

Introdução a Aprendizado de Máquina

Prof. Dr. Clodoaldo A M Lima

Escola de Artes, Ciências e Humanidades - USP

19 de agosto de 2025

Agenda

1 Apresentação

2 Pesquisa em Andamento

- Reconhecimento biométrico facial
- Reconhecimento biométrico baseado em EEG
- Técnicas para Envelhecimento usando GANs

3 Conceitos Básicos

- Tipos de Raciocínio
- Paradigmas de Aprendizado de Máquina
- Classificação
- Métricas de Avaliação
- Métodos para Estimar o Erro Verdadeiro
- Regressão
- Problema de Aprendizado de Máquina
- Erro verdadeiro de uma hipótese

Definição

Biometria é uma palavra de origem grega que significa 'medida da vida' e consiste em realizar medidas de traços humanos, que podem ser tanto físicos, biológicos quanto comportamentais

Modalidades Biométricas

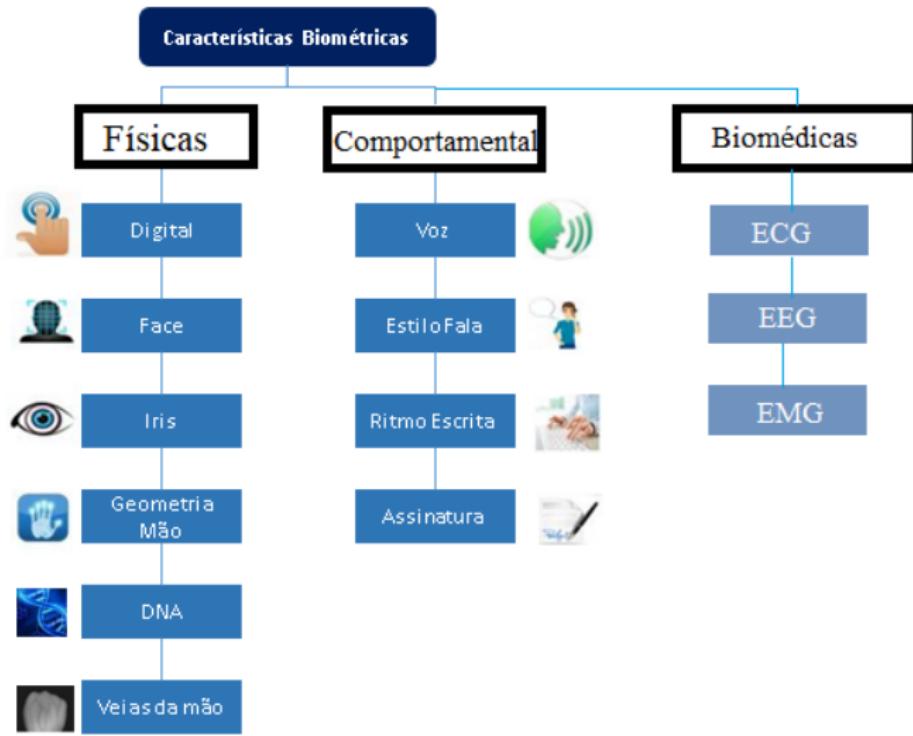
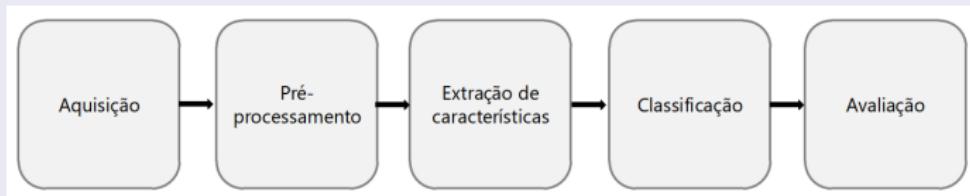


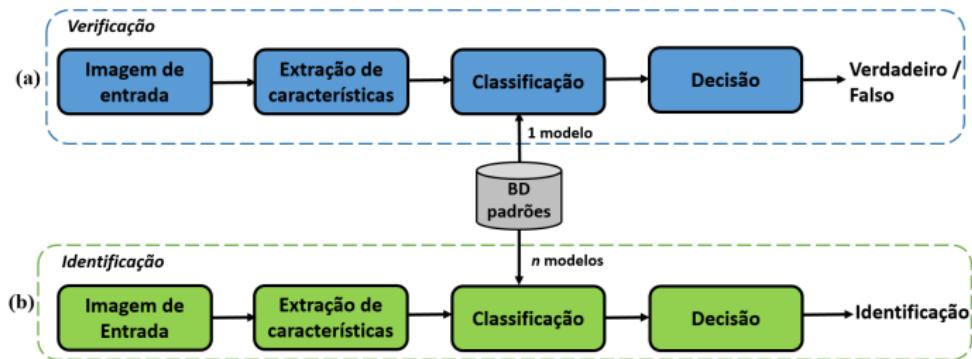
Figura: Modalidades biométricas).

Reconhecimento biométrico Tradicional

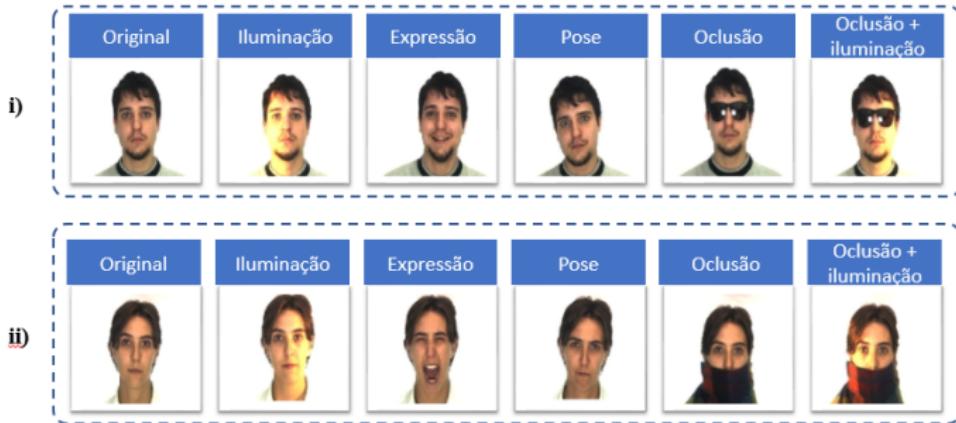
Diagrama de blocos



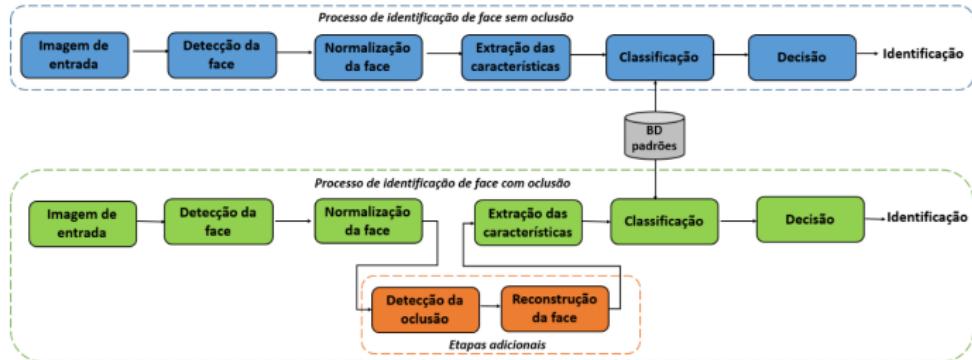
Verificação versus Identificação



Tipos de variações

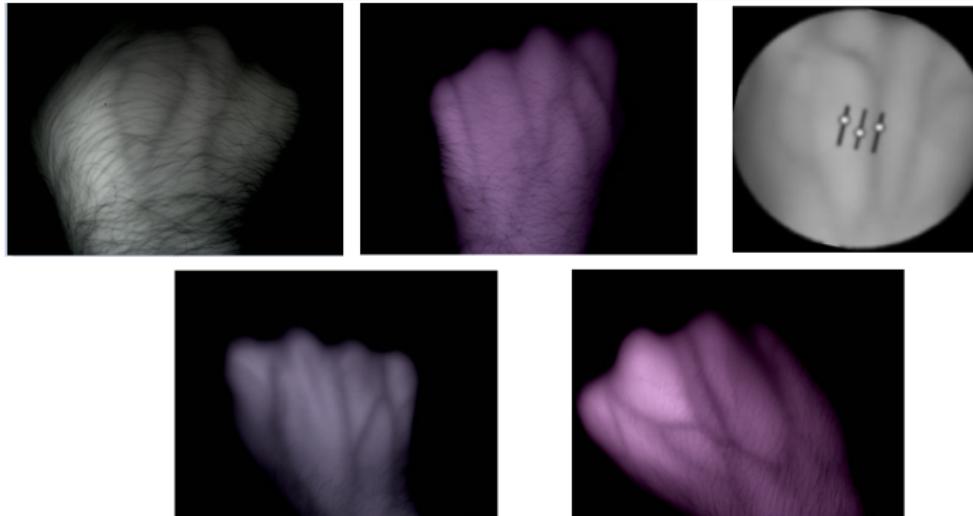


Reconhecimento facial com oclusão



Reconhecimento facial com oclusão

Existem diferentes desafios que são encontrados em imagens de veias do dorso da mão:



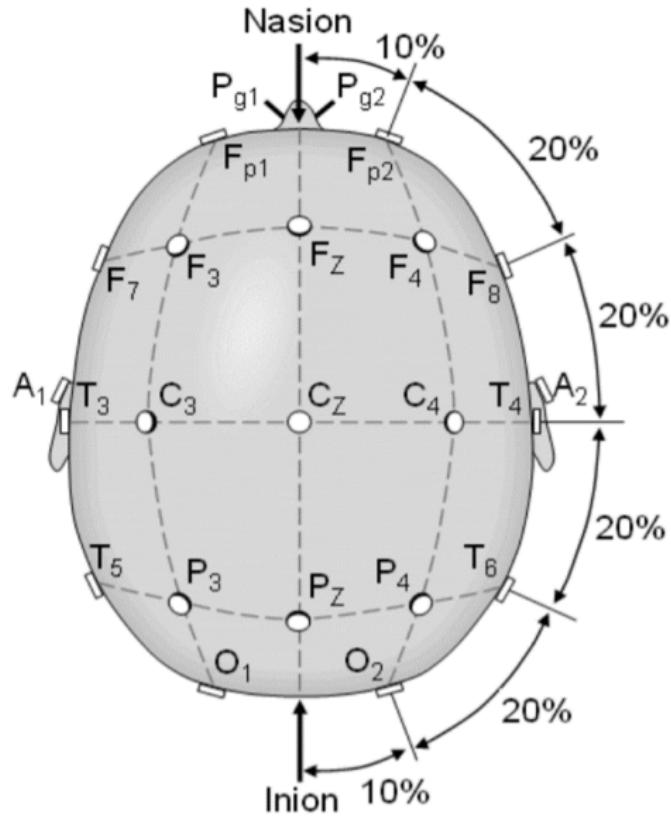
Forma de Captura

Composição do aparato:

- Eletrodos de Prata/Prata-Clorada com o diâmetro de 1 à 3 mm
- Amplificadores
- Dispositivo para aquisição de dados
- Sistema de processamento



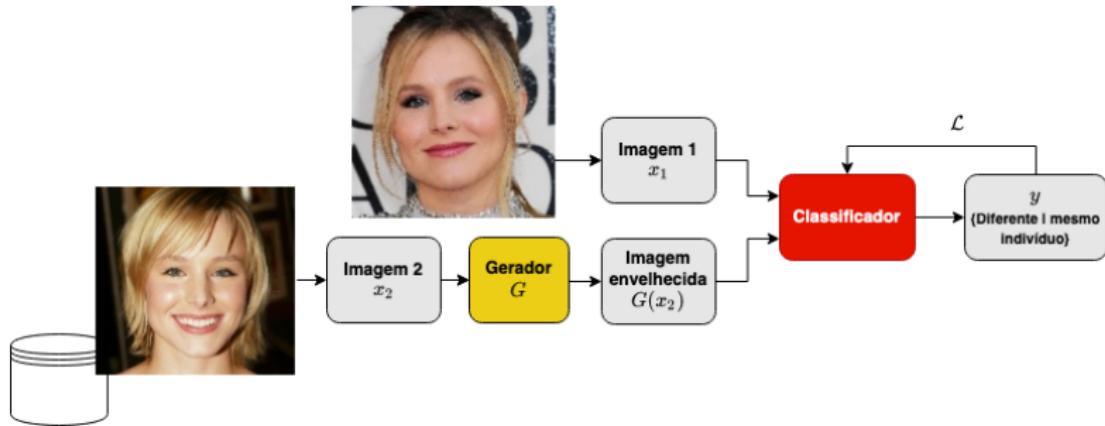
Sistema 10-20



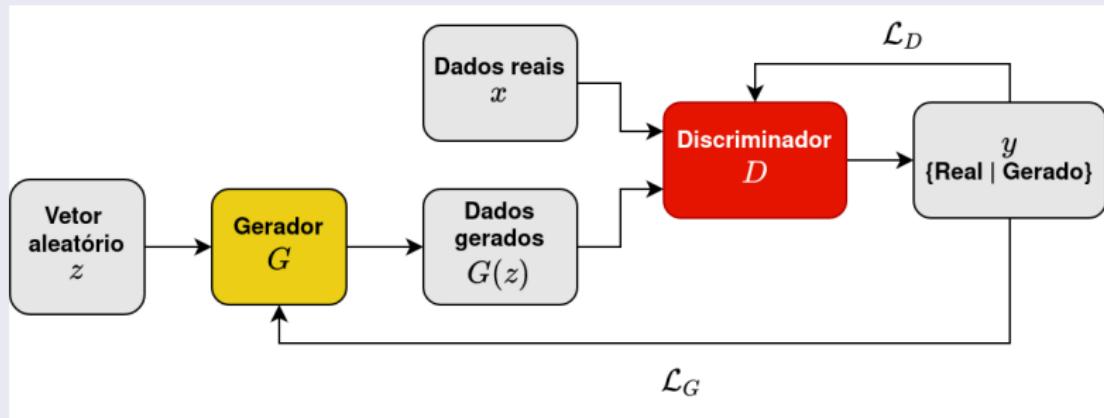
Envelhecimento facial

Os métodos computacionais de envelhecimento facial tem como objetivo gerar uma face envelhecida mantendo as características individuais.

Envelhecimento



Redes Neurais Generativas - GANs



Equação Clássica de Treinamento de uma GAN

Função Objetivo:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

- D é o discriminador
- G é o gerador
- $p_{\text{data}}(x)$ é a distribuição real dos dados
- $p_z(z)$ é a distribuição do ruído de entrada

Exemplo de Código GAN

```
1  for epoch in range(epochs):
2      for real in dataloader:
3          1) Treina o Discriminador
4          optim_D.zero_grad()
5          z = torch.randn(batch_size, z_dim)
6          fake = G(z)
7          loss_D = -torch.mean(torch.log(D(real)))
8              + torch.log(1 - D(fake.detach())))
9          loss_D.backward()
10         optim_D.step()
11         2) Treina o Gerador
12         optim_G.zero_grad()
13         z = torch.randn(batch_size, z_dim)
14         fake = G(z)
15         loss_G = -torch.mean(torch.log(D(fake)))
16         loss_G.backward()
17         optim_G.step()
```

Exemplo de Código GAN

Passos principais:

- Amostrar lote real do dataset.
- Gerar lote falso com G .
- Atualizar D para distinguir real e falso.
- Atualizar G para enganar D .

Observação: Essa implementação é simplificada para fins didáticos.

Evolução dos resultados das GANs em termos de resolução.



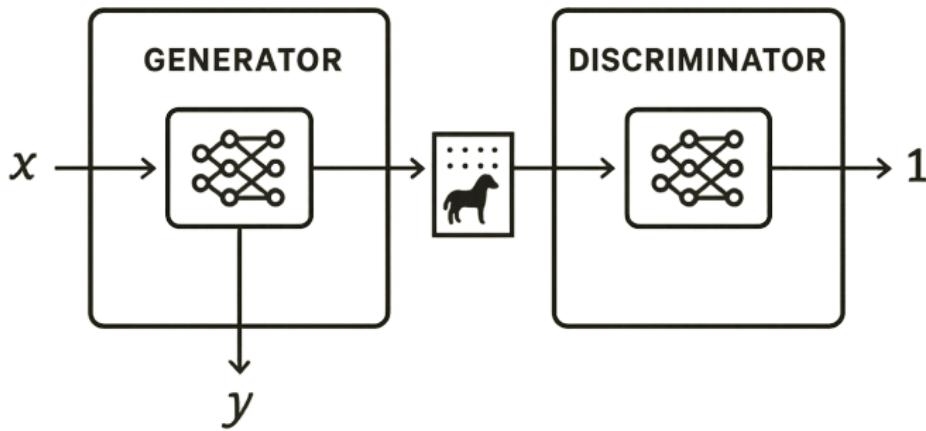
Redes adverárias geradoras condicionais, cGANs



O que é uma cGAN?

- Extensão das GANs que permite **controlar o conteúdo gerado** através de um rótulo ou vetor de condição.
- Útil para tarefas onde queremos **especificar a classe** ou características da saída.
- Exemplo: gerar dígitos específicos do MNIST, imagens de roupas por categoria, etc.

CONDITIONAL GAN



- O Gerador recebe ruído z e rótulo y (embedding) e produz $G(z|y)$.
- O Discriminador recebe imagem e o mesmo rótulo y para decidir real/falso.

Arquitetura (visão geral)

Gerador (Convolucional condicional)

- Entrada: vetor z (ruído) + embedding da classe.
- Usa ConvTranspose2d para subir a resolução até a imagem final.

Discriminador (condicional)

- Entrada: imagem concatenada com mapa da classe (canal extra) ou uso de projection discriminator.
- Saída: logit real/falso (BCEWithLogitsLoss ou alternativas).

Código PyTorch – Generator (DCGAN condicional)

```
1 import torch.nn as nn
2 import torch
3
4 class Generator(nn.Module):
5     def __init__(self, z_dim=100, num_classes=10, embed_dim=50,
6                  img_channels=1):
7         super().__init__()
8         self.label_emb = nn.Embedding(num_classes, embed_dim)
9         input_dim = z_dim + embed_dim # canais de entrada para
10            ConvTranspose2d
11         self.net = nn.Sequential(
12             # input is (input_dim) x 1 x 1
13             nn.ConvTranspose2d(input_dim, 256, 4, 1, 0,
14                               bias=False),
15             nn.BatchNorm2d(256),
16             nn.ReLU(True),
17             # state size. (256) x 4 x 4
18             nn.ConvTranspose2d(256, 128, 4, 2, 1, bias=False),
19             nn.BatchNorm2d(128),
20             nn.ReLU(True),
21             # state size. (128) x 8 x 8
22             nn.ConvTranspose2d(128, 64, 4, 2, 1, bias=False),
23             nn.BatchNorm2d(64),
24             nn.ReLU(True),
```

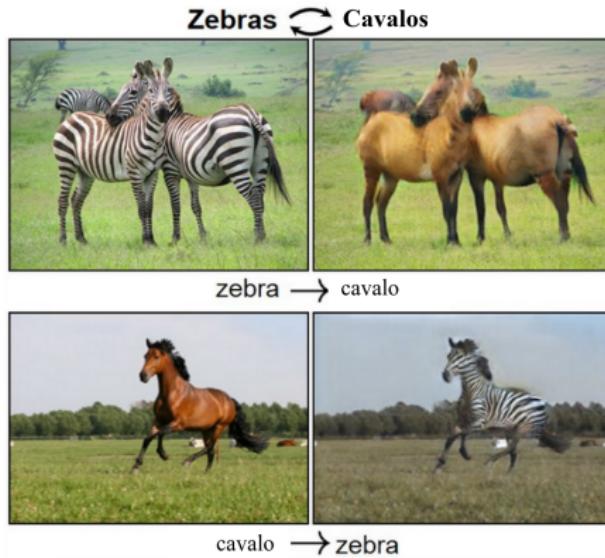
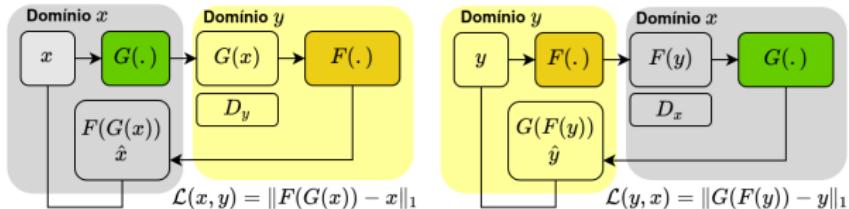
Código PyTorch – Discriminador (condicional)

```
1  class Discriminator(nn.Module):
2      def __init__(self, num_classes=10, img_channels=1,
3                   img_size=32):
4          super().__init__()
5          # vamos concatenar mapa de rótulo como canais extras
6          self.label_emb = nn.Embedding(num_classes, img_channels)
7          self.net = nn.Sequential(
8              nn.Conv2d(img_channels*2, 64, 4, 2, 1, bias=False),
9              nn.LeakyReLU(0.2, inplace=True),
10             nn.Conv2d(64, 128, 4, 2, 1, bias=False),
11             nn.BatchNorm2d(128),
12             nn.LeakyReLU(0.2, inplace=True),
13             nn.Conv2d(128, 1, 4, 1, 0, bias=False),
14         )
15
16         def forward(self, img, labels):
17             # img: (B, C, H, W)
18             le = self.label_emb(labels).unsqueeze(2).unsqueeze(3)
19             le = le.repeat(1, 1, img.size(2), img.size(3))
20             x = torch.cat([img, le], dim=1)
21             return self.net(x).view(-1)
```

Treino (esboço simplificado)

```
1 # --- esboço de loop de treino ---
2 criterion = nn.BCEWithLogitsLoss()
3 optim_G = torch.optim.Adam(generator.parameters(), lr=2e-4,
    betas=(0.5,0.999))
4 optim_D = torch.optim.Adam(discriminator.parameters(), lr=2e-4,
    betas=(0.5,0.999))
5
6 for epoch in range(epochs):
7     for imgs, labels in dataloader:
8         imgs, labels = imgs.to(device), labels.to(device)
9         B = imgs.size(0)
10
11         # Train D: reais
12         optim_D.zero_grad()
13         real_logits = discriminator(imgs, labels)
14         loss_real = criterion(real_logits, torch.ones(B,
15                               device=device))
16
17         # Train D: falsos
18         z = torch.randn(B, z_dim, device=device)
19         fake_labels = torch.randint(0, num_classes, (B,), 
20                                     device=device)
21         fake_imgs = generator(z, fake_labels)
22         fake_logits = discriminator(fake_imgs.detach(),
```

CycleGAN



Conceitos básicos

Prof. Dr. Clodoaldo A M Lima

Escola de Artes, Ciências e Humanidades - USP

19 de agosto de 2025

Introdução

Objetivo de aprendizado de Máquina

O objetivo do aprendizado de máquina é construir modelos computacionais que podem adaptar-se e aprender a partir da experiência (MITCHELL,1997).

Aprendizado Indutivo - Segundo MITCHELL (1997)

Um programa de computador aprende a partir de um elenco de experiências E, relacionadas a uma classe de tarefas T e dispondo de uma medida de desempenho M, se seu desempenho medido por M junto á tarefa T melhora com o elenco de experiências E.

Exemplo

No contexto de redes neurais artificiais, o seu processo de treinamento pode, então, ser caracterizado como aprendizado indutivo, sendo que o uso posterior da rede neural treinada para classificação, regressão ou agrupamento de dados é geralmente denominado de processo de inferência dedutiva.

Tipos de Racioncinio

Raciocínio Indutivo

- Conhece: $p(a,b)$, $p(a,d)$, $p(d,e)$, $p(d,g)$, $p(e,f)$
- Observa: $a(a,e)$ e $a(d,f)$
- Aprende: $p(X,Y) \cap a(Y,Z) \rightarrow p(X,Z)$

Raciocínio Abdutivo

- Conhece: $p(a,b)$, $p(a,d)$, $p(d,e)$, $p(d,g)$, $p(e,f)$, $p(X,Y) \cap p(Y,Z) \rightarrow a(X,Z)$
- Observa: $a(a,c)$
- Explica: $p(b,c)$ ou $p(d,c)$

Raciocínio Dedutivo

- Conhece: $p(a,b)$, $p(a,d)$, $p(b,c)$, $p(d,e)$, $p(d,g)$, $p(e,f)$, $p(X,Y) \cap p(Y,Z) \rightarrow a(X,Z)$
- Conclui: $a(a,c)$, $a(a,e)$, $a(a,g)$ e $a(d,f)$

Aprendizado de Máquina

Em termos práticos, algoritmos de aprendizado de máquina têm como objetivo descobrir o relacionamento entre as variáveis de um sistema (entrada/saída) a partir de dados amostrados (CHERKASSKY & MULIER, 2007).

Sendo assim, eles não são necessários quando os relacionamentos entre todas as variáveis do problema (entradas/saída) são completamente compreendidos. Este definitivamente não é o caso de muitos dos problemas reais com os quais nos defrontamos em nosso dia-a-dia.

Há três paradigmas principais em aprendizado de máquina

- Aprendizado supervisionado, que será o centro de atenção deste curso;
- Aprendizado por reforço, que não será abordado formalmente, pois foge ao escopo do curso;
- Aprendizado não supervisionado, abordado especificamente em alguns tópicos



Definição

Modelo aprende a partir de dados rotulados, mapeando entrada para saída. É um tipo de aprendizado de máquina em que um algoritmo é treinado usando um conjunto de dados rotulados, onde cada exemplo de entrada possui a saída correta correspondente

Exemplo e Aplicações

Exemplo: Prever preço de casas.

Aplicações: Classificação, regressão.

Definição

Descobre padrões ocultos em dados não rotulados. É um tipo de aprendizado de máquina em que o modelo aprende com dados não rotulados, identificando padrões e estruturas ocultas sem orientação humana

Exemplo e Aplicações

Exemplo: Agrupar clientes por hábitos.

Aplicações: Clustering, redução de dimensionalidade.

Definição

Poucos dados rotulados combinados com muitos não rotulados. Ele utiliza tanto dados rotulados quanto dados não rotulados para treinar modelos, o que é especialmente útil quando se tem uma quantidade limitada de dados rotulados.

Exemplo e Aplicações

Exemplo: Classificação de páginas web.

Aplicações: Reconhecimento de fala, visão computacional.

Definição

Agente aprende interagindo com o ambiente e recebendo recompensas. O agente experimenta o ambiente, recebe feedback na forma de recompensas ou penalidades e, com base nesse feedback, aprende a tomar decisões ótimas.

Exemplo e Aplicações

Exemplo: Treinar robô a andar.

Aplicações: Robótica, jogos.



Definição

Modelo gera rótulos internamente a partir de dados brutos. Aprender representações úteis dos dados sem precisar de rótulos humanos. Etapas típicas:

- 1) Criar um pré-treinamento com tarefa "falsa" gerada a partir dos dados brutos.
- 2) Depois, fazer fine-tuning com poucos dados rotulados para a tarefa final.

Exemplo e Aplicações

Exemplo: Prever palavras ocultas (BERT).

Aplicações: NLP, visão computacional.

Definição

Aprende continuamente com dados novos. O modelo aprende de forma contínua, recebendo dados em sequência e ajustando seus parâmetros gradualmente.

Exemplo e Aplicações

Exemplo: Sistemas de recomendação em tempo real.

Aplicações: Streaming, personalização.

Definição

Treinar para várias tarefas relacionadas. É uma abordagem onde um único modelo é treinado para resolver várias tarefas ao mesmo tempo.

Exemplo e Aplicações

Exemplo: Detecção e segmentação.

Aplicações: Visão, NLP multitarefa.

Definição

Aprender a aprender com poucos dados. É como treinar um atleta não só para um esporte, mas para que ele aprenda qualquer esporte novo de forma rápida.

Exemplo e Aplicações

Exemplo: Assistente de voz adaptável.

Aplicações: Personalização, robótica adaptativa.

Definição

Reaproveitar conhecimento de modelos pré-treinados. Técnica onde um modelo treinado em uma tarefa-fonte é reutilizado, total ou parcialmente, para resolver uma tarefa-alvo. A ideia é aproveitar o conhecimento já aprendido, reduzindo a necessidade de muitos dados ou treinamento extenso na tarefa-alvo.

Exemplo e Aplicações

Exemplo: ImageNet - imagens médicas.

Aplicações: Visão, NLP.

Definição

Prever apenas para um conjunto de dados de teste conhecido. É um tipo de aprendizado onde o modelo não tenta aprender uma função geral para qualquer dado novo. Em vez disso, ele usa diretamente os dados de teste disponíveis no treinamento para prever apenas esses exemplos específicos

Exemplo e Aplicações

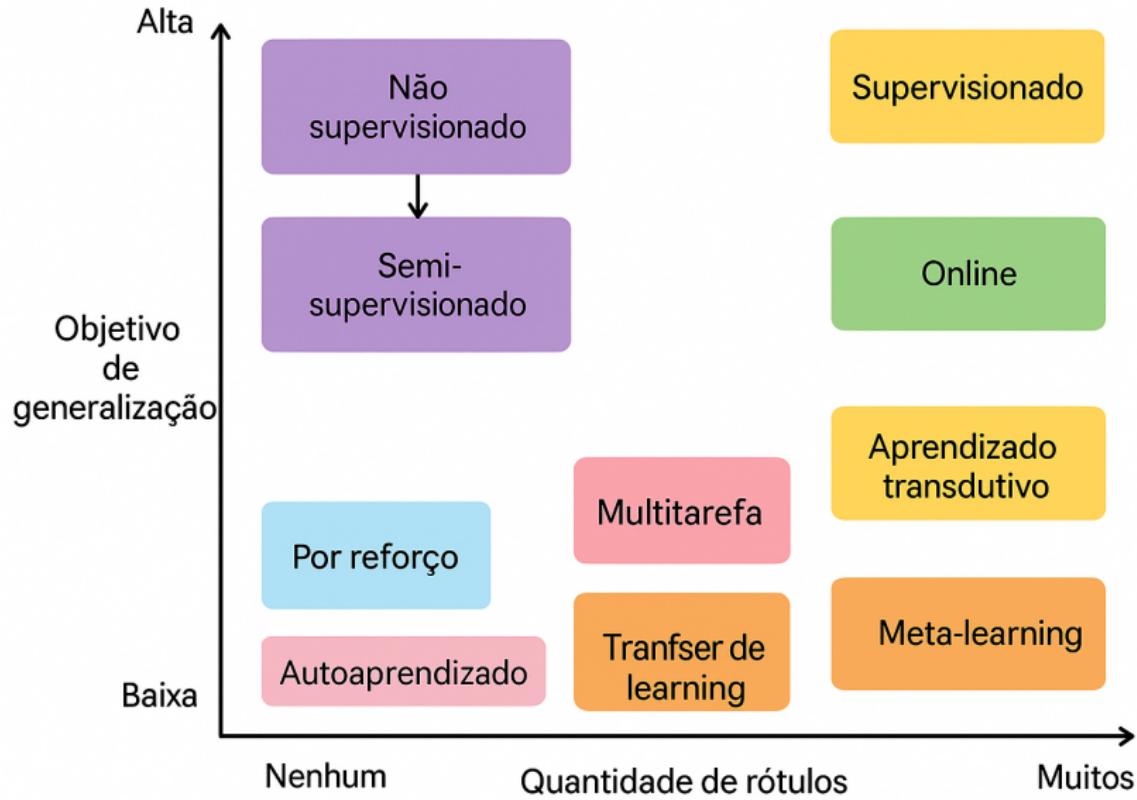
Exemplo: Classificação semi-supervisionada com teste fixo.

Aplicações: Competições de ML.

Resumo Comparativo

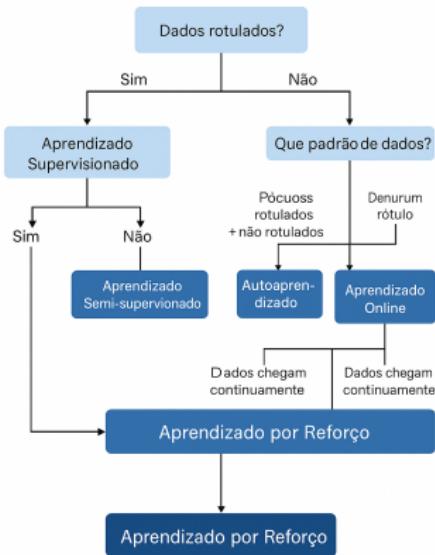
Tipos	Rótulos?	Objetivo	Exemplo
Supervisionado	Sim	EntradaSaída	Preço casas
Não supervisionado	Não	Encontrar padrões	Clustering clientes
Semi-supervisionado	Poucos	Usar dados não rotulados	Páginas web
Reforço	Não	Política ótima	Robô andar
Autosupervisionado	Não	Gerar rótulos	Prever palavra
Online	Depende	Aprender continuamente	Recomendação
Multitarefa	Sim	Várias tarefas	Detecção+Segmentação
Meta-learning	Sim	Aprender a aprender	Assistente adaptável
Transfer learning	Sim	Reaproveitar modelo	ImageNetmédicos
Transdutivo	Poucos	Prever para alvo fixo	Semi-supervisionado

Fluxograma para Escolha do Tipo de Aprendizado



Fluxograma para Escolha do Tipo de Aprendizado

Fluxograma para Escolha do Tipo de Aprendizado



Classificação

Dados

Oman Khan to Carlos
[show details](#) Jan 7 (6 days ago) [↳ Reply](#) x

sounds good
*ck

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos

Welcome to New Media Installation: Art that Learns
[show details](#) 3:15 PM (8 hours ago) [↳ Reply](#) x

Hi everyone.

Welcome to New Media Installation/Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Dorothy Hall C116, and will be Tues, Thu 1:30-2:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing 10615-instructors@cs.cmu.edu

Our source materials, syllabus, etc. are at
<http://artthatlearns.wordpress.com>

You will be sharing your projects there. So, please create an account on wordpress.com and send New Media Installation 10615-announce@cs.cmu.edu a note with the email you used for registering this account.

We are really excited to explore this new class with you.

Carlos & Osman

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only \$5.95 for shipping mfw rik [Open X](#) [show details](#) 0:52 PM (1 hour ago) [↳ Reply](#) x

Jequelyn Halley to mriemien, bcc: thehorney, bee: ang [show details](#) 0:52 PM (1 hour ago) [↳ Reply](#) x

*** Nature WeightLOSS Solution ***

Vital Acai is a natural WeightLOSS product that Enables people to lose weight and cleansing their bodies faster than most other products in the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

- * Rapid WeightLOSS
- * Increased Metabolism - Burn fat & calories easily!
- * Better Mood and Attitude
- * More Self Confidence
- * Detoxifies and Purify Your Body
- * Much More Energy!
- * Better Bowel Function
- * A Natural Colon Cleanse

<http://tinyurl.keweenawcon.com>
<http://tinyurl.keweenawcon.com>

- Classificação Binária
- Spam vs Não Spam

Classificação



Figure 1. Examples of training images for each face orientation

Reconhecimento Biométrico



Tabela Comparativa

Característica	Binária	Multiclasse
No de classes	2	≥ 3
Saída	1 probabilidade	Vetor de probabilidades
Ativação	Sigmoid	Softmax
Função de perda	BCE	Categorical CE
Decisão	Threshold	Argmax
Métricas	AUC, F1, Precision, Recall	F1 macro/micro, Matriz Confusão

Métricas de Avaliação para tarefa de classificação

Prof. Dr. Clodoaldo A M Lima

Escola de Artes, Ciências e Humanidades - USP

19 de agosto de 2025

Métricas de Avaliação

As métricas de avaliação permitem medir o desempenho de um modelo de classificação, identificando se ele está conseguindo prever corretamente as classes de interesse. São essenciais para:

- Comparar modelos diferentes.
- Ajustar hiperparâmetros.
- Identificar problemas como **desbalanceamento** de classes.

Erro Quadrático Médio (MSE) e Erro Absoluto Médio (MAE)

Erro Quadrático Médio (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

Erro Absoluto Médio (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

- MSE penaliza mais erros grandes devido à potência de 2.
- MAE é mais robusto a outliers.

Matriz de Confusão - Classificação Binária

Matriz de Confusão - Binária

Negativo Real			
		Positivo Previsto	Negativo Previsto
Positivo Real	50	10	
	5	35	

- **TP** Verdadeiros Positivos
- **TN** Verdadeiros Negativos
- **FP** Falsos Positivos
- **FN** Falsos Negativos

Matriz de Confusão - Multiclasse

Classe C	Classe B	Classe A	
Classe A	40	2	3
Classe B	4	35	1
Classe C	2	3	30

Cada célula (i, j) representa o número de instâncias da classe i previstas como classe j .

Métricas derivadas da matriz de confusão

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall (Sensibilidade)} = \frac{TP}{TP + FN} \quad (6)$$

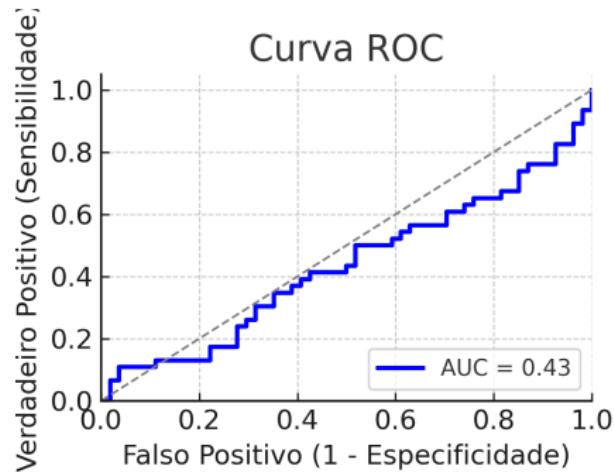
$$\text{Especificidade} = \frac{TN}{TN + FP} \quad (7)$$

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (8)$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (9)$$

- MCC varia de -1 (ruim) a 1 (perfeito), 0 indica classificação aleatória.

Curva ROC e AUC



- ROC: gráfico entre Sensibilidade e $(1 - \text{Especificidade})$.
- AUC: área sob a curva, mede a capacidade de separação.
- AUC = 0.5 Aleatório; AUC = 1 Perfeito.

Métodos para Estimar o Erro Verdadeiro

Prof. Dr. Clodoaldo A M Lima

Escola de Artes, Ciências e Humanidades - USP

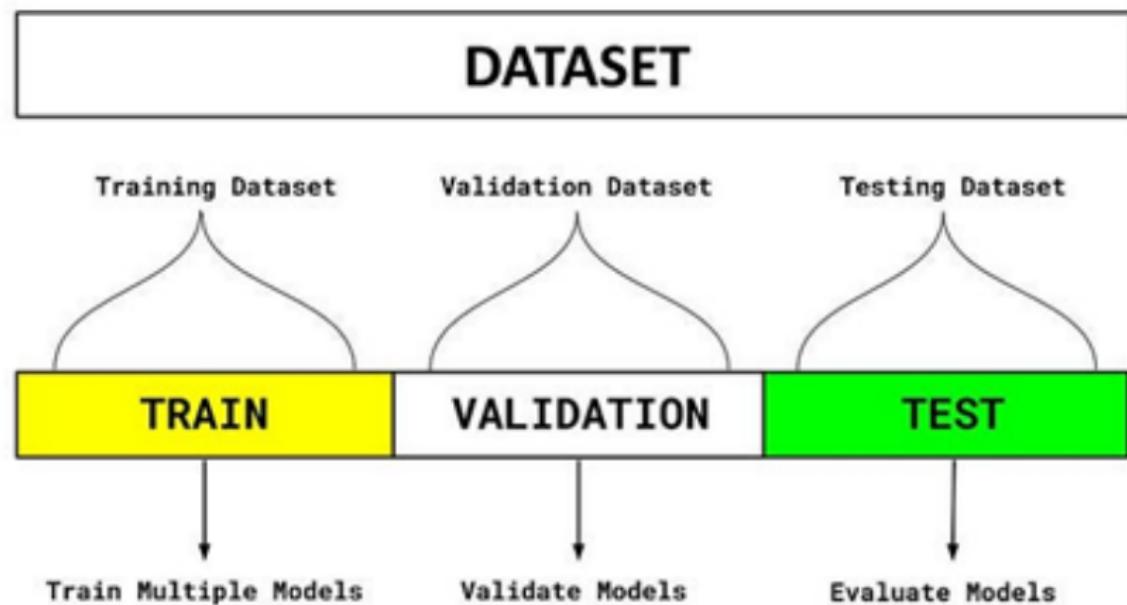
19 de agosto de 2025

Resubstitution

- Gera o classificador e testa a sua performance com o mesmo conjunto de dados
- Os desempenhos computados com este método são otimistas e tem grande bias
- Desde que o bias da resubstitution foi descoberto, os métodos de cross-validation são usados
- Muito simples, rápido, sem custo adicional de processamento.
- Simples e rápido, útil para prototipagem inicial.

Holdout

- Estratégia para teste de classificador que reserva um certo montante de dados para treino e o restante para teste (podendo ainda usar parte para validação).
- Comumente esta estratégia uma 1/3 dos dados para teste e o restante para treinamento, escolhido randomicamente.
- É interessante assegurar que a amostragem randômica seja feita de tal maneira que garanta que cada classe é apropriadamente representada tanto no conjunto de treinamento quanto no conjunto de teste. Este procedimento é chamado de estratificação (holdout estratificado).
- Também é útil, para amenizar tendências, repetir todo o processo de treino e teste várias vezes com diferentes amostragens randômicas (holdout repetitivo/iterativo).



Siga as etapas abaixo para usar o método de retenção para seleção de modelo:

- Divilde o conjunto de dados em três partes: conjunto de dados de treinamento, conjunto de dados de validação e conjunto de dados de teste.
- Treinar seu modelo de classificação, por exemplo, usando regressão logística, floresta aleatória e XGBoost.
- Altere as configurações dos hiperparâmetros para cada algoritmo mencionado na etapa 2 e crie vários modelos.
- Acalie o desempenho de cada um desses modelos (associando-os a cada um dos algoritmos).
- Selecione o modelo mais otimizado dentre aqueles testados no conjunto de dados de validação. O modelo mais otimizado será configurado com os hiperparâmetros mais otimizados.
- Por fim, no conjunto de dados de teste, teste o desempenho do modelo mais otimizado.

Treino (esboço simplificado)

```
1 # --- esboço de loop de treino ---
2 #Importing the dataset
3 from sklearn import datasets
4 from sklearn.model_selection import train_test_split#Then,
    loading the Boston Dataset
5 bhp = datasets.load_boston()#Finally, creating the Training and
    Test Split
6 X_train, X_test, y_train, y_test = train_test_split(bhp.data,
    bhp.target, random_state=42, test_size=0.3}
```

- Considere I classificadores, $I \ll N$, são induzidos de cada conjunto de treinamento
- O erro é a média dos erros dos classificadores medidos por conjuntos de treinamentos gerados aleatória e independentemente
- Pode produzir estimativas melhores que o holdout

K-fold Cross Validation

- Trata-se de uma estratégia para lidar com um montante de dados limitado.
- Nesta estratégia decide-se um numero fixos de folds, ou partições dos dados. Supondo que sejam usados três folds (3-fold cross validation):
 - o conjunto de dado é dividido em três partições de tamanhos aproximadamente iguais e, de maneira rotativa, cada uma delas é usada para teste enquanto as duas restantes são usadas para treinamento.
 - ou seja: use 2/3 para treinamento e 1/3 para teste e repita o processo três vezes, tal que, no fim, cada instância tenha sido usadas exatamente uma vez para teste.
 - se a estratificação é adotada, então o procedimento se chama 3-fold cross validation estratificado (aconselhável).
 - o padrão é executar o 10-fold cross validation, 10 vezes.
 - o erro final do classificador é a média dos erros obtidos em cada iteração da estratégia cross-validation

K-fold Cross Validation

Split	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

Training data Test data

K-fold Cross Validation - código

```
1 import numpy as np
2 from collections import defaultdict
3
4 def stratified_kfold_cross_validation(X, y, k):
5     """
6         Gera folds estratificados para K-Fold Cross Validation.
7         Parâmetros:
8             X (array-like): matriz de entrada (amostras x features).
9             y (array-like): matriz ou vetor de saída (amostras x 1).
10            k (int): número de folds.
11        Retorna:
12            folds (list): lista de tuplas (X_train, y_train,
13                           X_test, y_test)
14
15    X = np.array(X)
16    y = np.array(y)
17    n_samples = len(X)
18    if k <= 1 or k > n_samples:
19        raise ValueError("O número de folds deve ser entre 2 e
20                         o número total de amostras.")
```

K-fold Cross Validation - código

```
1      # Organizar índices por classe
2      class_indices = defaultdict(list)
3      for idx, label in enumerate(y):
4          class_indices[label].append(idx)
5      # Embaralhar os índices de cada classe
6      for label in class_indices:
7          np.random.shuffle(class_indices[label])
8      # Criar folds vazios
9      folds_idx = [[] for _ in range(k)]
```

K-fold Cross Validation - código

```
1      # Criar folds vazios
2      folds_idx = [[] for _ in range(k)]
3      # Distribuir as amostras de cada classe igualmente nos folds
4      for label, indices in class_indices.items():
5          for i, idx in enumerate(indices):
6              folds_idx[i % k].append(idx)
7      # Criar folds com dados de treino e teste
8      folds = []
9      for i in range(k):
10         test_idx = np.array(folds_idx[i])
11         train_idx = np.array([idx for fold in folds_idx if fold
12                           is not folds_idx[i] for idx in fold])
13         X_train, y_train = X[train_idx], y[train_idx]
14         X_test, y_test = X[test_idx], y[test_idx]
15
16         folds.append((X_train, y_train, X_test, y_test))
17
18     return folds
```

K-fold Cross Validation - código

```
1 # Exemplo de uso:  
2 if __name__ == "__main__":  
3     # Dados artificiais  
4     X = np.array([[i] for i in range(12)])  
5     y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2])    #  
          classes desbalanceadas  
6  
7     print("Conjunto de Dados")  
8     print(X.ravel())  
9     print(y.ravel())  
10    print("-" * 30)  
11  
12    folds = stratified_kfold_cross_validation(X, y, k=3)  
13  
14    for i, (X_train, y_train, X_test, y_test) in  
15        enumerate(folds):  
16        print(f"Fold {i + 1}")  
17        print("Treino:", X_train.ravel(), y_train)  
18        print("Teste:", X_test.ravel(), y_test)  
19        print("-" * 30)
```

Leave-one-out

- Leave-one-out cross-validation é um n-fold cross-validation, onde n é o número de instâncias no conjunto de dados.
- A avaliação é sobre a corretude de classificação da instância em teste um ou zero para sucesso ou falha, respectivamente.
- Os resultados de todas as n avaliações, uma para cada instância do conjunto de dados, são analisados via média, e tal média representa o erro final estimado.
- Motivações:
 - o maior número possível de dados é usado para treinamento em cada caso, o que aumenta as chance do classificador alcançar acuidade.
 - o procedimento é determinístico.
- Indicado para conjunto de dados pequenos.
- Não é possível aplicar qualquer procedimento de estratificação

Kx2 fold cross validation

- A validação cruzada 5x2 é um método estatístico para comparar o desempenho de dois modelos de aprendizado de máquina. Envolve a repetição de um procedimento de validação cruzada dupla cinco vezes. Esse processo ajuda a estimar o erro de generalização e a avaliar a significância estatística da diferença de desempenho entre os dois modelos.
- Validação cruzada dupla:
 - O conjunto de dados é dividido aleatoriamente em duas partes iguais (dobras), normalmente 50% cada.
 - Um fold é usado como conjunto de treinamento e a outro como conjunto de teste.
 - Os modelos são treinados no fold de treinamento e avaliados no fold de teste.
 - Os papéis dos dois folds são então trocados (o conjunto de treinamento se torna o conjunto de teste e vice-versa) e o processo é repetido.
 - Isso resulta em duas pontuações de desempenho (por exemplo, precisão, erro) para cada modelo.

Kx2 fold cross validation

- Repetição 5x2:
 - O processo de validação cruzada dupla é repetido cinco vezes, cada vez usando uma divisão aleatória diferente dos dados.
 - Isso significa que todo o processo de treinamento e avaliação dos modelos em dois folds é feito cinco vezes, gerando cinco pares de pontuações de desempenho para cada modelo

Outras variações

https://scikit-learn.org/stable/modules/cross_validation.html

Bootstrap

- Baseado em um procedimento estatístico de amostragem com reposição.
- Uma instância não é retirada do conjunto de dados original quando ela é escolhida para compor o conjunto de treinamento.
- Ou seja, a mesma instância pode ser selecionada várias vezes durante o procedimento de amostragem.
- As instâncias do conjunto original que não foram escolhidas para compor o conjunto de treinamento, compõem o conjunto de teste.
- O 0,632 bootstrap:
 - a probabilidade de uma instância ser escolhida é $1/n$. E de não ser escolhida é de $1 - (1/n)$.
 - Multiplicando essas probabilidades de acordo com o número de oportunidades de escolha (n), tem-se $(1(1/n))^n \approx e^{-1} = 0,368$ como a probabilidade de uma instância não ser escolhida. assim, para um conjunto de dados grande, o conjunto de testes conterá 36,8% de instâncias e o conjunto de treinamento, 63,2% delas.

- A medida de erro obtida é uma estimativa pessimista do erro verdadeiro porque o conjunto de treinamento, embora tenha tamanho n , contém somente 63% das instâncias, o que não é grande coisa se comparado com os 90% usados no 10-fold cross-validation.
- Para compensar isso, pode-se combinar o erro do conjunto de teste com o erro de resubstituição (estimativa otimista).
- O bootstrap combina da seguinte forma:

$$\text{erro} = 0,632 * \text{erro}_{\text{teste}} + 0,368 * \text{erro}_{\text{treinamento}}$$

- O procedimento deve ser repetido várias vezes, e uma média de erro final deve ser encontrada.

Bootstrap Normal vs. Estratificado

Critério	Bootstrap Normal	Bootstrap Estratificado
Seleção de amostras	Sorteio aleatório com reposição, sem restrições	Sorteio com reposição mantendo proporções originais das classes
Quando usar	Dados balanceados ou regressão	Dados desbalanceados em classificação
Vantagem	Mais simples de implementar	Representa melhor a distribuição original
Desvantagem	Pode gerar amostras com distribuição de classes distorcida	Mais complexo de implementar
Impacto na métrica	Pode ter alta variabilidade em classes raras	Reduz a variabilidade para classes pouco frequentes

Pseudocódigo do Bootstrap

Algoritmo

- ① Para cada iteração:
 - ① Gerar amostra de treino com reposição.
 - ② Definir conjunto de teste como amostras não selecionadas (*out-of-bag*).
 - ③ Treinar modelo no treino.
 - ④ Avaliar modelo no teste.
- ② Retornar estatísticas das métricas.

Código Python - Função Bootstrap

```
1 import numpy as np
2 from collections import defaultdict
3
4 def bootstrap_sampling(X, y, n_iterations=100, sample_size=None,
5                         model_fn=None, metric_fn=None,
6                         stratified=False):
7     X, y = np.array(X), np.array(y)
8     n_samples = len(X)
9     if sample_size is None:
10         sample_size = n_samples
11     if model_fn is None or metric_fn is None:
12         raise ValueError("Forneça model_fn e metric_fn.")
13
14     metrics = []
15     if stratified:
16         class_indices = defaultdict(list)
17         for idx, label in enumerate(y):
18             class_indices[label].append(idx)
```

Código Python - Função Bootstrap

```
1     for _ in range(n_iterations):
2         if stratified:
3             train_indices = []
4             for label, indices in class_indices.items():
5                 n_class_samples = int(round(sample_size *
6                     len(indices) / n_samples))
7                 chosen = np.random.choice(indices,
8                     size=n_class_samples, replace=True)
9                 train_indices.extend(chosen)
10            train_indices = np.array(train_indices)
11        else:
12            train_indices = np.random.randint(0, n_samples,
13                sample_size)
14        oob_indices = [idx for idx in range(n_samples) if idx
15                      not in train_indices]
16        if len(oob_indices) == 0:
17            continue
18        X_train, y_train = X[train_indices], y[train_indices]
19        X_test, y_test = X[oob_indices], y[oob_indices]
20        model = model_fn(X_train, y_train)
21        y_pred = model(X_test)
22        metrics.append(metric_fn(y_test, y_pred))
23    return metrics
```

Exemplo de uso - Classificação e Regressão

```
1 # Classificação (estratificado)
2 results_class = bootstrap_sampling(X_class, y_class,
3                                     n_iterations=200,
4                                     model_fn=train_centroid_classifier,
5                                     metric_fn=accuracy, stratified=True)
6
7 # Regressão (normal)
8 results_reg = bootstrap_sampling(X_reg, y_reg, n_iterations=200,
9                                   model_fn=train_linear_regression,
10                                  metric_fn=rmse, stratified=False)
11
12 print(f"Acurácia média: {np.mean(results_class):.3f}")
13 print(f"RMSE médio: {np.mean(results_reg):.3f}")
```

O que é um teste de hipótese?

- É um procedimento estatístico para decidir se uma afirmação sobre um parâmetro populacional é plausível.
- Hipótese nula (H_0): afirmação inicial a ser testada.
- Hipótese alternativa (H_1): afirmação contrária à nula.
- Escolhe-se um nível de significância (α), geralmente 5%.
- Calcula-se uma estatística de teste e compara-se com um valor crítico ou p -valor.

Etapas de um teste de hipótese

- ① Formular H_0 e H_1 .
- ② Escolher o teste estatístico apropriado.
- ③ Calcular a estatística de teste.
- ④ Determinar o p -valor.
- ⑤ Decidir: rejeitar ou não rejeitar H_0 .

t-test

- Usado para comparar médias.
- Assumem-se dados normalmente distribuídos.
- Tipos:
 - t-test para uma amostra (comparar média com valor fixo).
 - t-test para duas amostras independentes.
 - t-test pareado (mesmos indivíduos antes/depois).
- Estatística:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Teste de Wilcoxon

- Alternativa não-paramétrica ao t-test.
- Não requer normalidade dos dados.
- Baseado nas posições (ranks) das diferenças.
- Tipos:
 - Wilcoxon para amostras pareadas.
 - MannWhitney U (equivalente para amostras independentes).
- Útil quando há outliers ou dados assimétricos.

Código Python - t-test

```
1 from scipy import stats
2
3 # Dados fictícios
4 grupo1 = [5.1, 5.3, 5.5, 5.0, 5.2]
5 grupo2 = [5.7, 5.8, 5.6, 5.9, 6.0]
6
7 # t-test para amostras independentes
8 t_stat, p_val = stats.ttest_ind(grupo1, grupo2)
9
10 print(f"t = {t_stat:.3f}, p-valor = {p_val:.3f}")
```

Código Python - Teste de Wilcoxon

```
1 from scipy import stats
2
3 # Dados pareados fictícios
4 antes = [10, 12, 9, 14, 11]
5 depois = [12, 14, 11, 15, 13]
6
7 # Teste de Wilcoxon
8 w_stat, p_val = stats.wilcoxon(antes, depois)
9
10 print(f"estatística = {w_stat}, p-valor = {p_val:.3f}")
```

Comparativo entre t-test vs test wilcoxon

- Testes de hipótese ajudam a tomar decisões baseadas em dados.
- t-test: bom para dados normais.
- Wilcoxon: robusto para distribuições não-normais.
- Sempre verifique pressupostos antes de escolher o teste.

Regressão

Predição do valor da ação



Regressão

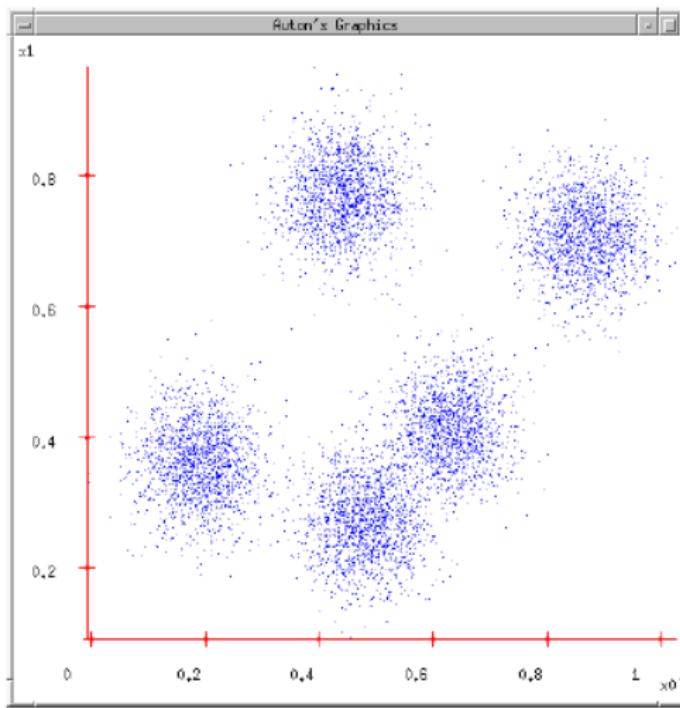
Predição do tempo



Temperature
72° F

Agrupamento - Clusterização

Agrupar coisas similares



Conceitos

Capacidade de generalização

Definida como a capacidade do classificador de prever corretamente a classe de novos dados

Sobre-ajuste - Overfitting

No caso em que o modelo se especializa nos dados utilizados em seu treinamento, apresentando uma baixa taxa de acerto quando confrontado com novos dados, tem-se a ocorrência de um superajuste (overfitting).

Sub-ajuste - Underfitting

E também possível induzir hipóteses que apresentem uma baixa taxa de acerto mesmo no subconjunto de treinamento, configurando uma condição de subajuste (underfitting).

Essas situações podem ocorrer, por exemplo, quando os exemplos de treinamento disponíveis são pouco representativos ou quando o modelo obtido é muito simples

Crescimento de Aprendizado de Máquina

Aprendizado de Maquina é uma abordagem muito utilizada para:

- Reconhecimento de fala, processamento de linguagem natural
- Visão computacional
- Análises médica
- Biologia computacional
- Redes de sensores

Esta tendência é acelerada por:

- Big Data
- Melhoria de Algoritmos de aprendizado de máquina
- Computadores mais rápido

Aprendizado Supervisionado

Definição

- Dado um conjunto de treinamento $f(x_i; y_i) \ i = 1, \dots, N$
- Encontrar uma boa aproximação para $f : XY$

Exemplos: O que representam X e Y

- Detecção Spam
 - Mapear texto para (Spam, Não-Spam)
- Reconhecimento de Dígito
 - Mapear pixels para 0,1,2,3,4,5,6,7,8,9
- Predição de Ações
 - Mapear preços históricos para < (número real)

Problema de Aprendizado de Máquina

Conjunto de Dados

Exemplo	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Nosso objetivo é encontrar uma função

$$f : XY$$

$$X = \{0, 1\}^4$$

$$Y = \{0, 1\}$$

Questão 1

Como definir o espaço de hipótese, o conjunto possíveis de f

Questão 2

Como encontrar melhor f no espaço de hipótese

Espaço de Hipóteses mais gerais

Considere todas as possíveis funções booleanas sobre 4 características como entrada:

2^{16} hipóteses possíveis
2⁹ são consistente com
nossa conjunto de dados.
Como escolher a melhor
hipótese?

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Conjunto de Dados					
Exemplo	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Espaço de Hipóteses mais gerais

Considere todas as funções booleanas conjuntivas

16 hipóteses possíveis
nenhuma é consistente
com nosso conjunto de
dados

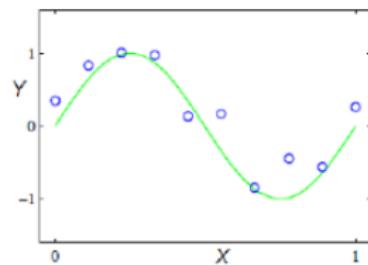
Como escolher a melhor
hipótese?

Rule	Counterexample
$\Rightarrow y$	1
$x_1 \Rightarrow y$	3
$x_2 \Rightarrow y$	2
$x_3 \Rightarrow y$	1
$x_4 \Rightarrow y$	7
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \Rightarrow y$	3
$x_2 \wedge x_4 \Rightarrow y$	3
$x_3 \wedge x_4 \Rightarrow y$	4
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3

Conjunto de Dados					
Exemplo	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Regressão

Conjunto de dados: 10 pontos (X, Y) gerados a partir da função seno com ruído



Regressão

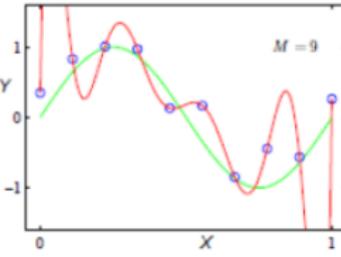
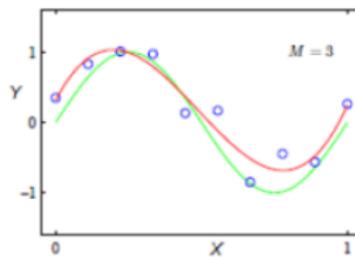
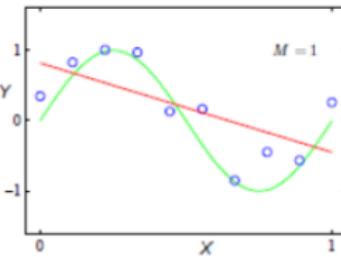
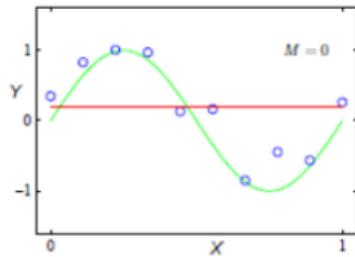
$$f : X \rightarrow Y$$

$$X = \mathcal{R}$$

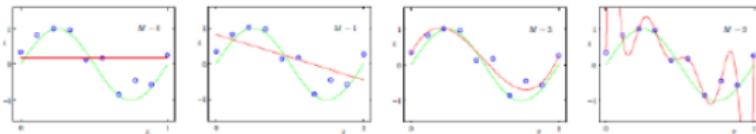
$$Y = \mathcal{R}$$

Regressão

Qual é o melhor grau do polinômio M ?



Regressão



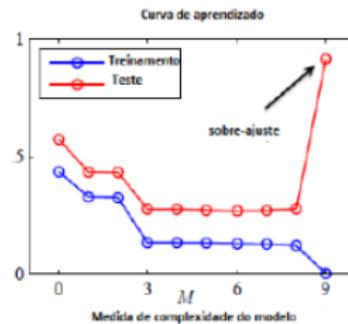
Nós medimos o erro usando uma função perda $L(y, \hat{y})$

Para regressão, uma escolha comum é perda quadrada:

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2$$

A perda empírica da função f aplicada para dados de treinamento é então

$$\frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$



Princípio de Occam's Razor

William of Occam: Monge viveu no século 14

Princípio da parcimônia:

"One should not increase, beyond what is necessary, the number entities required to explain anything"

- Quando várias soluções estão disponíveis para um dado problema, nós devemos selecionar aquela mais simples
- Mas o que nós queremos dizer por simples?
- Nós usaremos o conhecimento a priori do problema para solucionar e definir o que é uma solução simples
- Exemplo de um conhecimento a priori: suavidade

Questões chave em Aprendizado de Máquina

- Como vamos escolher um espaço de hipótese?
 - Frequentemente nós usamos conhecimento a priori para guiar esta escolha
- Como nós podemos avaliar a precisão de uma hipótese sobre dados não vistos?
 - Occam's razor: usa a hipótese mais simples consistente com dados. Isto ajudaria a evitar o sobre-ajuste
 - Teoria de Aprendizado vai nos ajudar a quantificar a capacidade de generalização como uma função da quantidade de dados de treinamento e o espaço de hipótese.
- Como iremos encontrar a melhor hipótese?

Esta é uma questão algorítmica, o tópico principal da ciência da computação
- Como modelar aplicações como problema de aprendizado de máquina?
(Desafio de engenharia)

Conceitos

Hipótese Suficiente

Uma hipótese é chamada de suficiente se e somente se ela tem valores 1 para todos os exemplos de treinamento rotulados por um 1

Hipótese Necessária

Uma hipótese é chamada de necessária se e somente se ela tem valores 0 para todos os exemplos de treinamento rotulados por um 0

Hipótese consistente

Uma hipótese é consistente com o conjunto de treinamento se ambas suficiente e necessária

Erro verdadeiro de uma hipótese

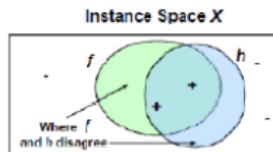
Duas noções de erro

- Erro de treinamento de uma hipótese h com respeito para um conceito f
 - Como frequentemente $h(x) \neq f(x)$ sobre as instâncias de treinamento
- Erro verdadeiro de uma hipótese h com relação á f
 - Como frequentemente $h(x) \neq f(x)$ sobre instâncias randômicas obtidas da distribuição D

Definição

O erro verdadeiro (denotado por $\text{erro}_D(h)$) de hipótese h com respeito para um conceito alvo f e distribuição D é a probabilidade que h irá classificar incorretamente uma instância obtida randomicamente de acordo com uma distribuição D .

$$\text{erro}_D(h) \equiv \text{Prob}_{x \in D}[f(x) \neq h(x)]$$



Limite PAC e Dilema Bias-Variância

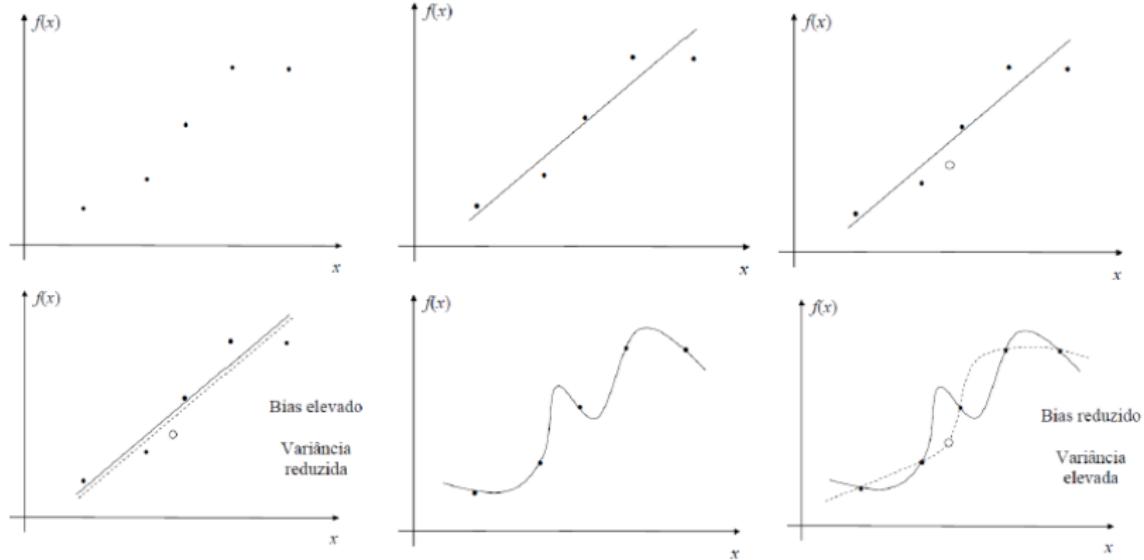
Para todo h , com probabilidade de pelo menos $1 - \delta$:

$$erro_{true}(h) \leq \underbrace{erro_D(h)}_{bias} + \sqrt{\underbrace{\frac{\ln |\mathcal{H}| + \ln(1/\delta)}{2m}}_{variance}}$$

Conclusão

- Para $|\mathcal{H}|$ grande
 - Baixo bias (assumindo que encontramos um bom h)
 - Alta variância (por que é mais flexível)
- Para $|\mathcal{H}|$ pequeno
 - Alto bias (há um bom h)
 - Baixa variância (por que esta mais justo)

Dilema Bias-Variância



Risco Empírico vs Risco Esperado

- Aprender uma função de classificação binária a partir dos dados
- Considere um conjunto de dados $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ onde cada $y_i \in \{-1, 1\}$.
- Aprender uma função $y = f(x; \theta)$ que irá classificar corretamente os exemplos não observados
- Como é que vamos escolher o tipo de f e θ ?
- Otimizando alguma medida de performance do modelo aprendido.
- O que é uma medida boa de performance?
- Uma medida boa de performance é o risco esperado

$$[R_f(\theta) = E[L(y, f(x; \theta))] = \int L(y, f(x; \theta)) dP(x, y)]$$

- Valor esperado da função de perda

Dimensão VC

A dimensão Vapnik-Chervonenkis

Esta é uma medida da complexidade / capacidade de uma classe de funções \mathcal{F} . Ela mede o maior número de exemplos que podem ser explicados pela família \mathcal{F} .

Compromisso entre Alta capacidade e Boa Generalização

Maior Capacidade

Se a família \mathcal{F} tem capacidade suficiente para explicar todos os possíveis conjuntos de dados → há risco de sobre-ajuste

Menor Capacidade

Funções $f \in \mathcal{F}$ tendo pequena capacidade podem não ser capazes de explicar nosso conjunto de dados particular, entretanto, são menos propensa a sobre-ajuste.

Como a dimensão VC caracteriza este compromisso?

Dimensão Vapnik-Chervonenkis

$$R_f(\theta) = E[\frac{1}{2}|y - f(x; \theta)|], R_f^{emp}(\theta) = \frac{1}{n} \sum_{i=1}^m \frac{1}{2}|y_i - f(x_i, \theta)|$$

- Dada uma classe de funções \mathcal{F} , seja $VcDim$ ser sua dimensão VC
- $VcDim$ é uma medida da capacidade de \mathcal{F} ($VcDim$ não depende da escolha do conjunto de treinamento)
- Vapnik mostrou que com probabilidade $1 - \delta$

$$R_f(\theta) \leq R_f^{emp}(\theta) + \sqrt{\frac{VcDim(\log(\frac{2m}{VcDim}) + 1) - \log(\frac{\delta}{4})}{m}}$$

Isto nos dá uma maneira de estimar o erro sobre dados futuros com base apenas no erro de treinamento e na dimensão VC de \mathcal{F} .

Dado \mathcal{F} como nós podemos definir e calcular $VcDim$, sua dimensão VC?

Classificando

Uma função $f(x; \theta)$ pode classificar um conjunto de pontos x_1, x_2, \dots, x_m se e somente se

Para todo conjunto de treinamento possível da forma $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ há algum valor de θ tal que $f(x_i, \theta) = y_i$ para $i = 1, \dots, m$.

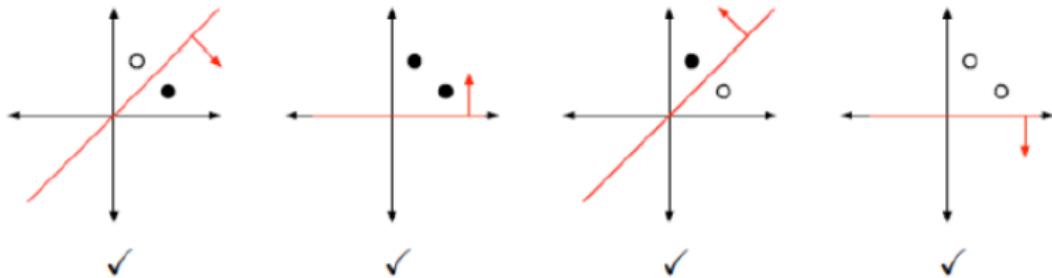
Há 2^m conjuntos de treinamento a considerar, cada um com uma combinação diferente de +1's e -1's para y 's.

Exemplo

Uma função $f(x; \theta)$ pode classificar um conjunto de pontos x_1, x_2, \dots, x_m se e somente se para todo conjunto de treinamento possível da forma $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ há algum valor de θ tal que $f(x_i, \theta) = y_i$ para $i = 1, \dots, m$.

Resposta

Nenhum problema. Há quatro conjuntos de dados a considerar.



Exemplo

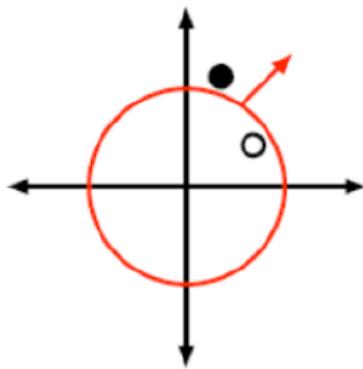
Uma função $f(x; \theta)$ pode classificar um conjunto de pontos x_1, x_2, \dots, x_m se e somente se

Para todo conjunto de treinamento possível da forma

$(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ há algum valor de θ tal que $f(x_i, \theta) = y_i$ para $i = 1, \dots, m$.

Pode a seguinte função classificar os seguintes pontos?

$$f(x; b) = \text{sign}(x^T x - b)$$



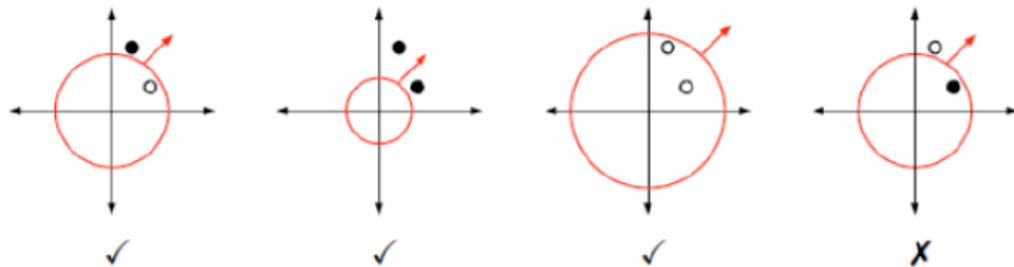
Exemplo

Uma função $f(x; \theta)$ pode classificar um conjunto de pontos x_1, x_2, \dots, x_m se e somente se

Para todo conjunto de treinamento possível da forma $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ há algum valor de θ tal que $f(x_i, \theta) = y_i$ para $i = 1, \dots, m$.

Resposta

Não é possível.



Definição da Dimensão VC

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$ (em geral, não será verdade que todo conjunto de pontos pode ser classificado).

Questão

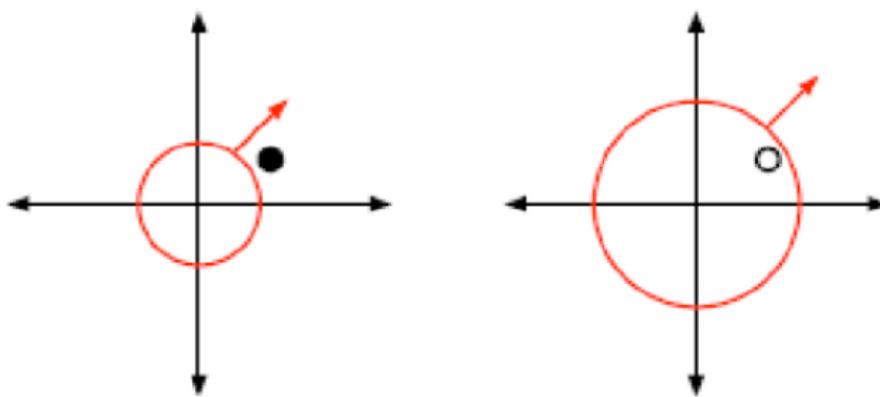
Qual é a dimensão VC de $f(x, b) = sign(x^T x - b)$?

Definição da Dimensão VC

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Nós não podemos mesmo classificar dois pontos. É claro que um ponto pode ser classificado



Definição da Dimensão VC

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Exemplo

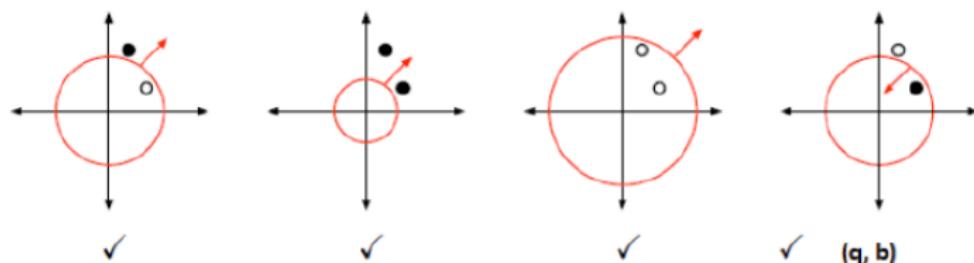
Para entradas bi-dimensional, qual é a dimensão de

$$f(x; q, b) = sign(qx^T x - b)$$

Definição da Dimensão VC

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta: 2



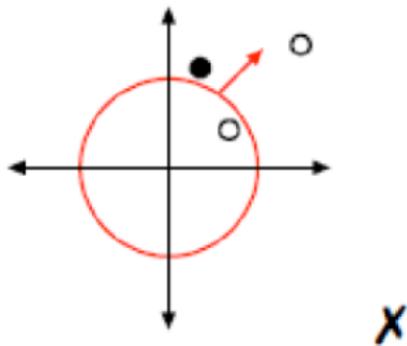
Definição da Dimensão VC

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Exemplo

Qual é a dimensão VC de $f(x; q, b) = sign(qx^T x - b)$

Resposta: 2 (claramente não pode ser 3)



Definição VC de uma reta

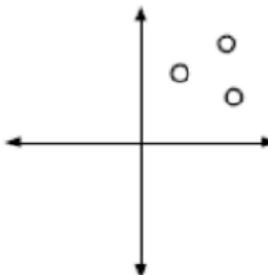
Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Para entradas bidimensionais, qual é a dimensão VC de

$$f(x; w, b) = sign(w^T x + b)$$

Pode f classificar estes 3 pontos?



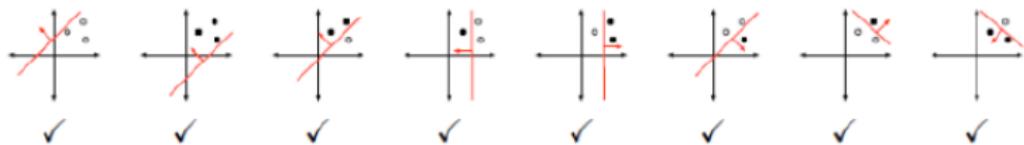
Definição VC de uma reta

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Exemplo

Qual é a dimensão VC de $f(x; w, b) = sign(w^T x + b)$

Resposta: Sim, pode classificar 3 pontos.



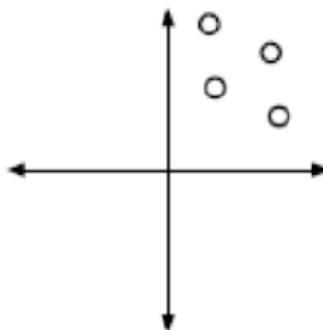
Definição VC de uma reta

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Para entradas bidimensionais, qual é a dimensão VC de $f(x; w, b) = sign(w^T x + b)$?

Pode f classificar estes 4 pontos?



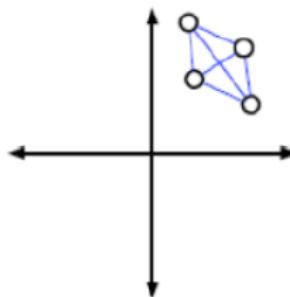
Definição VC de uma reta

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Para entradas bidimensionais, qual é a dimensão VC de $f(x; w, b) = sign(w^T x + b)$?

Podemos sempre desenhar 6 linhas entre pares de 4 pontos



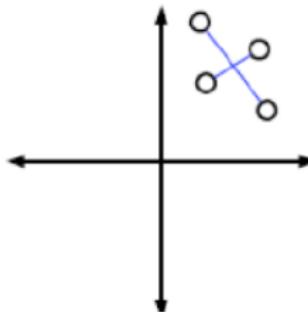
Definição VC de uma reta

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Para entradas bidimensionais, qual é a dimensão VC de $f(x; w, b) = sign(w^T x + b)$?

Podemos sempre desenhar 6 linhas entre pares de 4 pontos
Duas destas linhas se cruzam



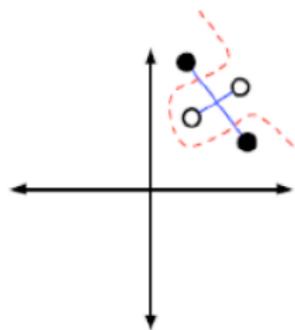
Definição VC de uma reta

Dada a classe de funções \mathcal{F} , ele tem dimensão VC $VCdim$ se há pelo menos um conjunto de $VCdim$ pontos que podem ser classificados por $f \in \mathcal{F}$

Resposta

Qual é a dimensão VC de $f(x; w, b) = sign(w^T x + b)$?

Podemos sempre desenhar 6 linhas entre pares de 4 pontos
Duas destas linhas se cruzam
Se analisarmos os pontos ligados pela reta que se cruzam, veremos que eles não poderão ser separados linearmente.



Uma linha pode classificar 3 pontos mas não 4 \rightarrow a dimensão VC de uma linha de separação é 3.

O que mede a dimensão VC

É o número de parâmetros?

Relacionado, mas não é a mesma coisa

Pode-se esperar que intuitivamente modelos com um número grande de parâmetros livres teriam maior dimensão VC, enquanto os modelos com poucos parâmetros teria dimensões VC baixa

Minimização do Risco Estrutural

Outro termo formal para um conceito intuitivo: o modelo ótimo é encontrado por estabelecendo um equilíbrio entre risco empírico e a dimensão VC.

Relembre

$$R(\theta) \leq R^{\text{emp}}(\theta) + \sqrt{\frac{VcDim(\log(\frac{2m}{VcDim}) + 1) - \log(\delta/4)}{m}}$$

O princípio SRM procede como a seguir

- Construa uma estrutura aninhada para a família de classes de funções $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \dots \subseteq \mathcal{F}_k$ com dimensão VC não decrescente ($VcDim_1 \leq VcDim_2 \leq \dots \leq VcDim_k$)
- Para cada classe de \mathcal{F}_j , compute a solução f_i que minimiza o risco empírico
- Escolha a classe de função \mathcal{F}_j , e correspondente solução f_i , que minimiza o limitante do risco.

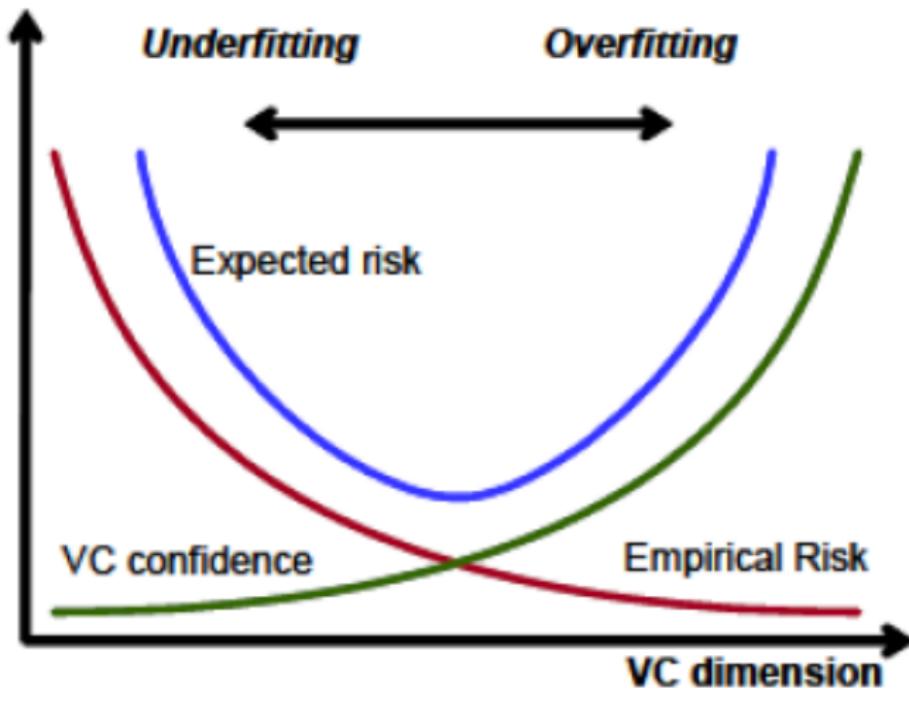
Em outras palavras

- Treine um conjunto de máquinas, um para cada subconjunto
- Para um dado subconjunto, minimize o risco empírico
- Escolha a máquina cuja soma do risco empírico e a confiança VC é mínimo

i	\mathcal{F}_i	$R^{\text{emp}}(\theta)$	VC Confidence	Probable Upper bound	Choice
1	\mathcal{F}_1				
2	\mathcal{F}_2				
3	\mathcal{F}_3				
4	\mathcal{F}_4				
5	\mathcal{F}_5				
6	\mathcal{F}_6				

Observe que o termo da dimensão VC é usualmente muito, muito conservador (pelo menos centena de vezes maior que o efeito de sobre-ajuste)

Minimização do Risco Estrutural



Usando dimensão VC

Vários pesquisadores tem trabalhado arduamente para encontrar a dimensão VC para

- Perceptrons
- Redes Neurais
- Máquinas de Vetores Suporte
- e Muito mais

Tudo com o objetivo de

Entender quais máquinas de aprendizado são mais ou menos poderosa sobre algumas circunstâncias

Usar a Minimização do Risco Estrutural para escolher a melhor máquina de aprendizado

Alternativas para Dimensão VC baseado seleção de modelo

Poderíamos potencialmente usar k-fold validação cruzada:

i	\mathcal{F}_i	$R^{\text{emp}}(\theta)$	VC Confidence	Probable Upper bound	Choice
1	\mathcal{F}_1				
2	\mathcal{F}_2				
3	\mathcal{F}_3				
4	\mathcal{F}_4				
5	\mathcal{F}_5				
6	\mathcal{F}_6				

Note erro de CV pode ter mais variância

Dimensão VC na prática

- Infelizmente, calcular o limite superior sobre o risco estrutural não é prático em várias situações
- A dimensão VC não pode ser precisamente estimada para modelos não lineares como rede neurais
- Implementação da Minimização do Risco Estrutural pode conduzir a problema de otimização não linear
- A dimensão VC pode ser infinita (por exemplo, Vizinho Mais Próximo com $k=1$), requerendo uma quantidade infinita de dados
- O limite superior pode algumas vezes ser trivial (maior que 1)
- Felizmente, Teoria de Aprendizado Estatística pode ser rigorosamente aplicada a modelos lineares.

Modelos Paramétricos vc Modelos não Paramétricos

- análise multivariável clássica: fornece ferramentas poderosas na obtenção de associações lineares entre as variáveis.
- se todas as informações relevantes puderem ser extraídas com base nestas ferramentas clássicas, nenhum passo adicional se faz necessário.
- nos casos em que associações não lineares arbitrárias estão presentes, a determinação do tipo de não linearidade é fundamental para a obtenção do melhor modelo de aproximação a partir dos dados de entrada-saída
- quando a forma da não linearidade é conhecida previamente e passível de descrição matemática, modelos paramétricos são normalmente empregados, simplificando o problema de aproximação, já que os parâmetros podem ser determinados com base em técnicas de regressão não linear.

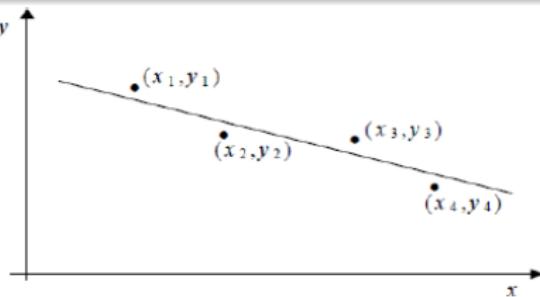
Modelos Paramétricos vc Modelos não Paramétricos

- regressão paramétrica: a forma do relacionamento funcional entre as variáveis dependentes e independentes é conhecida, mas podem existir parâmetros cujos valores são desconhecidos, embora passíveis de serem estimados a partir do conjunto de treinamento.
- em problemas paramétricos, os parâmetros livres, bem como as variáveis dependentes e independentes, geralmente têm uma interpretação física.
- Exemplo: ajuste de uma reta a uma distribuição de pontos

$$f(x) = y = ax + b$$

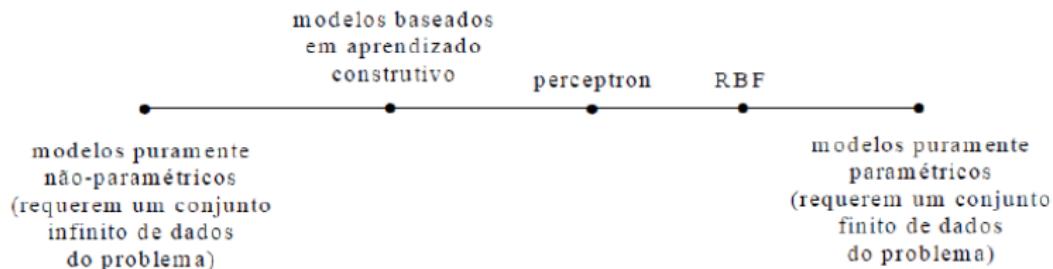
a, b desconhecidos

y : sujeito a ruído



Modelos Paramétricos vc Modelos não Paramétricos

- regressão não paramétrica: sua característica distintiva é a ausência (completa ou quase completa) de conhecimento a priori a respeito da forma da função que está sendo estimada. Sendo assim, mesmo que a função continue a ser estimada a partir do ajuste de parâmetros livres, o conjunto de "formas" que a função pode assumir (classe de funções que o modelo do estimador pode prever) é muito amplo.
- como consequência, vai existir um número elevado de parâmetros (por exemplo, quando comparado ao número de dados de entrada-saída para treinamento), os quais não mais admitem uma interpretação física isolada.



Modelos Paramétricos vc Modelos não Paramétricos

- todos os modelos de regressão que não são puramente paramétricos são denominados não paramétricos ou semi-paramétricos. Esta denominação não deve causar confusão, principalmente levando-se em conta que modelos de regressão puramente não paramétricos são intratáveis.
- com base no exposto acima, fica evidente que redes neurais artificiais para treinamento supervisionado pertencem à classe de modelos de regressão não paramétricos. Sendo assim, os pesos não apresentam um significado físico particular em relação ao problema de aplicação.
- além disso, estimar os parâmetros de um modelo não paramétrico (por exemplo, pesos de uma rede neural artificial) não é o objetivo primário do aprendizado supervisionado. O objetivo primário é estimar a “forma” da função em uma região compacta do espaço de aproximação (ou ao menos a saída para certos valores desejados de entrada).

Modelos Paramétricos vc Modelos não Paramétricos

- quando a forma da não linearidade é desconhecida, a utilização de modelos paramétricos pode representar uma perda acentuada de flexibilidade de representação, principalmente nos casos em que as restrições paramétricas impostas ao modelo de aproximação não correspondem à forma da não linearidade que deve ser aproximada.
- mesmo no caso de problemas de aproximação passíveis de tratamento paramétrico, é recomendada uma abordagem inicial utilizando modelos não paramétricos para auxiliar na determinação do tipo de parametrização que pode ser utilizada.
- STONE (1977) faz um estudo mais aprofundado desta e outras motivações para o emprego de modelos não paramétricos.
- dificuldade: o emprego de modelos não paramétricos provoca uma significativa acentuação de uma característica já presente em alguns problemas de aproximação que utilizam abordagens paramétricas: a maldição da dimensionalidade.

Modelos Paramétricos vc Modelos não Paramétricos

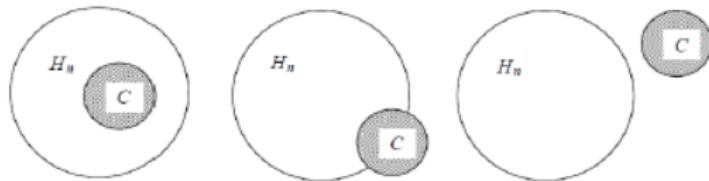
Maldição da dimensionalidade

- esta expressão foi empregada originalmente por BELLMAN (1961) e se refere basicamente à existência de uma relação direta entre a dimensionalidade dos dados e a quantidade de dados necessária para possibilitar o sucesso da tarefa de aproximação.
- a consequência prática da maldição da dimensionalidade é a necessidade de um aumento exponencial no número de dados para a manutenção do poder de aproximação com um aumento da dimensão do espaço de aproximação.

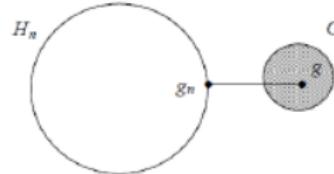
- apesar de estar invariavelmente associada a uma redução da capacidade de aproximação, a imposição de um conjunto de restrições (inclusive, restrições paramétricas) aos modelos não paramétricos pode reduzir significativamente o efeito da maldição da dimensionalidade.
- conclusão: os melhores modelos de aproximação são aqueles capazes de conciliar o nível de dependência da dimensionalidade com a flexibilidade do modelo de aproximação.

Representação

Modelagem paramétrica x modelagem não paramétrica



$$g_n = \arg \min_{\hat{g}_n \in H_n} \|g - \hat{g}_n\|, \text{ com } g \in C$$

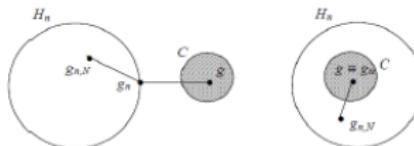


sobre que condições e que taxa $g_n \rightarrow g$ quando $n \rightarrow \infty$

Representação

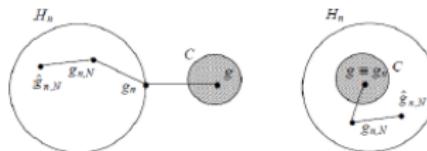
Estimação

$$g_{n,N} = \arg \min_{\hat{g}_n \in H_n, T_N} \|s_l - \hat{g}_{n,N}(x_l)\|$$



sob que condições e a que taxa $g_{n,N} \rightarrow g_n$ quando $N \rightarrow \infty$

Computação



Composição do erro de estimação

$$\hat{e}_{n,N} = g - \hat{g}_{n,N}$$

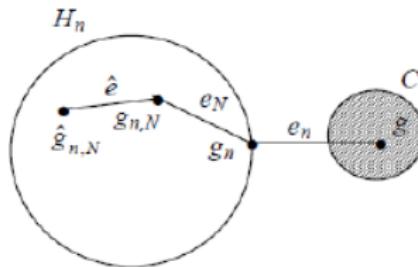
$$\hat{e}_{n,N} = \underbrace{g - g_n}_{e_n} + \underbrace{g_n - g_{n,N}}_{e_N} + \underbrace{g_{n,N} - \hat{g}_{n,N}}_{\hat{e}}$$

e_n : erro de aproximação

e_N : erro de estimação

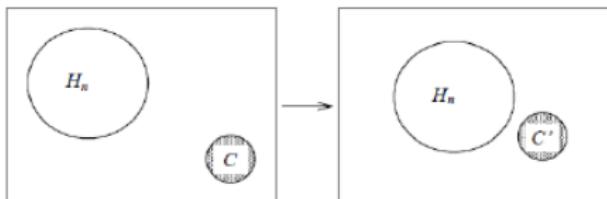
\hat{e} : erro de computação

Usando a desigualdade triangular: $\|\hat{e}_{n,N}\| \leq \|e_n\| + \|e_N\| + \|\hat{e}\|$



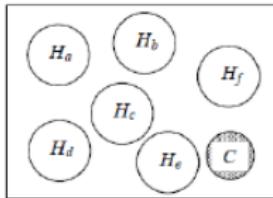
Técnicas para redução do erro de aproximação

- 1) Simplificar o problema de aproximação (modelo e/ou função)



- 2) Escolher uma melhor classe de modelos de aproximação

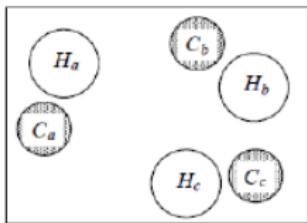
- não existe uma classe de modelos que seja a mais adequada para todos os problemas



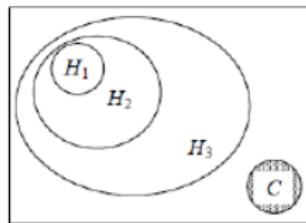
Técnicas para redução do erro de aproximação

3) Quebrar o problema em problemas menores (mais simples)

3(a) Mixture of experts



3(b) Métodos construtivos



4) Redução de dimensionalidade

