gerados de forma aleatória, de modo a conter 500 pontos de amostra pertencentes ao espaço $\mathbb{R} \times \mathbb{R}$.

```
# Gera um problema de regressão com 500 pontos
X, y, coef = make_regression(n_samples = 500, n_features = 1 , n_informative = 1,
        coef = True, noise = 10, random_state = 0)
```

A distribuição destes pontos é dada a seguir.

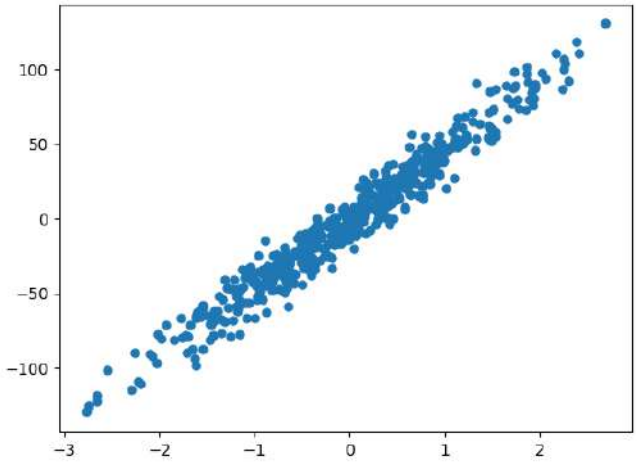

Figura 17 – Problema de regressão.

Através da Regressão Linear, obtemos uma pontuação de predição de $0,97$ o que demonstra que a função linear retornada pelo modelo descreve o comportamento do problema com bastante precisão. A seguir, temos o gráfico desta função.

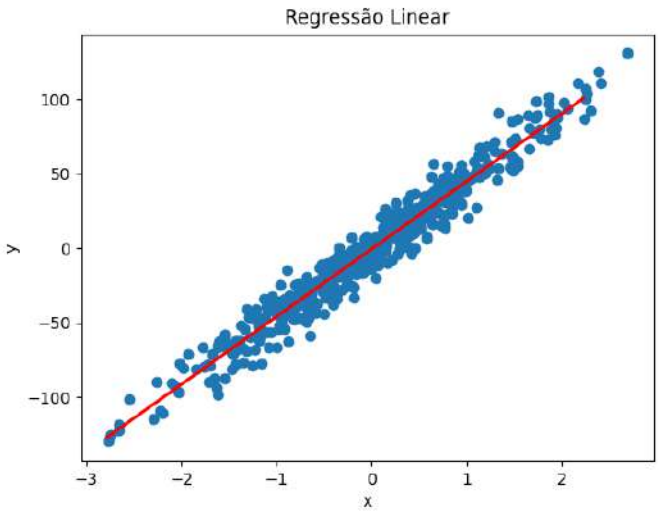

Figura 18 – Resultado da Regressão Linear.

7.2.2 Preço de casas em Boston

A primeira aplicação para problemas de regressão a qual trataremos é relativa a predição do preço de imóveis na cidade de Boston. Para tal, temos acesso a uma amostra contendo 503 pontos, cada um sendo composto por 13 informações distintas acerca do imóvel, dentre elas a taxa de criminalidade da região em que se encontra o imóvel, seu

número de dormitórios e seu tamanho. Desta maneira, o domínio do problema é dado por $\mathcal{X} = \mathbb{R}^{13}$.

```
# Obtém os dados do problema de predição do preço de imóveis
X, y = datasets.load_boston(return_X_y=True)
```

Aplicando o modelo de Regressão Linear, a pontuação de predição atingida foi $0,43$ indicando que este problema se beneficiaria da utilização de uma classe de hipóteses mais complexa do que a de funções lineares.

Entretanto, é importante destacar que não foi feita nenhuma seleção do modelo para excluir as variáveis não significativas, além disso não foi feita análise prévia para verificar variáveis aleatórias independentes muito correlacionadas, o que também atrapalha o ajuste do modelo. Ou seja, não se deve descartar o uso de regressão linear, antes, pode-se realizar uma análise dos dados [23].

7.2.3 Diabetes

Neste problema, baseado em dez características de um paciente com diabetes, tais como sexo, idade e massa corporal, queremos entender a evolução da doença no paciente. Sendo assim, a dimensionalidade do problema é igual a 10, ou seja, $\mathcal{X} = \mathbb{R}^{10}$.

```
# Obtém os dados do problema da evolução de diabetes
X, y = datasets.load_diabetes(return_X_y=True)
```

Para este problema, a pontuação de predição foi de $0,69$. Podemos comparar este resultado obtido com outros algoritmos, considerando que o maior valor que pode ser obtido é um. Assim, devemos utilizar o modelo que atinge maior pontuação de predição, em relação a outro.

Neste caso vale ressaltar que, assim como no exemplo anterior, não foi feita nenhuma análise dos dados antes de implementar a regressão linear e tal estudo poderia melhorar o ajuste do modelo aos dados.

8 Conclusão e trabalhos futuros

Como vimos, através da metodologia de aprendizado supervisionado, busca-se otimizar um modelo, através de um processo denominado treinamento. Após o algoritmo passar por esse procedimento, o qual depende de uma amostra de pontos do problema que se quer resolver, espera-se que o algoritmo seja capaz de acertar o resultado de um ponto que ele não teve acesso durante o treinamento. A isto, chamamos de capacidade de generalização.

Devido ao fato de buscarmos modelos com capacidade de generalização, devemos lidar com duas definições de erro, uma em relação a amostra, e outra em relação a um ponto qualquer do problema que é obtido de maneira aleatória. Embora este último seja aquele o qual desejamos minimizar, ele não pode ser calculado diretamente, o que motiva a definição de minimização do risco empírico. Sob este paradigma, foram definidas duas noções de aprendizado, utilizadas para formalizar quando é esperado que o erro na amostra não difira muito do erro para além dela.

Após discutirmos alguns aspectos relacionados a como fazer uma boa escolha de um modelo para resolver um problema de aprendizado, e a teoria que justifica o fato de que classes infinitas de funções podem ser utilizadas para embasar algoritmos, foram apresentadas aplicações concretas para as definições apresentadas.

No entanto, os modelos apresentados são simples, de modo que podem não se mostrar efetivos em alguns casos. A partir disso, um dos pontos inicias em trabalhos futuros seria realizar o estudo de outros modelos. Outro ponto seria pesquisar outros paradigmas para embasar definições de aprendizado, como o princípio de minimização do risco estrutural, que se mostra menos sujeito à ocorrência de *overfitting*. Finalmente, outro assunto que pode ser estudado diz respeito às medidas de complexidade e dimensão, e as cotas de aprendizado que podem ser obtidas através delas.

Referências

1 TZANIS, G. et al. Modern applications of machine learning. 01 2006. Citado na página 9.

2 DAS, S. et al. Applications of artificial intelligence in machine learning: Review and prospect. *International Journal of Computer Applications*, v. 115, p. 31–41, 04 2015. Citado na página 9.

3 Simeone, O. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, v. 4, n. 4, p. 648–664, 2018. Citado na página 9.

4 JAMES, G. et al. *An Introduction to Statistical Learning: With Applications in R*. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1461471370. Citado na página 17.

5 MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.]: The MIT Press, 2012. ISBN 026201825X. Citado na página 23.

6 STRATOS, K. *PAC Learnability*. <http://karlstratos.com/notes>. Acessado em: 23/02/2021. Citado na página 23.

7 HAUSSLER, D. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, v. 100, n. 1, p. 78–150, 1992. ISSN 0890-5401. Citado na página 31.

8 BASHIR, D. et al. *An Information-Theoretic Perspective on Overfitting and Underfitting*. 2020. Citado na página 39.

9 VAPNIK, V.; CHERVONENKIS, A. On the uniform convergence of relative frequencies of events to their probabilities. In: _____. [S.l.: s.n.], 2015. p. 11–30. Citado na página 48.

10 PLATEN, E. Pollard, d.:convergence of stochastic processes. (springer series in statistics). springer-verlag, new york - berlin - heidelberg - tokyo 1984, 216 pp., 36 illustr., dm 82. *Biometrical Journal*, v. 28, n. 5, p. 644–644, 1986. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.4710280516>. Citado na página 50.

11 DANIELY, A. et al. Multiclass learnability and the erm principle. *Journal of Machine Learning Research - Proceedings Track*, v. 19, p. 207–232, 01 2011. Citado 2 vezes nas páginas 50 e 55.

12 AWASTHI, P.; FRANK, N.; MOHRI, M. *On the Rademacher Complexity of Linear Hypothesis Sets*. 2020. Citado na página 50.

13 BENDAVID, S. et al. Characterizations of learnability for classes of (0, ..., n)-valued functions. *Journal of Computer and System Sciences*, v. 50, n. 1, p. 74–86, 1995. ISSN 0022-0000. Citado 2 vezes nas páginas 51 e 55.

14 SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014. ISBN 1107057132. Citado na página 52.

15 CORANI, G.; GATTO, M. Structural risk minimization: a robust method for density-dependence detection and model selection. *Ecography*, v. 30, n. 3, p. 400–416, 2007. Citado na página 53.

16 ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958. Citado na página 55.

17 STANTON, J. Galton, pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education*, v. 9, 01 2001. Citado na página 57.

18 BURDEN, A.; BURDEN, R.; FAIRES, J. *Numerical Analysis, 10th ed.* [S.l.: s.n.], 2016. ISBN 1305253663. Citado na página 58.

19 KUMARI, K.; YADAV, S. Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*, v. 4, p. 33, 01 2018. Citado na página 58.

20 DING, J.; TAROKH, V.; YANG, Y. Model selection techniques: An overview. *IEEE Signal Processing Magazine*, v. 35, 11 2018. Citado na página 59.

21 MIAO, J.; NIU, L. A survey on feature selection. *Procedia Computer Science*, v. 91, p. 919–926, 12 2016. Citado na página 59.

22 VISA, S. et al. Confusion matrix-based feature selection. In: . [S.l.: s.n.], 2011. v. 710, p. 120–127. Citado na página 59.

23 DUNTEMAN, G. H. *Introduction to Linear Models*. [S.l.]: SAGE Publications, 1984. ISBN 0803921756. Citado na página 66.

Apêndices

APÊNDICE A – Probabilidade

Neste apêndice, são expostas as definições e os resultados de teoria de Probabilidade que foram utilizados ao longo do texto.

Definição A.1. Uma classe de subconjuntos de Ω, denotada por $\mathcal{F}$, é dita uma σ-álgebra se satisfaz

1. $\Omega \in \mathcal{F}$;
2. Se $A \in \mathcal{F}$, então $A^c \in \mathcal{F}$;
3. Se $A_i \in \mathcal{F}$, para $i \geq 1$, então $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.

Exemplo A.2. *A σ-álgebra de Borel, denotada por $\mathcal{B}(\mathbb{R})$, é a menor σ-álgebra que contém todos os intervalos abertos, e portanto todos os intervalos fechados, de $\mathbb{R}$. Como, na maioria dos casos, os dados de um problema de aprendizado são números ou vetores de números reais, consideraremos esta σ-álgebra ao longo do texto. Esta σ-álgebra também é utilizada na teoria de Probabilidade, uma vez que, como veremos, uma probabilidade é uma função que assume valores reais.*

Definição A.3. Um espaço mensurável é uma par $(\Omega, \mathcal{F})$, onde Ω é um conjunto qualquer e $\mathcal{F}$ é uma σ-álgebra de Ω.

Definição A.4. Dado o espaço mensurável $(\Omega, \mathcal{F})$, uma função $f : \Omega \to \mathbb{R}$ é dita $\mathcal{F}$-mensurável se

$$f^{-1}(A) \in \mathcal{F}, \ \forall A \in \mathcal{B}(\mathbb{R}).$$

Definição A.5. Dado um espaço mensurável $(\Omega, \mathcal{F})$ uma medida, ou distribuição, de probabilidade é uma função $P : \mathcal{F} \to [0, 1]$ que satisfaz:

1. $P(\Omega) = 1$;
2. $\forall A \in \mathcal{F}, \ P(A) \geq 0$;
3. Para toda sequência $A_1, A_2, ... \in \mathcal{F}$ onde cada elemento é mutuamente exclusivo

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

Definição A.6. Um espaço de probabilidade é uma tríade $(\Omega, \mathcal{F}, P)$ onde Ω é um conjunto qualquer, $\mathcal{F}$ é uma σ-álgebra sobre Ω e P é uma medida de probabilidade definida no espaço mensurável $(\Omega, \mathcal{F})$.

Definição A.7. A probabilidade condicional de uma evento A dado um evento B é dada por

$$\mathbb{P}[A \mid B] = \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}.$$

Definição A.8. Dois eventos A e B são ditos independentes se

$$\mathbb{P}[A \cap B] = \mathbb{P}[A]\,\mathbb{P}[B].$$

Proposição A.9 (cota da união)**.** *Se $A_1, , ..., A_n$ são eventos independentes, então*

$$\mathbb{P}\left[\bigcup_{i=1}^{n} A_i\right] \leq \sum_{i=1}^{n} \mathbb{P}[A_i].$$

Definição A.10. Dado um espaço de probabilidade $(\Omega, \mathcal{F}, P)$, uma variável aleatória é uma função $X : \Omega \to \mathbb{R}$ que é $\mathcal{F}$-mensurável.

Definição A.11. Dado um espaço de probabilidade $(\Omega, \mathcal{F}, P)$, um vetor aleatório, ou variável aleatória multidimensional, é um vetor $X = (X_1, ..., X_m)$ m dimensional onde cada X_i , $i \in \{1, ..., m\}$, é uma variável aleatória no espaço de probabilidade $(\Omega, \mathcal{F}, P)$.

Definição A.12. Dadas duas variáveis aleatórias X e Y em um espaço de probabilidade $(\Omega, \mathcal{F}, P)$, definimos a distribuição conjunta de X e Y por

$$P_{X,Y}(x, y) = P(X = x \wedge Y = y).$$

A partir disto, definimos a distribuição marginal de X, e de forma análoga de Y, como

$$P_X(x) = \int_y P_{X,Y}(x, y)dy.$$

Definição A.13. Dadas duas variáveis aleatórias X e Y em $(\Omega, \mathcal{F}, P)$ e $B \in \mathcal{F}$ tal que $P[Y \in B] > 0$, a função $A \in \mathcal{F} \to P[X \in A \mid Y \in B]$ onde

$$P[X \in A \mid Y \in B] = \frac{P[(X \in A) \cap (Y \in B)]}{P[Y \in B]}$$

é chamada distribuição condicional, e satisfaz as condições para ser uma medida de probabilidade em $(\Omega, \mathcal{F})$.

Definição A.14. Duas variáveis aleatórias X e Y em $(\Omega, \mathcal{F}, P)$ são ditas independentes se, para quaisquer $A, B \in \mathcal{F}$,

$$P(X \in A \mid Y \in B) = P[X \in A].$$

Definição A.15. Se $X_1, ..., X_n$ são variáveis aleatórias independentes entre si no espaço de probabilidade $(\Omega, \mathcal{F}, P)$, então, dados $A_1, ..., A_n \in \mathcal{F}$

$$P[X_1 \in A_1, ..., X_n \in A_n] = \prod_{i=1}^{n} P[X_i \in A_i].$$

Definição A.16. As variáveis aleatórias $X_1, .., X_n$ no espaço de probabilidade $(\Omega, \mathcal{F}, P)$ são ditas independentes e identicamente distribuídas se são independentes entre si e possuem a mesma esperança μ.

Definição A.17. Seja X uma variável aleatória definida no espaço de probabilidade $(\Omega, \mathcal{F}, P)$. A esperança de X, ou valor esperado, é dado por

$$\mathbb{E}[X] = \int_{\Omega} X(w) dP(w).$$

Definição A.18. Seja X uma variável aleatória tal que $\mathbb{E}[X] = \mu < \infty$. A variância de X é dada por

$$Var(X) = \mathbb{E}[(X - \mu)^2].$$

Teorema A.19 (Lei fraca dos grandes números)**.** *Seja $(X_n)_{n\in\mathbb{N}}$ uma sequência de variáveis aleatórias independentes com mesma esperança μ e variância $\sigma^2 < \infty$. Dado $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, para qualquer $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}[|\overline{X}_n - \mu| \geq \epsilon] = 0.$$

Teorema A.20 (Desigualdade de Hoeffding)**.** *Seja $(X_n)_{n\in\mathbb{N}}$ uma sequência de variáveis aleatórias independentes com mesma esperança μ. Dado $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, se $\mathbb{E}[\overline{X}] = \mu$ e $\mathbb{P}[a \leq X_i \leq b]$ para todo $i \in [n]$, então, para todo $\epsilon > 0$,*

$$\mathbb{P}[|\overline{X} - \mu| > \epsilon] \leq 2\exp(-2m\epsilon^2/(b-a)^2).$$

Teorema A.21 (Desigualdade de Markov)**.** *Seja X uma variável aleatória não negativa. Para todo $a \geq 0$,*

$$\mathbb{P}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

Teorema A.22 (Desigualdade de Jensen)**.** *Sejam X uma variável aleatória e g uma função convexa. Então,*

$$\mathbb{E}[g(X)] \geq g(\mathbb{E}[X]).$$

APÊNDICE B – Lemas

Lema B.1. *Seja Z uma variável aleatória que toma valores em $[0,1]$. Assuma que $\mathbb{E}[Z] = \mu$. Logo, para todo $a \in (0,1)$,*

$$\mathbb{P}[Z > 1-a] \geq \frac{\mu - (1-a)}{a}.$$

Ademais, isto também implica que, para todo $a \in (0,1)$,

$$\mathbb{P}[Z > a] \geq \frac{\mu - a}{1-a} \geq \mu - a.$$

Demonstração. Seja $Y = 1 - Z$. Temos que Y é uma variável aleatória não-negativa com $\mathbb{E}[Y] = 1 - \mathbb{E}[Z] = 1 - \mu$. Aplicando a esigualdade de Markov em Y, obtemos

$$\mathbb{P}[Z \leq 1-a] = \mathbb{P}[1 - Z \geq a] = \mathbb{P}[Y \geq a] \leq \frac{\mathbb{E}[Y]}{a} = \frac{1-\mu}{a}.$$

Logo,

$$\mathbb{P}[Z > 1-a] \geq 1 - \frac{1-\mu}{a} = \frac{a + \mu - 1}{a}.$$

■

Lema B.2 (Sauer-Shelah-Perles)**.** *Seja $\mathcal{H}$ uma classe de hipóteses com $VCdim(\mathcal{H}) = d < \infty$. Então, para todo m*

$$\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$$

Em particular, se $m > d+1$ então $\tau_{\mathcal{H}}(m) \leq (em \backslash d)^d$.

Demonstração. É suficiente mostrar a afirmação mais forte:

para qualquer $C = \{c_1, ..., c_m\}$ temos

$$\forall \mathcal{H},\ |\mathcal{H}_C| \leq |\{B \subseteq C\ :\ \mathcal{H} \text{ fragmenta } B\} \tag{B.1}$$

pois, se $VCdim(\mathcal{H}) \leq d$ então nenhum conjunto de tamanho maior do que d é fragmentado por $\mathcal{H}$, e portanto

$$|\{B \subseteq C\ :\ \mathcal{H} \text{ fragmenta } B\}| \leq \sum_{i=0}^{d} \binom{m}{i}$$

quando $m > d+1$ o lado direito da equação é no máximo $(em/d)^d$.

Vamos provar (B.1) usando indução:

Para $m = 1$, não importa qual $\mathcal{H}$ tomamos, os dois lados da equação tem que ser iguais a 1 ou 2 (consideramos que $\mathcal{H}$ fragmenta o conjunto vazio).

Assuma que (B.1) valha para conjuntos de tamanho $k < m$, vamos provar que também tem validade para conjuntos de tamanho m.

Fixe $\mathcal{H}$ e $C = \{c_1, ..., c_m\}$. Denote $C' = \{c_2, ..., c_m\}$ e defina

$$Y_0 = \{(y_2, ..., y_m) \ : \ (0, y_2, ..., y_m) \in \mathcal{H}_C \vee (1, y_2, ..., y_m) \in \mathcal{H}_C\}$$

$$Y_1 = \{(y_2, ..., y_m) \ : \ (0, y_2, ..., y_m) \in \mathcal{H}_C \wedge (1, y_2, ..., y_m) \in \mathcal{H}_C\}$$

é fácil ver que $|\mathcal{H}_C| = |Y_0| + |Y_1|$.

Ademais, como $Y_0 = \mathcal{H}_{C'}$, segue da hipótese de indução que

$$|Y_0| = |\mathcal{H}_{C'}| \leq |\{B \subseteq C^{'} \ : \ \mathcal{H} \text{ fragmenta } B\}| = |\{B \subseteq C \ : \ c_1 \notin B \ \wedge \ \mathcal{H} \text{ fragmenta } B\}|$$

Agora, defina $\mathcal{H}^{'} \subseteq \mathcal{H}$ como sendo

$$\mathcal{H}^{'} = \{h \in \mathcal{H} \ : \ \exists \, h^{'} \in \mathcal{H}; \ (1 - h'(c_1), h'(c_2), ..., h'(c_m)) = (h(c_1), ..., h(c_m))\}$$

ou seja, $\mathcal{H}'$ contém pares de hipóteses que concordam em C' e discordam em c_1.

Devido a esta definição, temos que se $\mathcal{H}'$ fragmenta um conjunto $B \subseteq C'$ então ele também fragmenta $B \cup \{c_1\}$ e vice versa. Combinando isto com o fato de $Y_1 = \mathcal{H}'_{C'}$ e utilizando a hipótese de indução, obtemos

$$\begin{aligned} |Y_1| = |\mathcal{H}'_{C'}| &\leq |\{B \subseteq C' \ : \ \mathcal{H}' \text{ fragmenta } B\}| \\ &= |\{B \subseteq C' \ : \ \mathcal{H}' \text{ fragmenta } B \cup \{c_1\}\}| \\ &= |\{B \subseteq C \ : \ c_1 \in B \ \wedge \ \mathcal{H}' \text{ fragmenta } B\}| \\ &\leq |\{B \subseteq C \ : \ c_1 \in B \ \wedge \ \mathcal{H} \text{ fragmenta } B\}|. \end{aligned}$$

Portanto, mostramos que

$$\begin{aligned} |\mathcal{H}_C| &= |Y_0| + |Y_1| \\ &\leq |\{B \subseteq C \ : \ c_1 \notin B \ \wedge \ \mathcal{H} \text{ fragmenta } B\}| + |\{B \subseteq C \ : \ c_1 \in B \ \wedge \ \mathcal{H} \text{ fragmenta } B\}| \\ &= |\{B \subseteq C \ : \ \mathcal{H} \text{ fragmenta } B\}| \end{aligned}$$

■

Lema B.3. *Sejam X uma variável aleatória e $x' \in \mathbb{R}$. Assuma que existem $a > 0$ e $b \geq e$ tais que, para todo $t \geq 0$, temos $\mathbb{P}[|X - x'| > t] \leq 2be^{-t^2/a^2}$. Logo, $\mathbb{E}[|X - x'|] \leq a(2 + \sqrt{\log(b)})$.*

Demonstração. Como $|X - x'|$ é uma variável aleatória positiva, podemos escrever

$$\mathbb{E}[|X - x'|] = \int_0^\infty \mathbb{P}[|X - x'| > x]dx.$$

Para todo $i = 0, 1, ...,$ defina $t_i = a(i + \sqrt{\log(b)})$. Uma vez que t_i é monotonamente crescente, temos que $\mathbb{P}[|X - x'| > t_0] \geq \mathbb{P}[|X - x'| > x]$ para todo $x \in [t_0, t_1]$. Assim,

$$\begin{aligned}
\mathbb{E}[|X - x'|] &= \int_0^\infty \mathbb{P}[|X - x'| > x]dx \\
&= \int_0^{t_0} \mathbb{P}[|X - x'| > x]dx \ + \ \sum_{i=1}^{\infty} \int_{t_{i-1}}^{t_i} \mathbb{P}[|X - x'| > x]dx \\
&\leq \int_0^{t_0} \mathbb{P}[|X - x'| > 0]dx \ + \ \sum_{i=1}^{\infty} \int_{t_{i-1}}^{t_i} \mathbb{P}[|X - x'| > t_{i-1}]dx \\
&= 1 \int_0^{t_0} dx \ + \ \sum_{i=1}^{\infty} \mathbb{P}[|X - x'| > t_{i-1}] \int_{ti-1}^{t_i} dx \\
&= 1(t_0 - 0) \ + \ \sum_{i=1}^{\infty} \mathbb{P}[|X - x'| > t_i](t_i - t_{i-1}) \\
&\leq t_0 + \sum_{i=1}^{\infty} t_i \, \mathbb{P}[|X - x'| > t_{i-1}] \\
&= a\sqrt{\log(b)} + \sum_{i=1}^{\infty} t_i \, \mathbb{P}[|X - x'| > t_{i-1}].
\end{aligned}$$

Ou seja,

$$\mathbb{E}[|X - x'|] \leq a\sqrt{\log(b)} + \sum_{i=1}^{\infty} t_i \, \mathbb{P}[|X - x'| > t_{i-1}]. \tag{B.2}$$

Utilizando a hipótese feita sobre $\mathbb{P}[|X - x'| > t]$, temos

$$\begin{aligned}
\sum_{i=1}^{\infty} t_i \, \mathbb{P}[|X - x'| > t_{i-1}] &\leq 2ab \sum_{i=1}^{\infty} (i + \sqrt{\log(b)}) e^{-(i-1+\sqrt{\log(b)})^2} \\
&\leq 2ab \int_{1+\sqrt{\log(b)}}^{\infty} x e^{-(x-1)^2} dx \\
&= 2ab \int_{\sqrt{\log(b)}}^{\infty} (y + 1) e^{-y^2} dy \\
&\leq 4ab \int_{\sqrt{\log(b)}}^{\infty} y e^{-y^2} dy \\
&= 2ab \Big[- e^{-y^2} \Big]_{\sqrt{\log(b)}}^{\infty} \\
&= \frac{2ab}{b} \\
&= 2a.
\end{aligned} \tag{B.3}$$

Finalmente, combinando as desigualdades (*B.2*) e (*B.3*) , o resultado segue. ■

Lema B.4. *Seja $\mathcal{H}$ uma classe e $\tau_{\mathcal{H}}$ sua função de crescimento. Então, para cada distribuição D e $\delta \in (0, 1)$, com probabilidade de pelo menos $1 - \delta$ sobre $S \sim D^m$, temos*

$$|L_D(h) - L_S(h)| \leq \frac{4 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\delta\sqrt{2m}}.$$

Demonstração. Vamos começar mostrando que

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|L_D(h) - L_{\mathcal{S}}(h)\right|\right] \leq \frac{4 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}}. \tag{B.4}$$

Para cotar o lado esquerdo da equação, note que $\forall h \in \mathcal{H}$ podemos reescrever $L_D(h) = \mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m}[L_{\mathcal{S}'}(h)]$, onde $S' = z'_1, ..., z'_m$ é uma amostra i.i.d adicional. Portanto

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|L_D(h) - L_{\mathcal{S}}(h)\right|\right] = \mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|\mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m} L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right|\right]. \tag{B.5}$$

Pela desigualdade de Jensen

$$\left|\mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m} L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right| \leq \mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m}\left|L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right| \tag{B.6}$$

e,

$$\sup_{h\in\mathcal{H}}\mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m}\left|L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right| \leq \mathop{\mathbb{E}}_{\mathcal{S}'\sim D^m}\sup_{h\in\mathcal{H}}\left|L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right|. \tag{B.7}$$

Combinando (B.5), (B.6) e (B.7), obtemos

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|L_D(h) - L_{\mathcal{S}}(h)\right|\right] &\leq \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|L_{\mathcal{S}'}(h) - L_{\mathcal{S}}(h)\right|\right] \\ &= \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'\sim D^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}(l(h, z'_i) - l(h, z_i))\right|\right]. \end{aligned} \tag{B.8}$$

O valor esperado é sobre a escolha de duas amostras i.i.d $S = z_1, ..., z_m$ e $S' = z'_1, ..., z'_m$. Como esses $2m$ vetores são escolhidos i.i.d, podemos trocar z_i por z'_i. Se isto for feito, na equação acima, ao invés de termos $(l(h, z'_i) - l(h, z_i))$ teremos $-(l(h, z'_i) - l(h, z_i))$. Logo, para qualquer $\sigma \in \{\pm 1\}^m$, (B.8) é igual a

$$\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'\sim D^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(l(h, z'_i) - l(h, z_i))\right|\right].$$

Como isto é válido para qualquer $\sigma \in \{\pm 1\}^m$, também será válido se obtermos cada σ_i, uniformemente de acordo com uma distribuição $U_\pm$ sobre $\{\pm 1\}$. Assim, (B.8) também é igual a

$$\mathop{\mathbb{E}}_{\sigma\sim U_\pm}\mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'\sim D^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(l(h, z'_i) - l(h, z_i))\right|\right] = \mathop{\mathbb{E}}_{\mathcal{S},\mathcal{S}'\sim D^m}\mathop{\mathbb{E}}_{\sigma\sim U_\pm}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(l(h, z'_i) - l(h, z_i))\right|\right].$$

Agora, fixe S e S' e seja C as intâncias que estão em S e S'. Assim, podemos tomar o supremo apenas sobre H_C, portanto

$$\mathop{\mathbb{E}}_{\sigma\sim U_\pm}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(l(h,z_i')-l(h,z_i))\right|\right]=\mathop{\mathbb{E}}_{\sigma\sim U_\pm}\left[\max_{h\in\mathcal{H}_C}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(l(h,z_i')-l(h,z_i))\right|\right].$$

Fixe $h\in\mathcal{H}_C$ e denote $\theta_h=\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\sigma_i(l(h,z_i')-l(h,z_i))$. Como $\mathbb{E}[\theta_h]=0$ e θ_h é a média de variáveis aleatórias independentes que têm valores em $[-1,1]$, pela desigualdade de Hoeffding

$$\forall\ \rho>0,\ \mathbb{P}[|\theta_h|>\rho]\leq 2e^{-2m\rho^2}$$

e, aplicando a cota da união

$$\forall\ \rho>0,\ \mathbb{P}\left[\max_{h\in\mathcal{H}_C}|\theta_h|>\rho\right]\leq 2|\mathcal{H}_C|e^{-2m\rho^2}.$$

Ademais, aplicando o Lema B.3, temos

$$\mathbb{E}\left[\max_{h\in\mathcal{H}_C}\left|\theta_h\right|\right]\leq\frac{4+\sqrt{log(|\mathcal{H}_C|)}}{\sqrt{2m}}.$$

Combinando estas relações com a definição de $\tau_{\mathcal{H}}$, temos que

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}\left|L_D(h)-L_{\mathcal{S}}(h)\right|\right]\leq\frac{4+\sqrt{log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}}.\tag{B.9}$$

Por outro lado, sabemos que

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_D(h)-L_{\mathcal{S}}(h)|>\epsilon\right]\leq\frac{1}{\epsilon}\mathop{\mathbb{E}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_D(h)-L_{\mathcal{S}}(h)|\right].$$

Assim, utilizando a desigualdade (B.9) e tomando $\epsilon=\dfrac{1}{\delta}\sqrt{\dfrac{2d\log(2em/d)}{m}}$, obtemos que

$$\mathop{\mathbb{P}}_{\mathcal{S}\sim D^m}\left[\sup_{h\in\mathcal{H}}|L_D(h)-L_{\mathcal{S}}(h)|>\epsilon\right]\leq\delta$$

de modo que o resultado segue. ■

Lema B.5. *Seja $a\in\mathbb{R}$, $a>0$. Se $x\geq 2a\log(a)$, então $x\geq a\log(x)$. Disto segue que $x<a\log(x)$ se, e somente se, $x<2a\log(a)$.*

Demonstração. Primeiramente, note que a desigualdade é válida para todo $a\in(0,\sqrt{e}]$. Ademais, para $a>\sqrt{e}$, considere a função $f(x)=x-a\log(x)$ cuja derivada é

$f'(x) = 1 - \frac{a}{x}$. Assim, para x > a, f' é positiva de modo que f é crescente. Além disto,

$$\begin{aligned} f(2a\log(a)) &= 2a\log(a) - a\log(2a\log(a)) \\ &= 2a\log(a) - a\log(a) - a\log(2\log(a)) \\ &= a\log(a) - a\log(2\log(a)). \end{aligned}$$

Como $a - 2\log(a) > 0$ para todo $a > 0$, o resultado segue. ■

Lema B.6. *Sejam $a, b \in \mathbb{R}$ tais que $a \geq 1$ e $b > 0$. Se $x \geq 4a\log(2a) + 2b$, então $x \geq a\log(x) + b$.*

Demonstração. Provar este resultado é análogo a mostrar que $x \geq 4a\log(2a) + 2b$ implica, ao mesmo tempo, que $x \geq 2a\log(x)$ e $x \geq 2b$.

Por hipótese, como $a \geq 1$, temos $4a\log(2a) \geq 0$, o que, por sua vez, implica que $x \geq 4a\log(2a) + 2b \geq 2b$. Por outro lado, como $b > 0$, temos $x \geq 4a\log(2a) + 2b \geq 4a\log(2a)$ de modo que, pelo Lema B.5, segue que $x \geq 2a\log(x)$. ■