

# Reconhecimento de Padrões em Imagens

## Introdução

Dainf - UTFPR

Leyza Baldo Dorini - Rodrigo Minetto

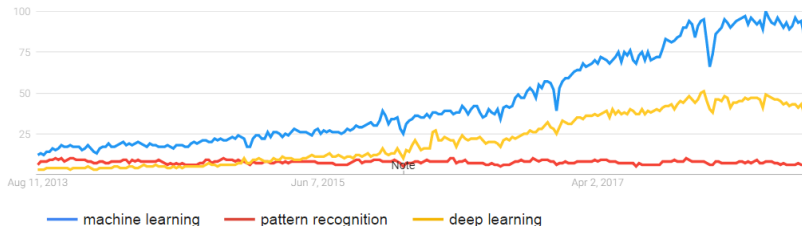
## Reconhecimento de padrões

Tem como objetivo identificar características relevantes em objetos ou eventos que permitam classificá-los em categorias. Os algoritmos utilizados são baseados em diferentes técnicas, as quais visam não apenas um processo de “memorização”, mas sim de “generalização”.

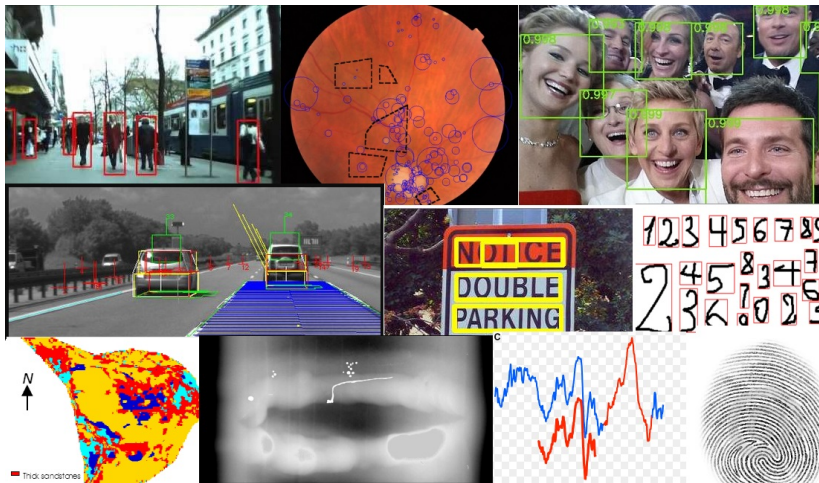
Foco do curso: aplicações de processamento e análise de imagens.

# Pattern recognition $\times$ Machine/Deep Learning

- *Machine Learning*  $\times$  *Pattern Recognition*: sinônimos, escolas de pensamento (anos 70/90), PR é um ramo de ML...
- *Deep Learning*: redes neurais com mais camadas (utilização vinculada à disponibilidade de dados e poder computacional - GPU), “*current state-of-the-art*”, ...
- Google trends:



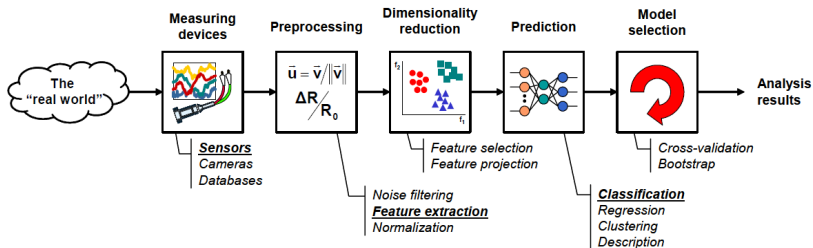
# Reconhecimento de padrões



Algoritmos de RP “dão aos computadores a habilidade de aprender algo para o qual não foram explicitamente programados”.

# Etapas básicas

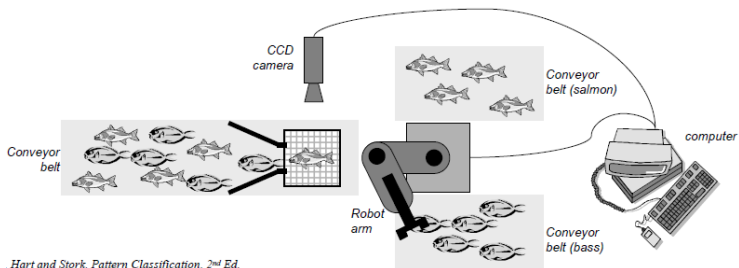
Um sistema básico é composto pelos seguintes componentes:



- Os estágios são dependentes entre si e devem ser otimizados.
- Agregar conhecimento *a priori* sobre o processo pode ser essencial.

# Exemplo: salmão $\times$ robalo

Suponha que uma câmera seja colocada na correia onde são transportados peixes pertencentes a duas classes: salmão e robalo. O objetivo é construir um sistema de reconhecimento de padrões.



# Aquisição das imagens de entrada

Dependendo da forma de aquisição, é preciso considerar variações nas imagens de entrada.



Inicialmente: bases de dados públicas! Exemplos:

- <http://www.primaresearch.org/datasets>
- <http://datasets.visionbib.com/index.html>
- <http://mlr.cs.umass.edu/ml/>

O pré-processamento é uma etapa fundamental em grande parte dos problemas práticos. Seu objetivo é simplificar a execução das tarefas subsequentes ao:

- eliminar informações que não são relevantes (tais como ruídos ou variações nas condições de iluminação);
- selecionar as regiões de interesse (segmentação);
- padronizar as amostras de entrada (por exemplo, o alinhamento);
- etc.



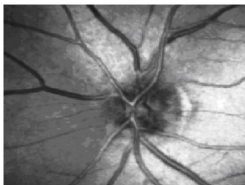
# Extração de características

## Extração de características

Visa obter medidas numéricas que permitam caracterizar um objeto ou evento.

Dependem do problema sendo tratado, bem como podem ter diferentes níveis de abstração.

Gray Level Retina Image

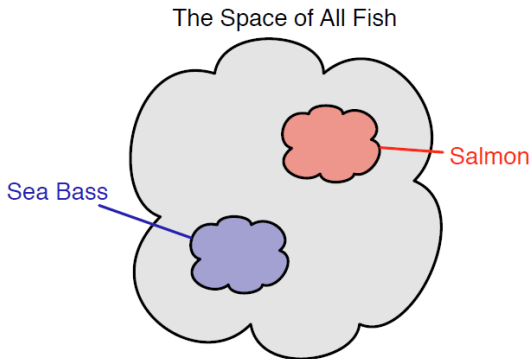


$$\longrightarrow f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & \dots & f(0, 199) \\ f(1, 0) & f(1, 1) & \dots & f(1, 199) \\ \vdots & \vdots & & \vdots \\ f(149, 0) & f(149, 1) & \dots & f(149, 199) \end{pmatrix}$$

$$f(x, y) \in \{0, 1, 2, \dots, 255\}$$

$\longrightarrow$  Papilla Shape - OK  
Blood Vessel Width - OK

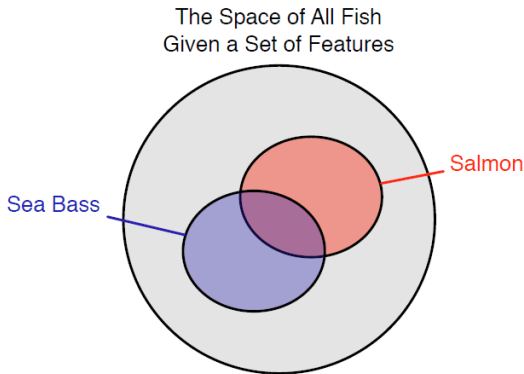
# Modelos e padrões: o espaço ideal



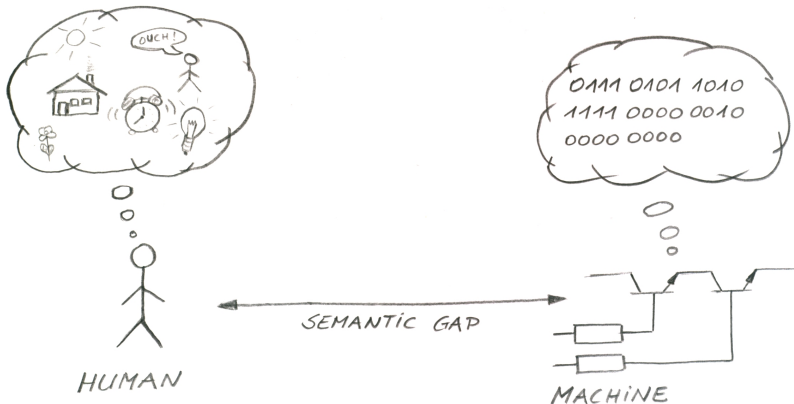
Considerando o espaço de todos os peixes, as populações de robalo e salmão são distintas. Contudo, são contempladas muitas características, o que inviabilizaria sua sistematização (algumas não podem ser medidas com a câmera, por exemplo).

# Modelos e padrões: o espaço real

Ao selecionar algumas características, estamos fazendo uma projeção em um espaço de menor dimensão.



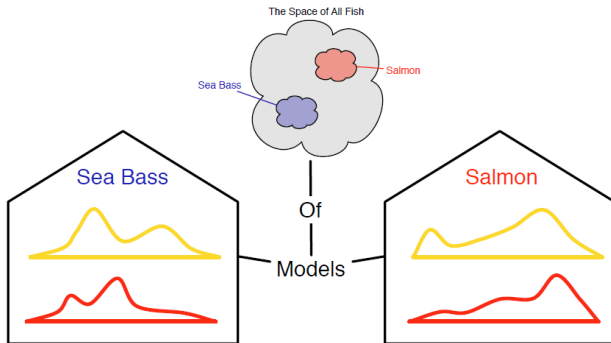
# Modelos e padrões: *Semantic Gap*



*" (...) the gap means the difference between ambiguous formulation of contextual knowledge in a powerful language (e.g. natural language) and its sound, reproducible and computational representation in a formal language (e.g. programming language)"*

# Modelos e padrões

Para cada fenômeno que precisa ser classificado é preciso construir um modelo, ou seja, uma representação aproximada das características selecionadas.



A seleção das características é essencial!

# Seleção de características

Quais características você sugere?



(a) Salmão



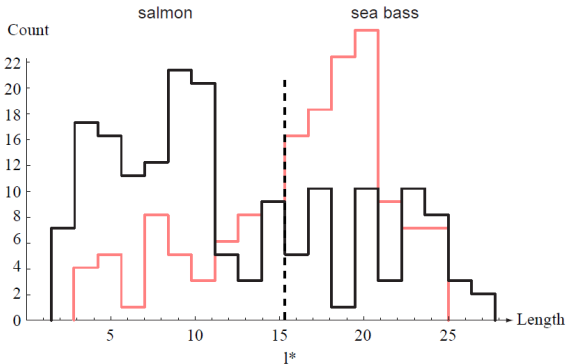
(b) Robalo

Além de agrupar objetos de uma mesma classe, as características devem permitir a separação inter-classes. Além disso, pode ser desejado considerar invariância às transformações que não são relevantes (escala e orientação, por exemplo).

# Seleção de características: comprimento

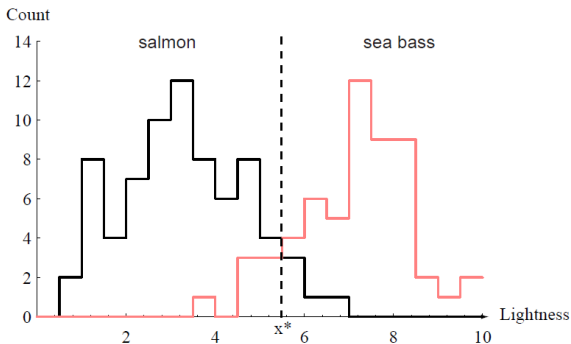
Para analisar a característica, podemos selecionar alguns peixes na esteira e realizar a medição.

- Esses exemplos são parte do conjunto de treinamento.
- É preciso assegurar que são amostras representativas.
- A característica é analisada por meio da construção de histogramas: distribuição marginal.



# Seleção de características: brilho

Essa característica apresenta uma melhor separação entre as duas classes.

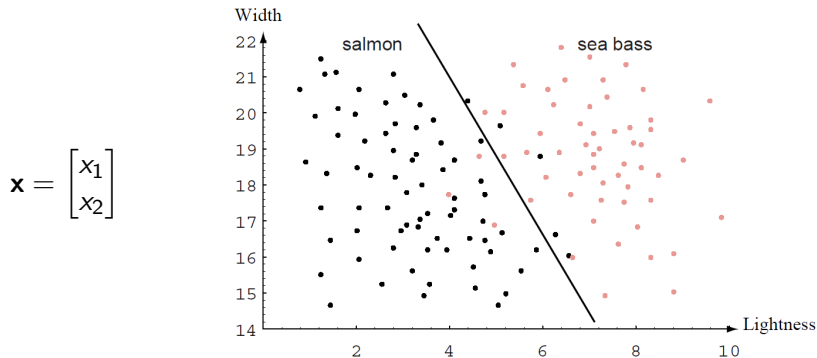


Mesmo assim, nenhum limiar  $x^*$  consegue delimitar uma fronteira de decisão sem erros.



# Combinando características

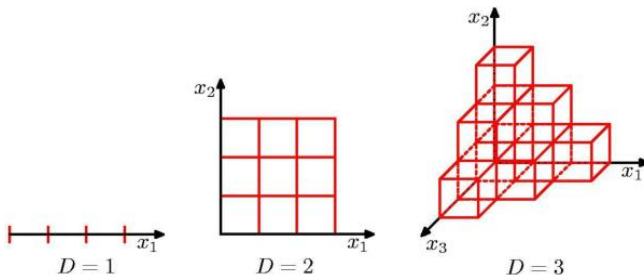
Raramente uma única característica é suficiente. Para o exemplo dos peixes, podemos considerar um **vetor de características** (ou seja, trabalhar um **espaço de características 2D**).



Ao considerar a linha como fronteira de decisão, o erro seria menor do que considerar as características individualmente (observe o *scatter plot*).

# Curse of Dimensionality

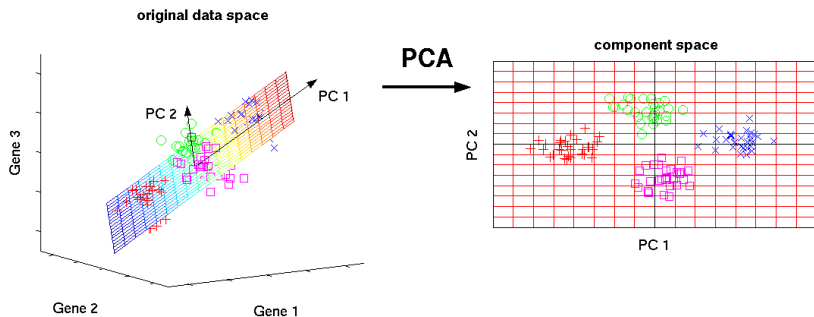
Conforme o número de características (ou dimensões) aumenta, a quantidade de dados necessária para a generalização precisa cresce exponencialmente.



- Como saber quais características tem um melhor desempenho?
- Como lidar com redundância?

# Redução da dimensionalidade

Existem abordagens que permitem reduzir a dimensionalidade das características utilizadas. Por exemplo, a *Principal Component Analysis* (PCA) é uma transformação que leva os dados iniciais para um novo sistema de coordenadas.



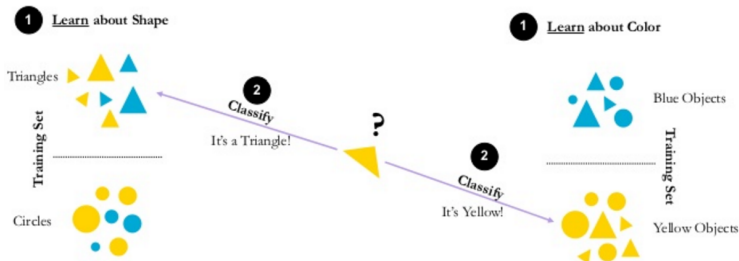
## Classificação

Atribuir uma categoria a um objeto com base nas medidas presentes no vetor de características.

A grande questão é: construir classificadores com capacidade de generalização, ou seja, que sejam capazes de classificar novas entradas com erro similar àquele obtido com as amostras conhecidas.

# Estratégias de aprendizagem: supervisionada

Busca modelar as dependências e os relacionamentos entre as saídas esperadas e as características de entrada.

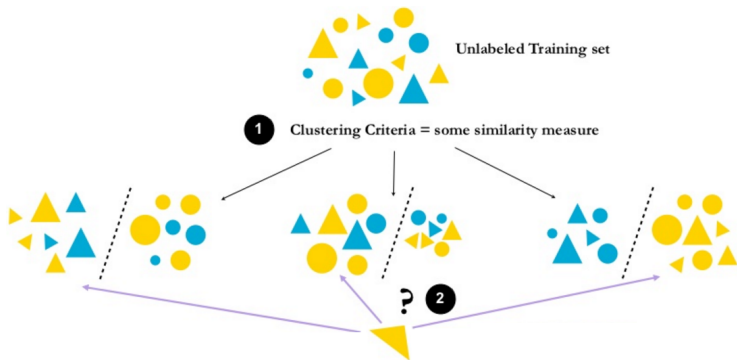


São subdivididos em:

- Algoritmos de classificação: usam as características aprendidas nos dados dados de treinamento para classificar novos dados. Os valores de saída são discretos.
- Algoritmos de regressão: constroem um modelo baseado nas características e rótulos dos dados de treinamento, o qual é utilizado para prever valores para novos dados. Os valores de saída são contínuos.

# Estratégias de aprendizagem: não-supervisionada

Neste caso, os rótulos não estão disponíveis.



Os algoritmos usam técnicas para buscar por regras e padrões que permitam sumarizar e agrupar dados de tal forma a permitir a identificação de padrões significativos que permitam descrever os dados de forma adequada (classificação). Nesta categoria, destacam-se as abordagens de agrupamento (*clustering*).

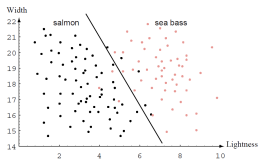
- **Aprendizagem supervisionada.**
- **Aprendizagem não-supervisionada.**
- **Aprendizagem semi-supervisionada:** utiliza as duas estratégias anteriores.
- **Reinforcement learning:** utiliza as informações coletadas a partir da interação com o meio para tomar decisões que maximizam a recompensa ou minimizam o risco.

# Fronteira de decisão

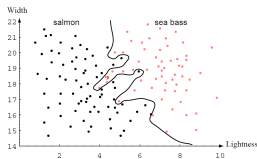
## Decision Boundary

*"A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous."*

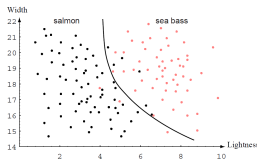
A complexidade da fronteira de decisão depende das características e da complexidade do modelo e do classificador.



(a) Linear



(b) Problema com novas amostras



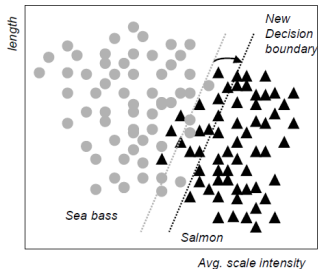
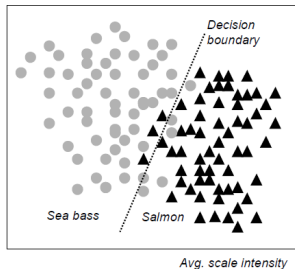
(c) Desempenho  $\times$  Simplicidade



# Funções objetivo

O processo de aprendizagem de um classificador pode ser interpretado como a busca por regiões de decisão que otimizam a função objetivo.

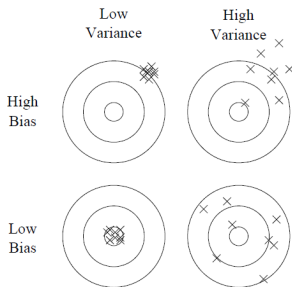
- *Minimum error rate*: busca minimizar o erro de classificação, ou seja, a quantidade de atribuições incorretas de uma classe a um padrão.
- *Total expected cost*: minimização de risco (a questão é como estimar o risco aceitável).



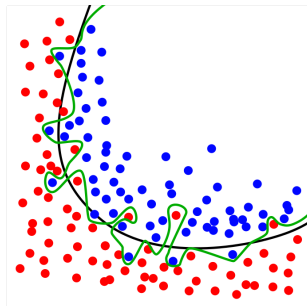
# Overfitting e Underfitting

O erro de generalização pode ser decomposto em bias e variância:

- Bias: tendência de aprender consistentemente algo errado. Um bias alto pode fazer com que o algoritmo perca relações relevantes entre características e as saídas esperadas (é o chamado *underfitting*).
- Variância: tendência de aprender coisas aleatórias. Valores altos causam *overfitting*.



(a) Exemplo



(b) Overfitting

# No Free Lunch Theorem

Qual é o melhor classificador? Não há consenso. Isso é descrito pelo *No Free Lunch Theorem*<sup>1</sup>!

- Em resumo, significa que “não é possível definir uma estratégia de otimização que seja de uso universal”. Uma estratégia é superior a outra se adaptada à estrutura de um problema específico.
- No contexto de reconhecimento de padrões, ressalta que é preciso focar nos aspectos mais relevantes (distribuição dos dados, função de custo, quantidade de dados de treinamento...)

---

<sup>1</sup>(Wolpert, 1996)(Wolpert e Macready, 1997)

Em diversas situações, a combinação de múltiplos modelos pode trazer benefícios na resolução de um problema. Diversas estratégias podem ser consideradas:

- Combinação: classificadores são combinados utilizando uma regra qualquer (voto majoritário, por exemplo).
- *Ensembles* (Conjuntos): classificadores são criados automaticamente através de técnicas de aprendizagem de máquina (*bagging e boosting*, por exemplo).
- Seleção Dinâmica de Classificadores: busca selecionar em função dos outros classificadores aquele que melhor classifica um dado padrão.

Existem diferentes métodos para avaliação do desempenho de um classificador.

- Precisão
- Revocação
- Matrix de confusão
- ROC
- ...

Em breve veremos com mais detalhes.