Teoria de Aprendizado Estatístico

Thiago Rodrigo Ramos

October 10, 2024

O que é Aprendizado?

- O aprendizado envolve a habilidade de prever resultados futuros com base em dados históricos.
- Exemplo prático: No conjunto de dados **Iris**, o modelo aprende a classificar diferentes espécies de flores utilizando características como o comprimento e a largura das pétalas.
- A aprendizagem permite generalizar: o modelo pode prever a espécie de uma flor com características similares às de outras flores já observadas durante o treinamento.

Exemplo de Aprendizado em ML

- Suponha que desejamos prever o preço de imóveis usando o conjunto de dados **California Housing**.
- Um modelo de aprendizado pode analisar atributos como o número de quartos, proximidade do oceano, e outros fatores.
- O objetivo é treinar um modelo que consiga generalizar bem, prevendo o preço de imóveis que não fizeram parte do conjunto de treinamento original.

O Framework de Aprendizado Estatístico

- **Entrada do Modelo**: No aprendizado estatístico, o modelo tem acesso a:
  - **Conjunto de Domínio** ($\mathcal{X}$): Um conjunto arbitrário de características, como as do dataset *Iris* (comprimento das pétalas, largura das sépalas, etc.) ou as características de imóveis no *California Housing*.
  - **Conjunto de Rótulos** ($\mathcal{Y}$): Um conjunto de rótulos, que pode ser binário, como $\{0, 1\}$ (ex.: 0 = não-setosa, 1 = setosa no *Iris*).
  - **Dados de Treinamento**: Um conjunto de pares rotulados $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, onde cada $x_i \in \mathcal{X}$ e cada $y_i \in \mathcal{Y}$. Exemplo: características de flores no *Iris* e seus rótulos, ou características de imóveis no *California Housing* e seus preços.

O Framework de Aprendizado Estatístico

- **Saída do Modelo**: O modelo retorna uma regra de predição $h : \mathcal{X} \to \mathcal{Y}$, também chamada de classificador ou hipótese.
  - O objetivo é encontrar um classificador que consiga prever com precisão os rótulos de novos dados.
  - **Exemplo**: No conjunto de dados *Iris*, a função h pode classificar se uma flor pertence à espécie setosa com base em características como o comprimento e a largura das pétalas.

Minimização de Risco Empírico (ERM)

- Um algoritmo de aprendizado recebe como entrada um conjunto de treinamento $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ **i.i.d.**, amostrado de uma distribuição desconhecida D sobre o espaço de características $\mathcal{X}$.
- Cada x_i é uma amostra de $\mathcal{X}$, rotulada por uma função-alvo desconhecida $f : \mathcal{X} \to \mathcal{Y}$, que mapeia cada amostra para seu rótulo correspondente y_i.
- O objetivo do algoritmo é encontrar um preditor $h_S : \mathcal{X} \to \mathcal{Y}$ que minimize o erro em relação a D e f, mesmo que ambos sejam desconhecidos.

Amostra i.i.d.

Amostra i.i.d. (independente e identicamente distribuída):

- Uma amostra é dita i.i.d. se seus elementos são:
  - **Independentes**: O resultado de uma observação não afeta as outras. Cada observação é gerada de forma independente das demais.
  - **Identicamente distribuídos**: Todas as observações são geradas a partir da mesma distribuição de probabilidade. Ou seja, todas seguem a mesma distribuição $\mathcal{D}$.
- Exemplo prático: Ao selecionar várias flores do conjunto de dados *Iris*, cada flor é uma observação independente das demais e todas as flores são amostradas da mesma distribuição de características.

Erro de Treinamento

- Como D (a distribuição que gera as amostras) e f (a função que rotula as amostras) são desconhecidos, o erro verdadeiro não pode ser calculado diretamente.
- No entanto, o **erro de treinamento**, que pode ser calculado pelo modelo, é uma aproximação do erro verdadeiro:

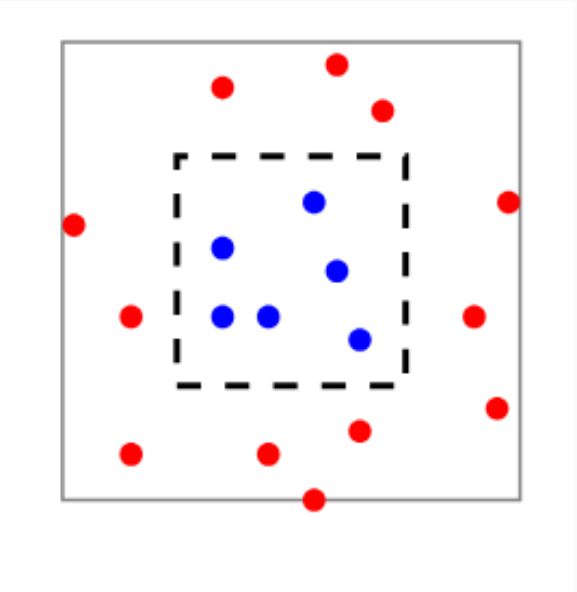
$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{h(x_i) \neq y_i\},$$

  onde m é o número de exemplos no conjunto de treinamento S, e $\mathbb{1}\{h(x_i) \neq y_i\}$ é uma função indicadora que vale 1 se $h(x_i) \neq y_i$ e 0 caso contrário.
- O erro de treinamento também é chamado de *erro empírico* ou *risco empírico*.

Paradigma de Aprendizado ERM

- Como o conjunto de treinamento S é a única amostra disponível do mundo real, o modelo busca uma solução que minimize o erro sobre esses dados.
- Este paradigma de aprendizado, que procura encontrar um preditor h que minimize $L_S(h)$, é chamado de **Minimização de Risco Empírico (ERM)**.

- A regra de Minimização de Risco Empírico (ERM) pode levar a *overfitting*, ou seja, quando o modelo se ajusta excessivamente aos dados de treinamento.



- Em vez de abandonar o paradigma de ERM, uma solução comum é restringir o espaço de busca do modelo, aplicando a regra ERM sobre uma **classe de hipóteses** limitada, denotada por H.

Classe de Hipóteses e Escolha de Preditor

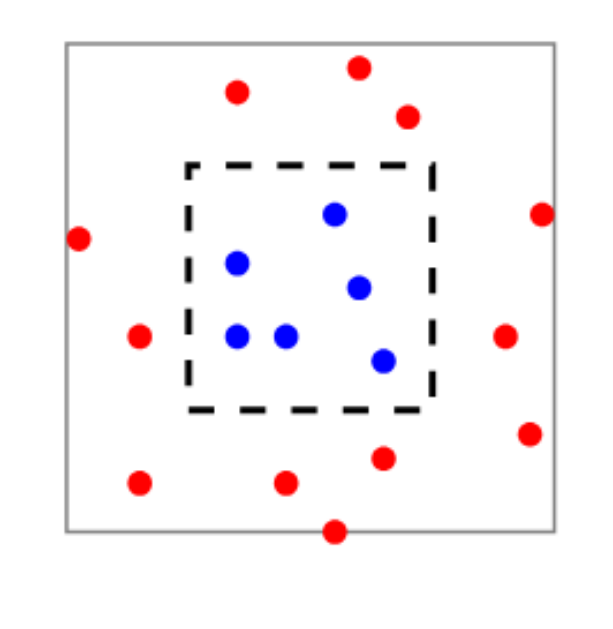
- Cada $h \in \mathcal{H}$ é uma função que mapeia de $\mathcal{X}$ para $\mathcal{Y}$, ou seja, $h : \mathcal{X} \to \mathcal{Y}$.
- Para uma dada classe $\mathcal{H}$ e um conjunto de treinamento S, o modelo ERM busca escolher um preditor $h \in \mathcal{H}$ que minimize o erro de treinamento $L_S(h)$:

$$ERM_{\mathcal{H}}(S) \in \arg\min_{h \in \mathcal{H}} L_S(h),$$

  onde $\arg\min$ denota o conjunto de hipóteses em $\mathcal{H}$ que minimizam $L_S(h)$.
- Ao restringir a escolha de preditores a uma classe $\mathcal{H}$, introduzimos um **viés indutivo**, ou seja, uma preferência por uma classe específica de preditores.

Viés Indutivo

- O viés indutivo é escolhido antes de observar os dados e é baseado em algum conhecimento prévio sobre o problema.
- Exemplo: Para prever o exemplo abaixo, poderíamos restringir $\mathcal{H}$ a um conjunto de preditores definidos por retângulos alinhados aos eixos.



Restrição de Hipóteses e Overfitting

- Restringir a classe de hipóteses $\mathcal{H}$ nos protege contra *overfitting*, mas também aumenta o viés indutivo.
- O desafio está em encontrar uma classe $\mathcal{H}$ suficientemente restrita para evitar *overfitting*, mas não tão restrita a ponto de introduzir um viés muito forte.

Erro Empírico e Erro Verdadeiro Revisados

- Para uma distribuição de probabilidade D sobre $\mathcal{X} \times \mathcal{Y}$, podemos medir a probabilidade de um preditor h cometer um erro quando pontos rotulados são amostrados de acordo com D.
- O **erro verdadeiro** (ou risco) de uma regra de predição h é redefinido como:

$$L_D(h) \stackrel{\text{def}}{=} P_{(x,y)\sim D}[h(x) \neq y].$$

- Nosso objetivo é encontrar um preditor h para o qual esse erro seja minimizado.

O Problema do Aprendiz

- O modelo não tem conhecimento direto da distribuição D que gera os dados. O que ele tem acesso é aos **dados de treinamento**, S.
- A definição de **risco empírico** continua a mesma, ou seja:

$$L_S(h) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{h(x_i) \neq y_i\},$$

onde $\mathbb{1}\{h(x_i) \neq y_i\}$ é uma função indicadora que vale 1 se $h(x_i) \neq y_i$ e 0 caso contrário.

Meta do Aprendizado

- O objetivo é encontrar uma hipótese $h : \mathcal{X} \to \mathcal{Y}$, com $h \in \mathcal{H}$ que minimize aproximadamente o **erro verdadeiro** $L_D(h)$.
- Como não conhecemos D, usamos $L_S(h)$ como uma aproximação, mas devemos garantir que minimizando o risco empírico, também minimizamos (provavelmente e aproximadamente) o risco verdadeiro.

Aprendizado PAC

Definição 3.3 (Agnostic PAC Learnability): Uma classe de hipóteses $\mathcal{H}$ é agnosticamente PAC se existir uma função $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ e um algoritmo de aprendizado com a propriedade de que, para todo $\epsilon, \delta \in (0,1)$ e para toda distribuição $\mathcal{D}$ sobre $\mathcal{X} \times \mathcal{Y}$, ao executar o algoritmo de aprendizado em $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ exemplos i.i.d. gerados de acordo com $\mathcal{D}$, o algoritmo retorna uma hipótese h tal que, com probabilidade de pelo menos $1 - \delta$ (sobre a escolha dos m exemplos de treinamento):

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon.$$

Importante, a amostra é finita!

Desvio entre Erro Verdadeiro e Empírico

- Lembre-se que $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$ e que $L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i)$.
- Como cada z_i é amostrado i.i.d. de $\mathcal{D}$, o valor esperado da variável aleatória $\ell(h,z_i)$ é $L_{\mathcal{D}}(h)$.
- Pela linearidade da esperança, segue que $L_{\mathcal{D}}(h)$ é também o valor esperado de $L_S(h)$.
- Portanto, a quantidade $|L_{\mathcal{D}}(h) - L_S(h)|$ é o desvio da variável aleatória $L_S(h)$ em relação à sua esperança.
- Precisamos mostrar que $L_S(h)$ está **concentrado** em torno de seu valor esperado.

Lei dos Grandes Números

- Um fato estatístico básico, a **lei dos grandes números**, afirma que, à medida que $m \to \infty$, as médias empíricas convergem para sua esperança verdadeira.

$$\lim_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$$

- Isso é verdade para $L_S(h)$, já que ele é a média empírica de m variáveis aleatórias i.i.d.
- No entanto, a lei dos grandes números é apenas um resultado assintótico, e não nos informa sobre o desvio entre o erro empírico e seu valor verdadeiro para um tamanho de amostra finito.

Desigualdade de Hoeffding

- Em vez disso, usaremos uma **desigualdade de concentração de medida**, a **Desigualdade de Hoeffding**, que quantifica o desvio entre as médias empíricas e seu valor esperado.

Lema 4.5 (Desigualdade de Hoeffding):

- Seja $\theta_1, \ldots, \theta_m$ uma sequência de variáveis aleatórias i.i.d., e suponha que, para todo i, $\mathbb{E}[\theta_i] = \mu$ e $P[a \leq \theta_i \leq b] = 1$.
- Então, para qualquer $\epsilon > 0$, temos:

$$P\left(\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \epsilon\right) \leq 2\exp\left(\frac{-2m\epsilon^2}{(b-a)^2}\right).$$

Conjunto finito de hipóteses é PAC!

Corolário: Seja $\mathcal{H}$ uma classe de hipóteses finita. Então, $\mathcal{H}$ possui a propriedade de **convergência uniforme** com complexidade de amostra dada por:

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

- Além disso, a classe é **PAC aprendível de forma agnóstica** usando o algoritmo ERM, com a complexidade de amostra:

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

Sem Almoço Grátis

Teorema: Não existe um modelo universal. Isto é, nenhum algoritmo de aprendizado pode ser bem-sucedido em todas as tarefas de aprendizado, conforme formalizado no teorema a seguir:

- Seja A um algoritmo de aprendizado para a tarefa de classificação binária com respeito à perda 0-1 sobre um domínio $\mathcal{X}$.
- Seja m qualquer número menor que $|\mathcal{X}|/2$, representando o tamanho do conjunto de treinamento.
- Então, existe uma distribuição $\mathcal{D}$ sobre $\mathcal{X} \times \{0, 1\}$ tal que:
  1. Existe uma função $f : \mathcal{X} \to \{0, 1\}$ tal que $L_{\mathcal{D}}(f) = 0$.
  2. Com probabilidade de pelo menos $1/7$ sobre a escolha de $S \sim \mathcal{D}^m$, temos que:

$$L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}.$$

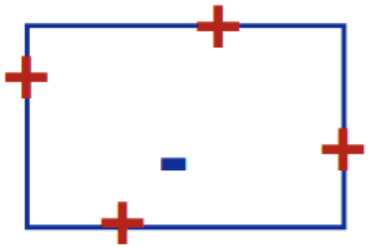
Dimensão VC

- A finitude de $\mathcal{H}$ é uma condição suficiente, mas não necessária para a classe ser PAC.
- A **Dimensão VC** é uma propriedade que caracteriza corretamente se uma classe de hipóteses $\mathcal{H}$ é PAC.
- Através da Dimensão VC, podemos entender como algumas classes de hipóteses infinitas podem ser aprendíveis, e como isso afeta a complexidade de amostra.
- De forma simplificada, a Dimensão VC mede o maior conjunto de pontos que uma classe de hipóteses $\mathcal{H}$ pode fragmentar, e isso tem implicações diretas na quantidade de dados necessária para o aprendizado PAC.

Dimensão VC

Por exemplo, suponha que nossa classe de classificadores consiste de retângulos paralelos aos eixos.

(a)



(b)

Dimensão VC

Desafio: Suponha que nossa classe de classificadores consiste em conjuntos convexos. A dimensão VC é finita ou infinita?

Teorema sobr PAC e VC

Teorema: Seja $\mathcal{H}$ uma classe de hipóteses de funções de um domínio $\mathcal{X}$ para $\{0, 1\}$ e seja a função de perda a perda 0-1. Então, as seguintes afirmações são equivalentes:

1. $\mathcal{H}$ é PAC aprendível.
2. $\mathcal{H}$ possui uma **dimensão VC** finita.

A prova disso é muito bonitinha =)

Mais um teorema legal

Teorema: Seja $\mathcal{H}$ uma classe de hipóteses com dimensão VC d. Então, para toda distribuição $\mathcal{D}$ e todo $\delta \in (0,1)$, com probabilidade de pelo menos $1-\delta$ sobre a escolha de $S \sim \mathcal{D}^m$, temos:

$$|L_S(h) - L_{\mathcal{D}}(h)| \leq \frac{4 + \sqrt{d \log\left(\frac{2em}{d}\right)}}{\sqrt{2m\delta}}.$$

Preditores Lineares

- Preditores lineares são uma das famílias mais úteis de classes de hipóteses devido à sua eficiência de aprendizado e facilidade de interpretação.
- As classes de preditores lineares incluem **halfspaces** (meios-hiperplanos), preditores de regressão linear e preditores de regressão logística.
- Aprender preditores lineares pode ser feito de forma eficiente usando algoritmos como programação linear e o algoritmo Perceptron (para halfspaces) e o algoritmo de mínimos quadrados (para regressão linear).

Funções Afins

- A classe de funções afins é definida como:

$$\mathcal{L}_d = \{h_{w,b} : w \in \mathbb{R}^d, b \in \mathbb{R}\},$$

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

- Essa classe representa funções lineares com um termo de viés b.

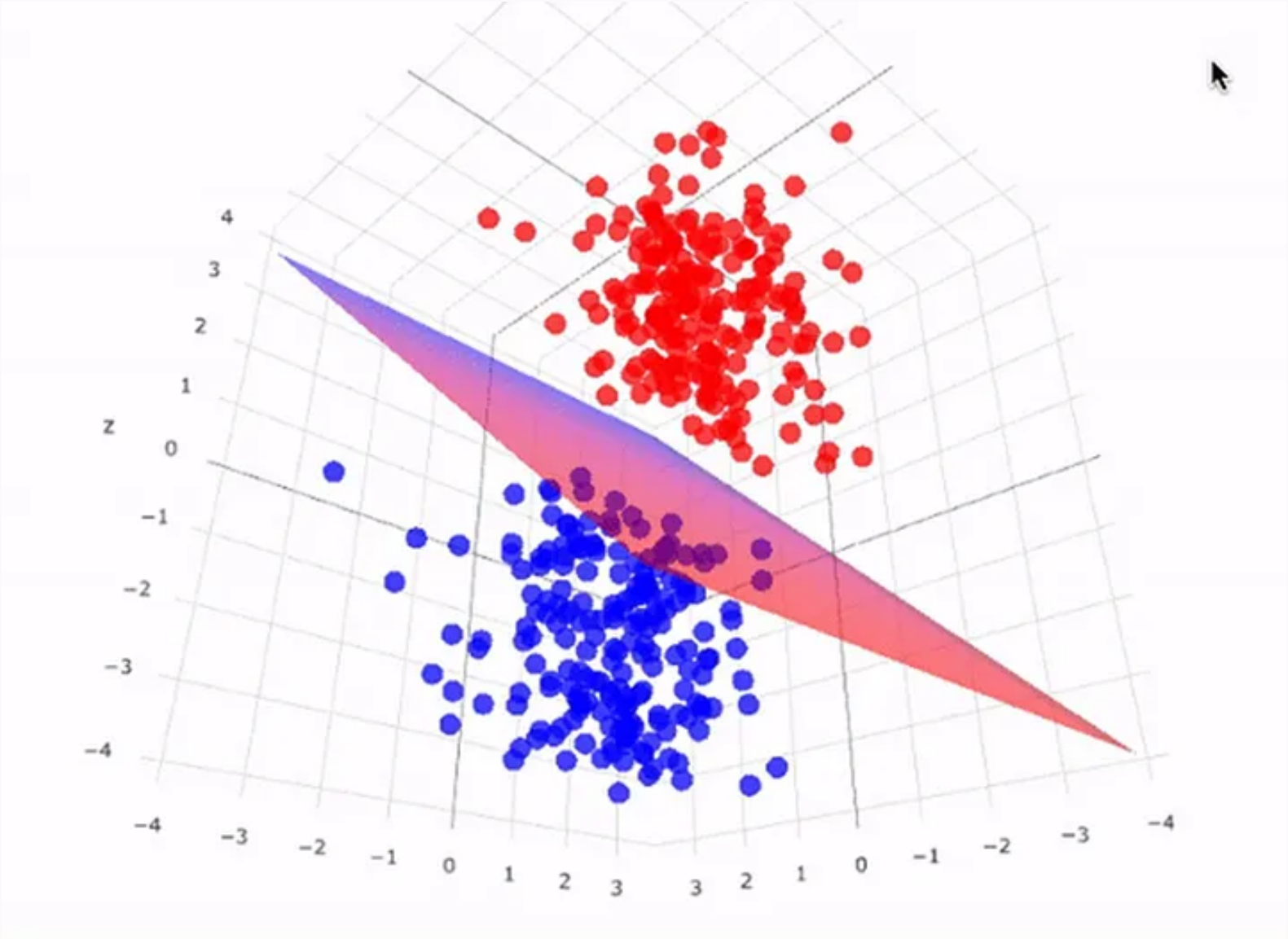
Halfspaces para Classificação Binária

- A classe de **halfspaces** é usada para problemas de classificação binária, onde $\mathcal{X} = \mathbb{R}^d$ e $\mathcal{Y} = \{-1, +1\}$.
- Cada hipótese em $\mathcal{HS}_d$ é parametrizada por $w \in \mathbb{R}^d$ e $b \in \mathbb{R}$ e devolve o rótulo $sign(\langle w, x \rangle + b)$.
- Geometricamente, cada halfspace forma um hiperplano perpendicular ao vetor w e divide o espaço em duas regiões: uma rotulada positivamente e outra negativamente.

ERM para Halfspaces

- A classe de halfspaces possui uma dimensão VC de $d + 1$.
- Podemos aprender halfspaces usando o paradigma ERM, desde que o tamanho da amostra seja $\Omega\left(\frac{d+\log(1/\delta)}{\epsilon}\right)$.
- Discussões sobre como implementar um procedimento ERM para halfspaces serão abordadas em seções posteriores.

ERM para Halfspaces



AdaBoost

- O AdaBoost constrói um preditor forte combinando múltiplas hipóteses fracas da classe base $\mathcal{B}$ (por exemplo, hiperplanos paralelos aos eixos) em uma composição linear de predições.
- A saída do AdaBoost após T iterações pertence à classe de funções:

$$\mathcal{L}(\mathcal{B}, T) = \left\{ x \mapsto \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : w \in \mathbb{R}^T, \forall t, h_t \in \mathcal{B} \right\}.$$

- Cada função $h \in \mathcal{L}(\mathcal{B}, T)$ é parametrizada por T hipóteses base da classe $\mathcal{B}$ e por um vetor de pesos $w \in \mathbb{R}^T$.
- A predição para uma instância x é obtida aplicando as T hipóteses base para construir o vetor $\psi(x) = (h_1(x), \ldots, h_T(x)) \in \mathbb{R}^T$ e, em seguida, aplicando um halfspace homogêneo definido por w sobre $\psi(x)$.

Análise da Dimensão VC de $\mathcal{L}(\mathcal{B}, T)$

- A dimensão VC da classe $\mathcal{L}(\mathcal{B}, T)$ está relacionada à dimensão VC da classe base $\mathcal{B}$ e ao número de iterações T.
- O AdaBoost tem a propriedade de que a **dimensão VC** de $\mathcal{L}(\mathcal{B}, T)$ é, aproximadamente, limitada por T vezes a dimensão VC de $\mathcal{B}$, ou seja:

$$VC(\mathcal{L}(\mathcal{B}, T)) \leq T \cdot VC(\mathcal{B}) + \mathcal{O}(\log T).$$

- Isso implica que o **erro de estimação** do AdaBoost cresce linearmente com T.
- Por outro lado, o **risco empírico** do AdaBoost diminui com o aumento de T, permitindo que o parâmetro T seja utilizado para controlar o trade-off entre **viés** e **complexidade**.

Conclusão

Principais Conceitos:

- **Preditores Lineares**: São eficientes e fáceis de interpretar. Incluem halfspaces e regressão linear, aprendidos por algoritmos como o Perceptron.
- **AdaBoost**: Combina preditores fracos para formar um preditor forte, ajustando pesos a cada iteração e diminuindo o risco empírico com mais iterações (T).
- **Dimensão VC**: A dimensão VC do AdaBoost cresce com T e a dimensão da classe base $\mathcal{B}$. O trade-off entre erro de estimação e risco empírico é controlado por T.
- **Combinações Lineares**: O AdaBoost realiza predições aplicando um halfspace sobre as predições das hipóteses base.

Conclusão

- **Aprendizado Estatístico**: O objetivo é encontrar um preditor que generalize bem com base em dados de treinamento, minimizando o erro verdadeiro.
- **ERM (Minimização de Risco Empírico)**: Busca um preditor que minimize o erro nos dados de treinamento, com viés indutivo para evitar overfitting.
- **Dimensão VC**: Caracteriza a complexidade de uma classe de hipóteses e determina a quantidade de amostras necessárias para garantir a aprendibilidade PAC.
- **Preditores Lineares e AdaBoost**: Preditores lineares são eficientes e fáceis de interpretar. O AdaBoost combina preditores fracos para formar um preditor forte, com controle de trade-offs entre viés e complexidade.