

Introdução ao Aprendizado de Máquina

Prof. Danilo Silva

EEL7514/EEL7513 - Tópico Avançado em Processamento de Sinais:
Introdução ao Aprendizado de Máquina

EEL / CTC / UFSC

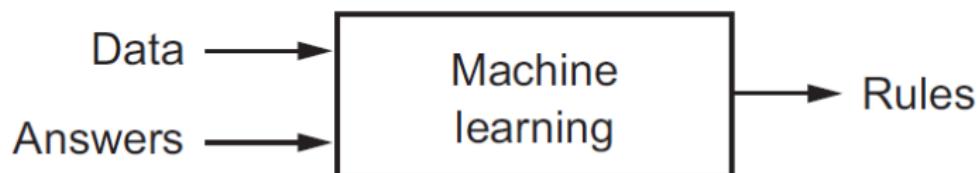
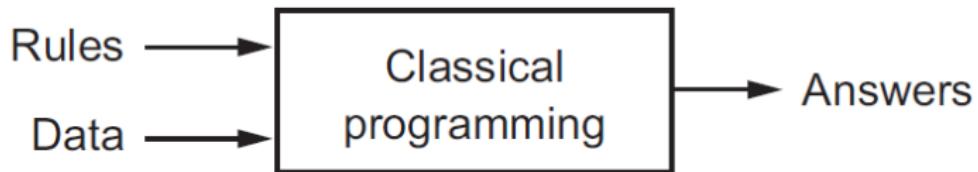
Introdução

- ▶ Aprendizado de máquina (*machine learning*) é um campo de estudo voltado ao projeto e análise de métodos computacionais para realizar tarefas sem necessitar de instruções explícitas
- ▶ Aprendizado refere-se à capacidade de um programa de computador de melhorar seu desempenho em uma dada tarefa a partir da experiência
- ▶ Experiência refere-se à observação de **exemplos** (conjunto de variáveis observadas) e/ou de **feedback** (recompensa/punição) sobre seu desempenho na tarefa
- ▶ **Exemplo:** reconhecimento de faces
 - ▶ Difícil descrever ou programar
 - ▶ Fácil a partir de exemplos

Introdução

- ▶ Abordagem tradicional de engenharia:
 - ▶ Análise do problema
 - ▶ Definição de um modelo matemático
 - ▶ Soluções são criadas a partir do modelo
- ▶ Abordagem do aprendizado de máquina:
 - ▶ Coletar dados (exemplos) que relacionam entrada e saída desejada
 - ▶ Definir um métrica de desempenho
 - ▶ Treinar um algoritmo de aprendizado genérico para executar a tarefa
- ▶ **Motivação:** Em muitas situações, dados + computação podem produzir uma solução em menos tempo e com menor custo do que contratar especialistas para resolver o problema

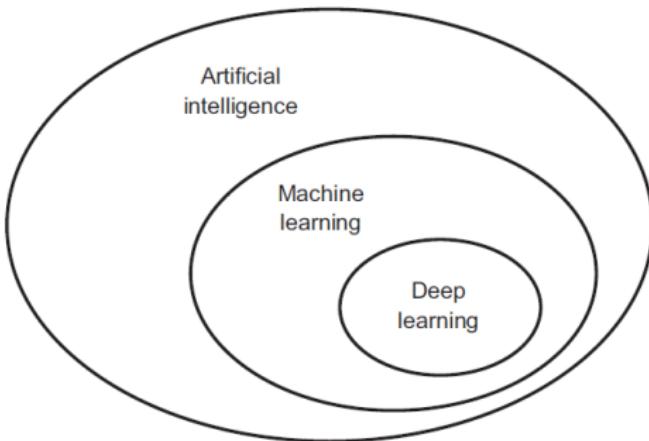
Introdução



Tarefas adequadas para aprendizado de máquina

- ▶ Critérios sugeridos por Brynjolfsson & Mitchell (2017):
 - ▶ *The task involves a function that maps well-defined inputs to well-defined outputs;*
 - ▶ *large data sets exist or can be created containing input-output pairs;*
 - ▶ *the task provides clear feedback with clearly definable goals and metrics;*
 - ▶ *the task does not involve long chains of logic or reasoning that depend on diverse background knowledge or common sense;*
 - ▶ *the task does not require detailed explanations for how the decision was made;*
 - ▶ *the task has a tolerance for error and no need for provably correct or optimal solutions;*
 - ▶ *the phenomenon or function being learned should not change rapidly over time; and*
 - ▶ *no specialized dexterity, physical skills, or mobility is required.*

Artificial Intelligence / Machine Learning / Deep Learning



- ▶ Aprendizado de máquina:
 - ▶ difere da inteligência artificial simbólica clássica (baseada em regras lógicas e buscas estruturadas), por permitir o aprendizado a partir de dados
 - ▶ difere da estatística convencional apenas pelo enfoque computacional e em modelos que fazem uso de um grande volume de dados
- ▶ Aprendizado profundo (*deep learning*):
 - ▶ refere-se a **redes neurais com múltiplas camadas**
 - ▶ responsável pelo *boom* da inteligência artificial a partir de 2012

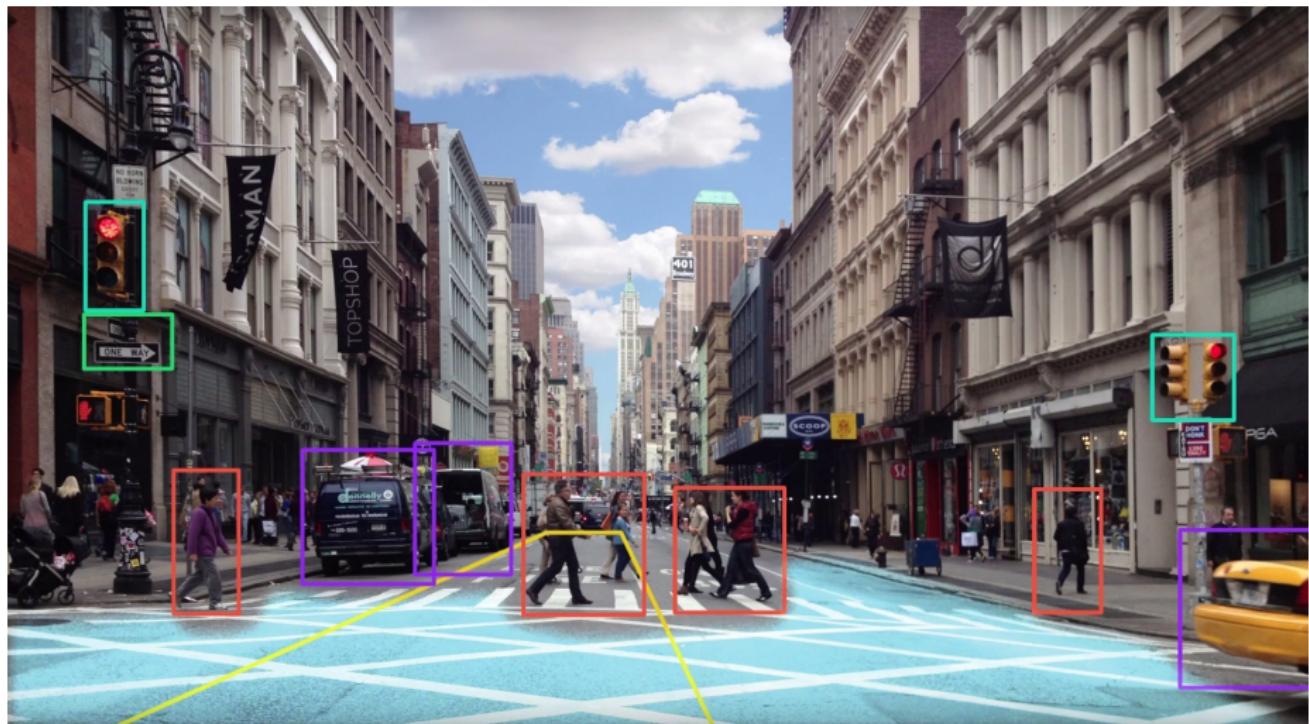
Exemplos de aplicações

- Reconhecimento de caracteres

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

Exemplos de aplicações

- Detecção e localização de objetos em imagens:



Exemplos de aplicações

- ▶ Navegação autônoma
- ▶ Reconhecimento de faces
- ▶ Reconhecimento de fala
- ▶ Tradução automática
- ▶ Detecção de spam
- ▶ Detecção de transações fraudulentas
- ▶ Diagnóstico médico
- ▶ Predição de preços de ações
- ▶ Avaliação de risco
- ▶ Recomendação de produtos
- ▶ Segmentação de clientes
- ▶ Jogos eletrônicos
- ▶ ...

Exemplos de aplicações

- ▶ Alguns trabalhos recentes na UFSC:
 - ▶ Detecção de pessoas em ambiente industrial utilizando imagens de profundidade (2016)
 - ▶ Classificação automática de modulações (2017)
 - ▶ Reconhecimento de gestos em vídeos (2018)
 - ▶ Detecção de veículos em imagens (2018)
 - ▶ Estimação do fluxo de calor em uma superfície através de imagens do escoamento (2019)
 - ▶ Detecção de palavras-chave em sinais de fala (2019)
 - ▶ Levantamento de defeitos em pavimento rodoviário (2019)
 - ▶ Detecção de massas em mamografias (2019)
 - ▶ Classificação de imagens de frutas (2019)
 - ▶ Detecção e reconhecimento de placas veiculares (em andamento)
 - ▶ Decodificação de códigos de canal usando *deep learning* (em andamento)

Tipos de Aprendizado

- ▶ Aprendizado supervisionado
- ▶ Aprendizado não-supervisionado
- ▶ Aprendizado por reforço

Aprendizado Supervisionado

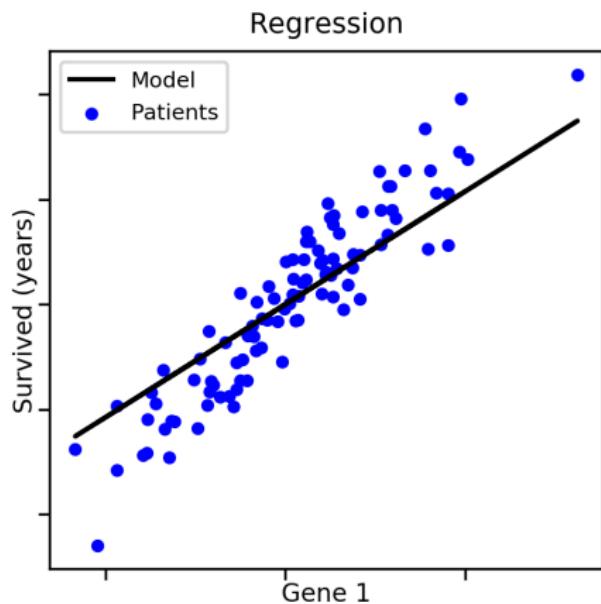
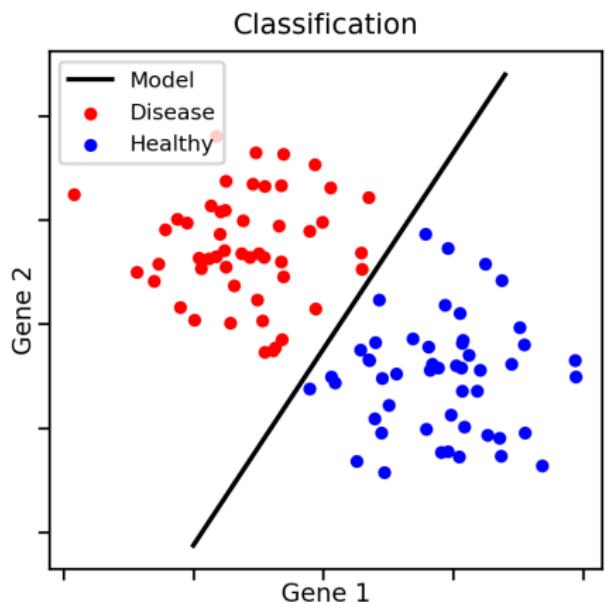
- Dispõe-se de um conjunto de dados rotulados (entrada \mathbf{x} , saída y)

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)})\}$$

provenientes de uma distribuição $p(\mathbf{x}, y)$ desconhecida

- O objetivo é construir uma função $y = f(\mathbf{x})$ (modelo) para prever o rótulo y de uma nova amostra \mathbf{x} (não-previamente observada) da mesma distribuição
- Tarefas:
 - **Classificação:** a variável de saída é discreta: $y \in \{1, \dots, K\}$
 - Exemplos: classificação de objetos em imagens, detecção de patologias, reconhecimento de fala, detecção de spam
 - **Regressão:** a variável de saída é contínua: $y \in \mathbb{R}$
 - Exemplos: predição do preço de um imóvel, predição de demanda por um serviço, avaliação de risco de um empréstimo

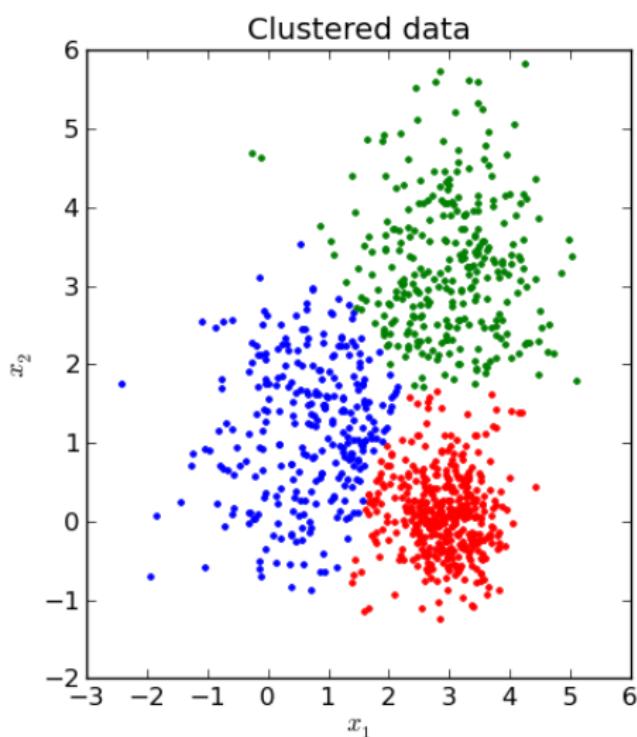
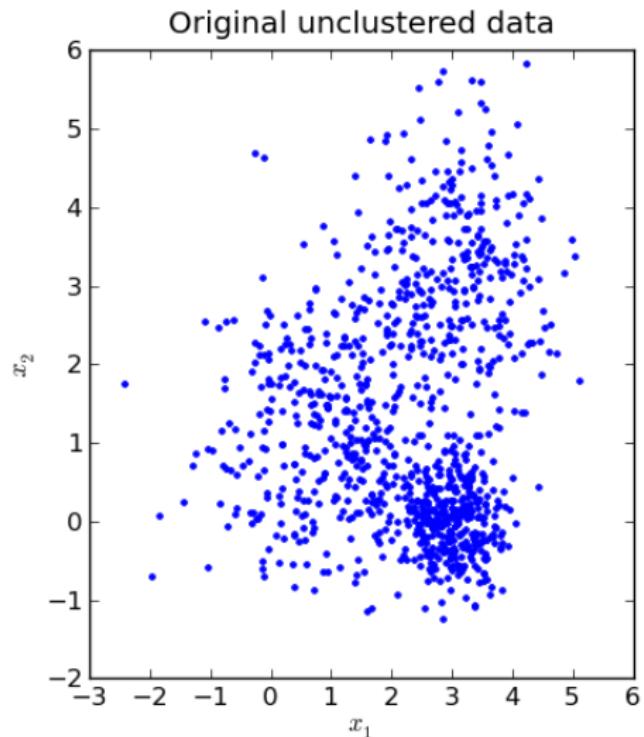
Exemplos: Classificação e Regressão



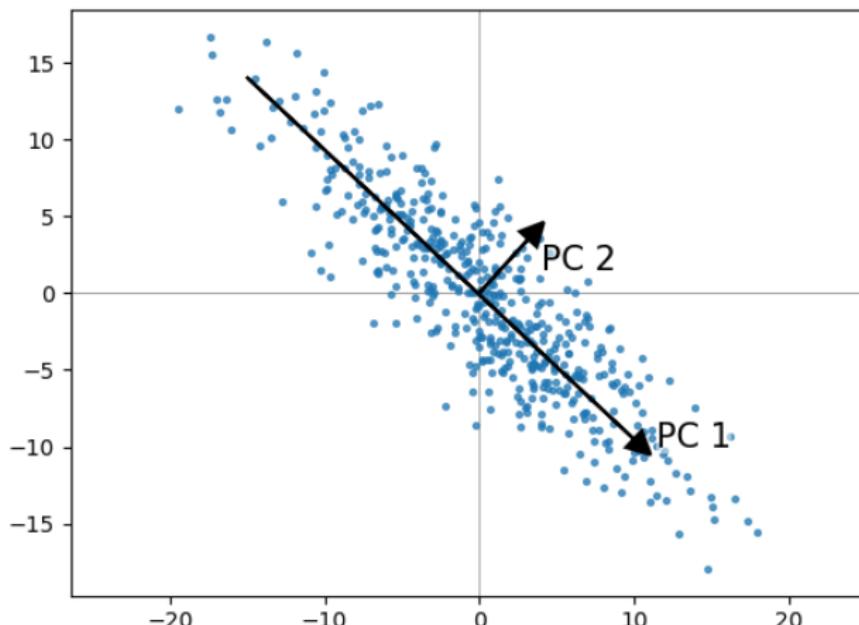
Aprendizado Não-Supervisionado

- ▶ Conjunto de dados **não-rotulados**: $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$
- ▶ O objetivo é descobrir **propriedades** da estrutura do conjunto de dados (caracterizada pela densidade $p(\mathbf{x}) = p(x_1, \dots, x_n)$)
- ▶ Tarefas:
 - ▶ **Clustering**: descobrir grupos de exemplos (dados) similares
 - ▶ Exemplos: segmentação de mercado, agrupamento de resultados de busca, identificação de famílias de genes, segmentação de imagens
 - ▶ **Redução de dimensionalidade**: encontrar uma representação mais simples dos dados para agilizar algoritmos ou permitir visualização
 - ▶ **Detecção de anomalias**: identificar casos que fogem ao padrão esperado
 - ▶ Exemplos: detecção de fraudes, detecção de falhas em sistemas, monitoramento de saúde
 - ▶ **Modelos generativos**: gerar novos exemplos (imagens, vídeos, etc), tipicamente com características específicas

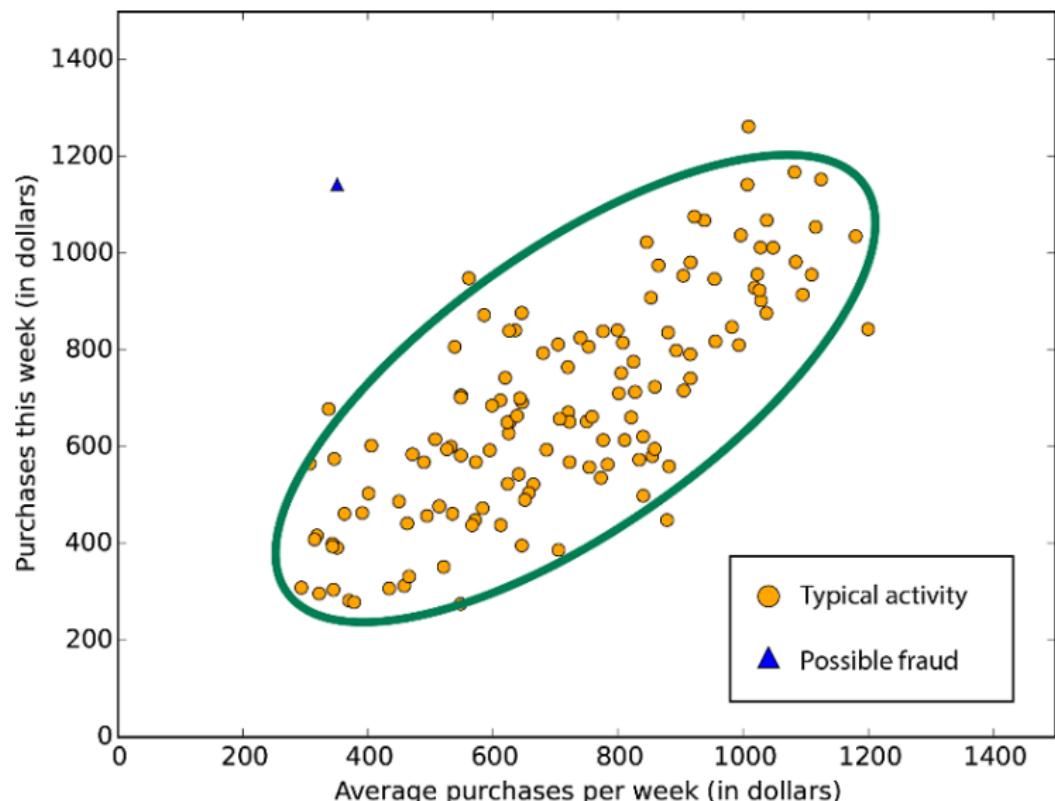
Exemplo: Clustering



Exemplo: Redução de Dimensionalidade



Exemplo: Detecção de Anomalias



Exemplo: Modelos Generativos



Exemplo: Modelos Generativos



Exemplo: Modelos Generativos

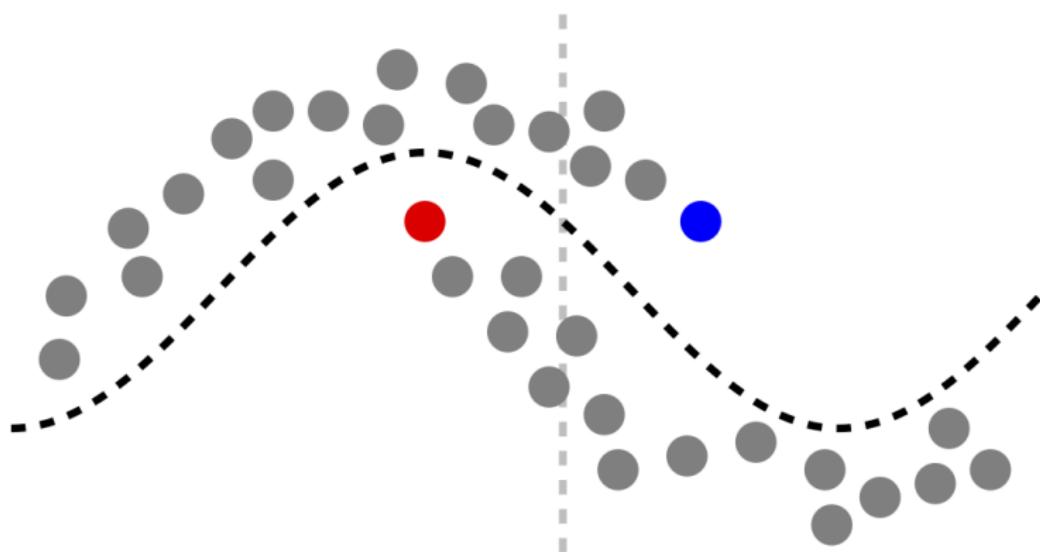
<https://www.youtube.com/watch?v=cQ54GDm1eL0>

<https://www.youtube.com/watch?v=p1b5aiTrGzY>

<https://www.youtube.com/watch?v=0ybLCfVeFL4>

Aprendizado Semi-Supervisionado

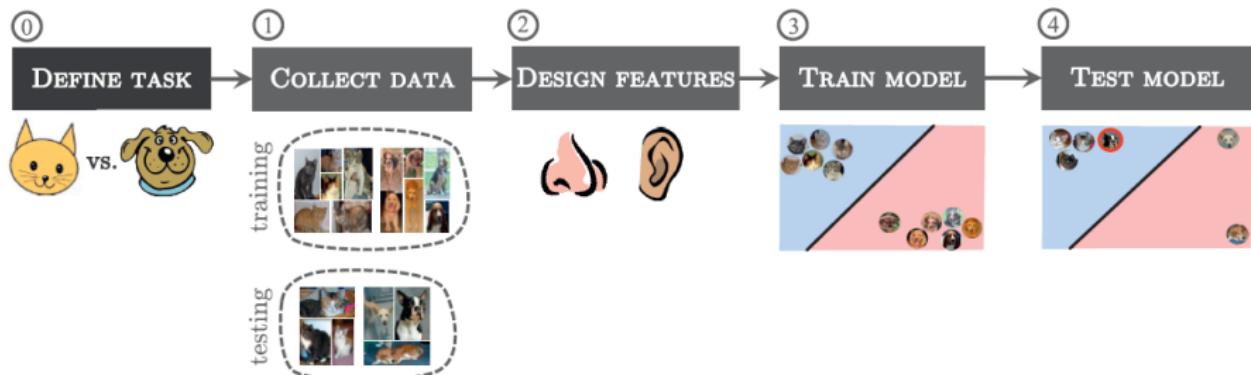
- ▶ Semelhante ao aprendizado supervisionado, porém dispõe-se também de dados não-rotulados



Aprendizado por Reforço

- ▶ O algoritmo interage com o ambiente, recebendo um sinal de feedback (recompensa/punição) a cada ação tomada
 - ▶ Formulação envolve um conjunto de estados \mathcal{S} , um conjunto de ações \mathcal{A} , uma probabilidade de transição de estados $p(s'|s, a)$ e uma recompensa associada $R_a(s, s')$
- ▶ O objetivo é descobrir e executar as melhores ações em cada situação de forma a maximizar a recompensa obtida
- ▶ O aprendizado é feito por tentativa e erro e deve balancear **descoberta** (*exploration*) e **aproveitamento** (*exploitation*)
- ▶ Exemplos: movimentação de robôs, jogos eletrônicos, otimização de redes de comunicação, aplicações financeiras, publicidade
- ▶ Frequentemente combinado com técnicas de aprendizado supervisionado para aprender uma função utilidade $Q(s, a)$

Pipeline do Aprendizado Supervisionado



1. Definição da tarefa
2. Coleta de dados
3. Desenvolvimento de atributos
4. Treinamento do modelo
5. Teste do modelo

Definição da tarefa

- ▶ Especificação do espaço de possibilidades \mathcal{Y} da variável de saída y
 - ▶ Classificação: $\mathcal{Y} = \{1, \dots, K\}$
 - ▶ Regressão: $\mathcal{Y} = \mathbb{R}$
- ▶ Especificação de uma métrica de avaliação
 - ▶ Classificação: ex: \uparrow **acurácia** (taxa de acerto) = $1 - \text{taxa de erro}$
 - ▶ Regressão: ex: \downarrow **erro quadrático médio**
- ▶ **Obs:** Em geral, a avaliação do desempenho de um preditor não é um problema trivial, podendo envolver múltiplos critérios

Coleta de dados

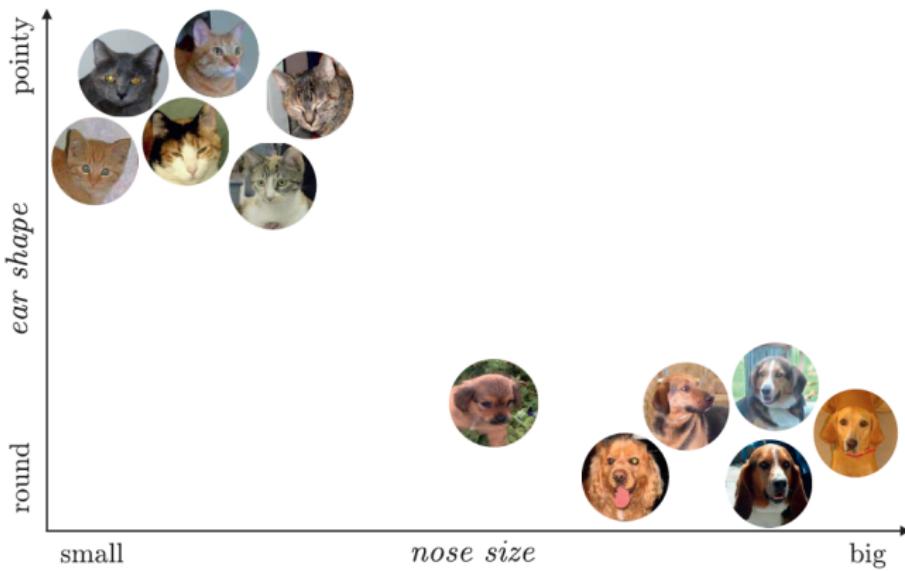
- ▶ Coletar e dividir o conjunto de dados (*dataset*) em:
 - ▶ Conjunto de treinamento (*training set*)



- ▶ Conjunto de teste (*test set*)

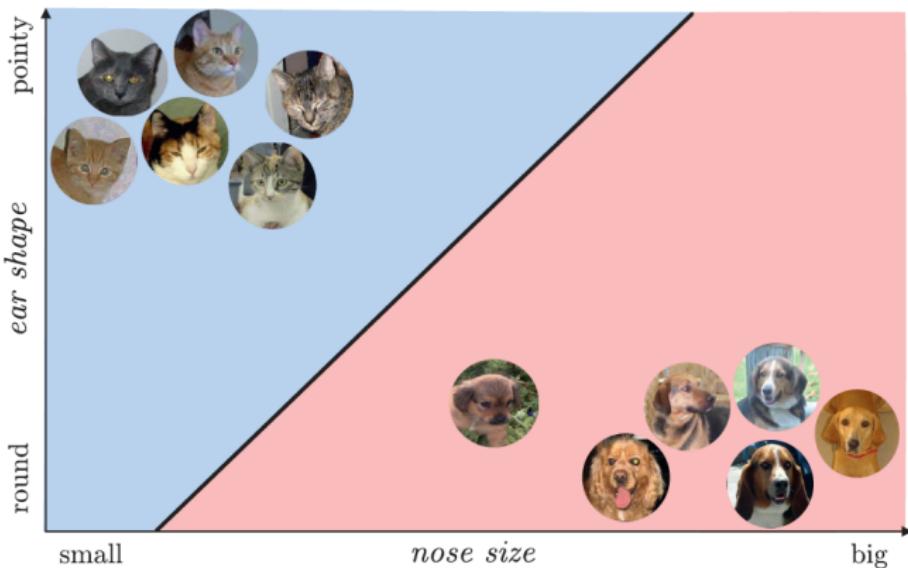


Desenvolvimento de atributos (*features*)



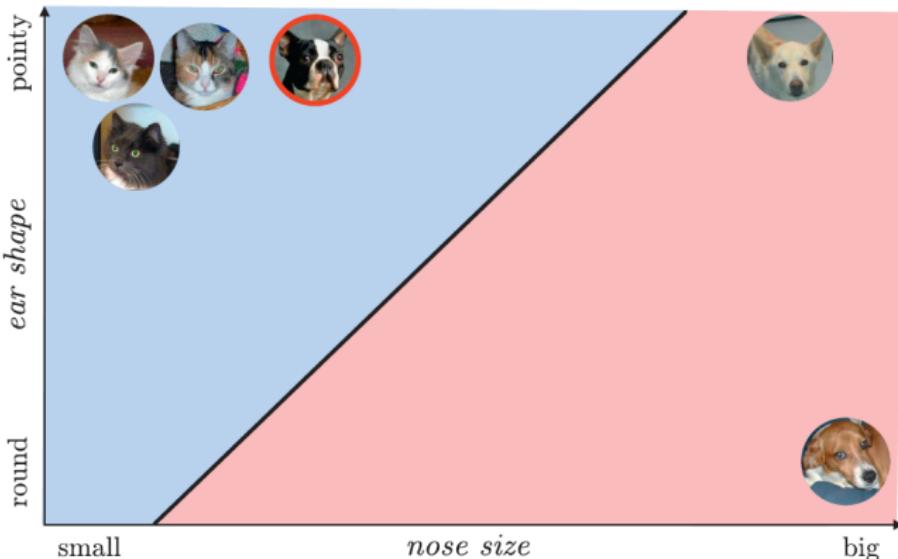
- ▶ Extrair atributos (ou características—*features*) reduz a dimensão do problema, facilitando o aprendizado
 - ▶ Vetor de atributos (*feature vector*): $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$
- ▶ Requer conhecimento específico da área de aplicação

Treinamento do modelo



- ▶ Definição de uma família de modelos: [classe/espac](#)o de hipóteses \mathcal{H}
 - ▶ Ex: modelo linear
- ▶ Treinar = escolher um modelo $f \in \mathcal{H}$ por meio de otimização numérica, de forma a minimizar o erro no conjunto de treinamento

Teste do modelo

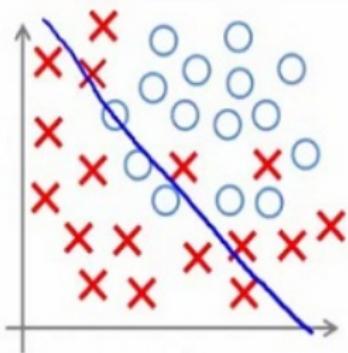


- ▶ A partir do modelo treinado, realiza-se uma predição $\hat{y} = f(\mathbf{x})$ para cada amostra \mathbf{x} do conjunto de teste
- ▶ Usando a métrica pré-definida, avalia-se o desempenho das predições realizadas
 - ▶ Ex: acurácia de $5/6 = 83.3\%$

Teste do modelo

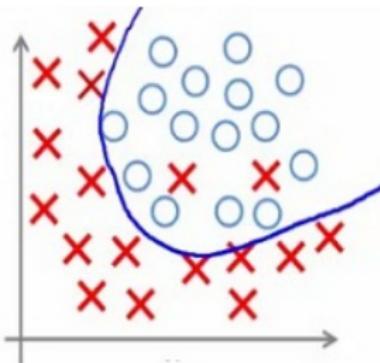
- ▶ Se o conjunto de teste é suficientemente grande, representativo e estatisticamente independente do modelo treinado, o desempenho no conjunto de teste produz uma boa estimativa do desempenho real do modelo
- ▶ Caso o desempenho seja insatisfatório, pode-se iterar o desenvolvimento do modelo, isto é: desenvolver melhores atributos, escolher outro (tipo de) modelo, e treiná-lo novamente
- ▶ No entanto, deve-se evitar realizar muitas iterações de desenvolvimento usando o mesmo conjunto de teste, caso contrário viola-se a hipótese de independência e o desempenho não será mais representativo
 - ▶ Nesse caso, deve-se coletar um novo conjunto de teste
- ▶ Em geral, utilizar um conjunto de teste permite detectar overfitting

Underfitting e Overfitting

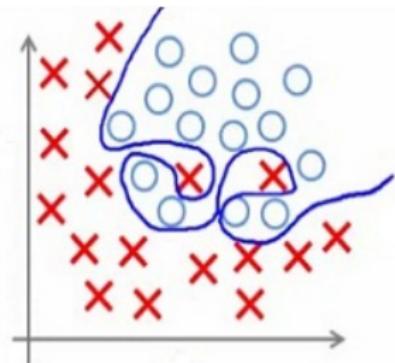


Under-fitting

(too simple to
explain the
variance)



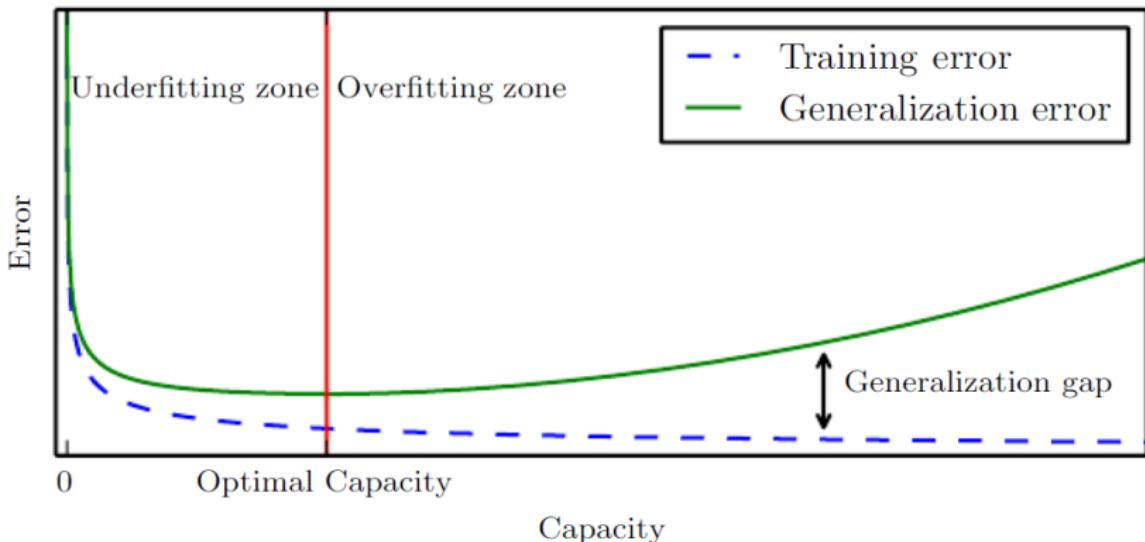
Appropriate-fitting



Over-fitting

(forcefitting -- too
good to be true)

Underfitting e Overfitting: Tradeoff



- ▶ **Capacidade** do modelo refere-se à capacidade de representar com precisão o conjunto de treinamento
- ▶ Está associada à complexidade do modelo, por exemplo, número de parâmetros treináveis

Plano de Ensino

Aulas

1. Apresentação da disciplina / Python
2. Regressão linear (2 aulas)
3. Regressão logística
4. Máquinas de vetores de suporte (SVM)
5. Redes neurais e redes profundas (2 aulas)
6. Aprendizado não-supervisionado (clustering e PCA)
7. Detecção de anomalias e sistemas de recomendação
8. Redes convolucionais
9. Redes recorrentes
10. Tópicos adicionais

Metodologia

- ▶ Primeira parte da aula: resolução do exercício da aula anterior
 - ▶ Apresentado pelos alunos com correção pelo professor
- ▶ Segunda parte da aula: exposição da teoria
 - ▶ Será passado um exercício computacional para ser feito em casa
- ▶ Total de 10 exercícios, valendo ao todo 50% da nota final
 - ▶ Cada exercício entregue **NO PRAZO** (mesmo que incompleto ou com erros) vale 2,5%
 - ▶ O conjunto dos 10 exercícios (completos e corretos) até o fim da disciplina vale os 25% restantes

Projeto Final

- ▶ Corresponde aos demais 50% da nota final
- ▶ A ser feito **preferencialmente em dupla**
- ▶ Consiste de:
 - ▶ Relatório em formato de artigo
 - ▶ Apresentação de slides
 - ▶ Implementação comentada
- ▶ Opções:
 - ▶ Técnica abordada na disciplina \implies enfoque na aplicação (de interesse do aluno)
 - ▶ Técnica não abordada na disciplina \implies a aplicação pode ser mais modesta, para permitir mais ênfase na teoria

Software

- ▶ Python 3
- ▶ Bibliotecas científicas (numpy, matplotlib, scikit-learn, keras)
- ▶ Jupyter notebook
- ▶ Recomendável instalar a distribuição Anaconda

Recursos

- ▶ Livro: *Machine Learning Refined: Foundations, Algorithms, and Applications* (Watt et al., 2016)
- ▶ Outros livros: James, Bishop, Goodfellow, Chollet, ...
- ▶ Tutoriais online
- ▶ Cursos online: Coursera, edX, fastai, ...
- ▶ Fórum online (e fórum de dúvidas da disciplina)
- ▶ Disciplina INE410121 - Visão Computacional (2019.2)