Jaime S. Cardoso
jaime.cardoso@fe.up.pt

INESC TEC and **Faculdade de Engenharia, Universidade do Porto**, Portugal

Introduction to Machine Learning

PDEEC – Machine Learning 2018/19
Sept 20th, 2018, Porto, Portugal

Roadmap

- What's Machine Learning
- Distinct Learning Problems
- For the same problem, different solutions
- Different solutions but with common traits
  - … and ingredients
- Avoiding overfitting and data memorization
- A fair judgement of your algorithm
- Some classical ML algorithms
- Beyond the classics

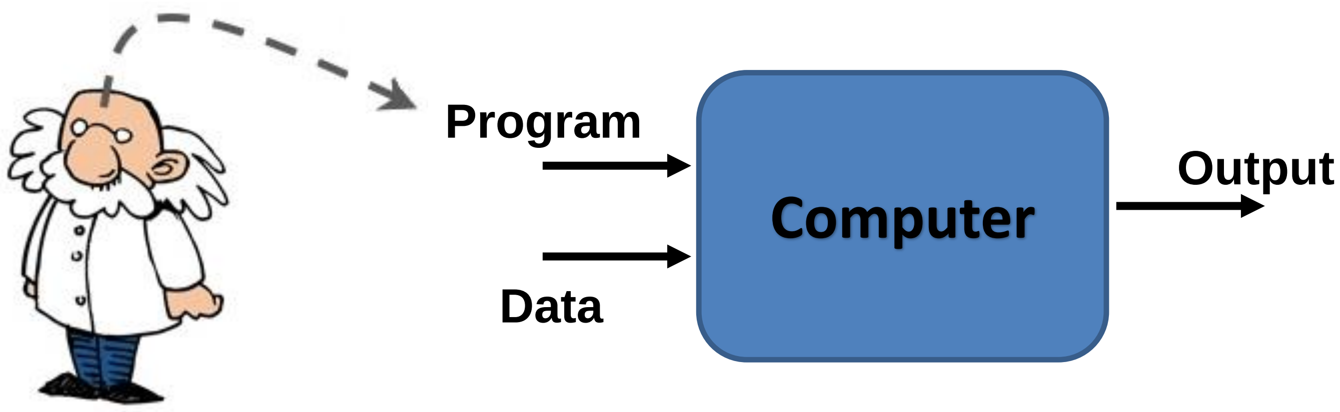
Artificial Intelligence (AI)

- “ [...automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning...” (Bellman, 1978)
- “ The branch of computer science that is concerned with the automation of intelligent behaviour.” (Luger and Stubblefield, 1993)
- “The ultimate goal of AI is to create technology that allows computational machines to function in a highly intelligent manner. (Li Deng 2018)

AI: three generations

1st wave of AI: **the sixties**

- emulates the decision-making process of a human expert



AI: three generations

1st wave of AI: **the sixties**

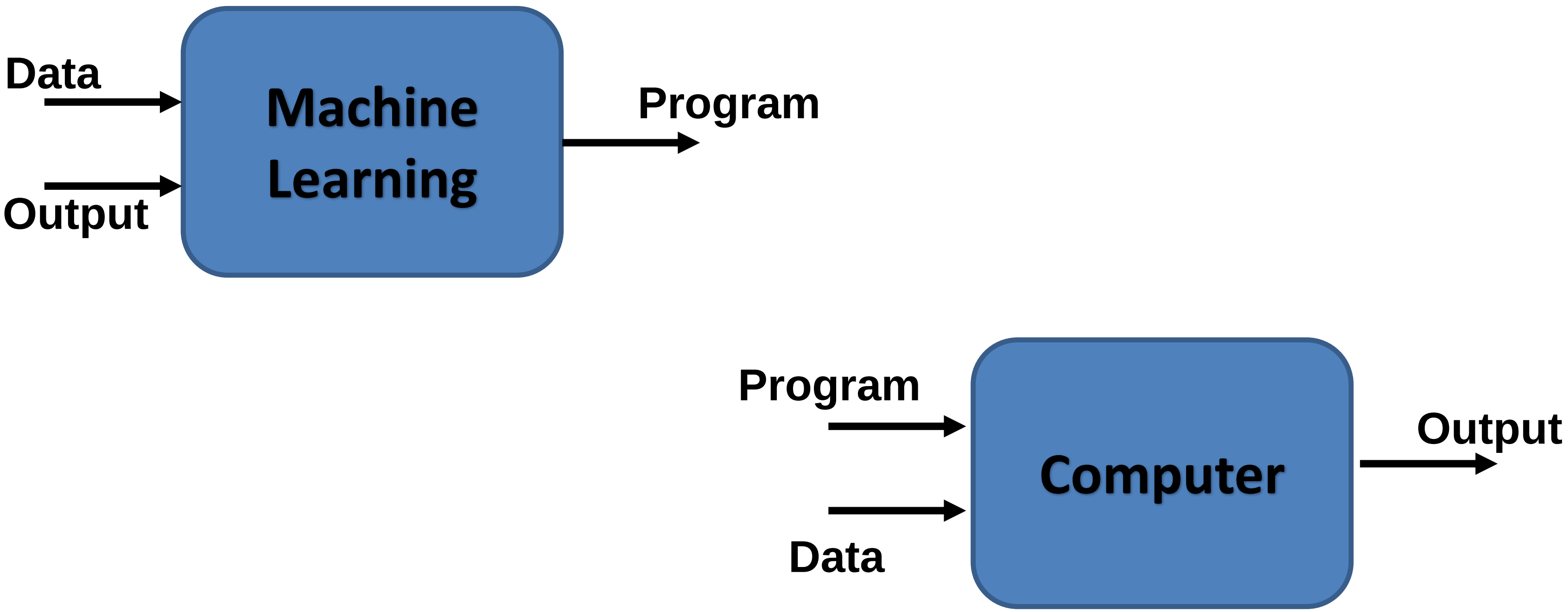
- Based on expert knowledge
  - “if-then-else”
- Effective in narrow-domain problems
- Focus on the head or most important parameters (identified in advance), leaving the “tail” parameters and cases untouched.

- Transparent and interpretable
- Difficulty in generalizing to new situations and domains
- Cannot handle uncertainty
- Lack the ability to learn algorithmically from data

AI: three generations

2nd wave of AI: **the eighties**

- Based on (shallow) machine learning



An example*

- **Problem**: sorting incoming fish on a conveyor belt according to species

- Assume that we have only two kinds of fish:
  - Salmon
  - Sea bass

Picture taken with a camera

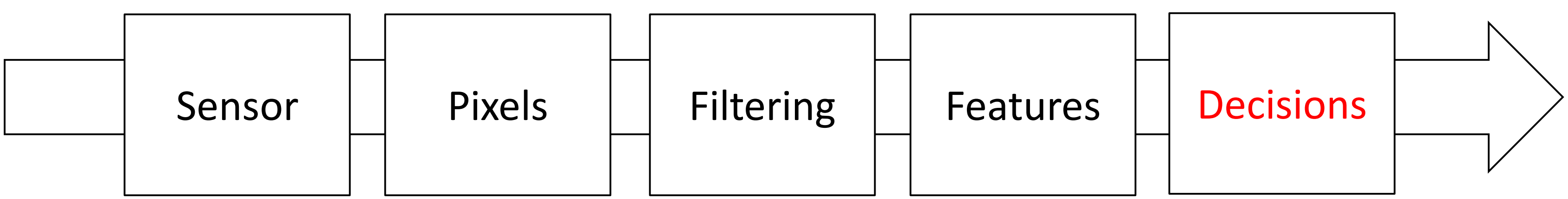
*Adapted from Duda, Hart and Stork, Pattern Classification, 2nd Ed.

An example: decision process

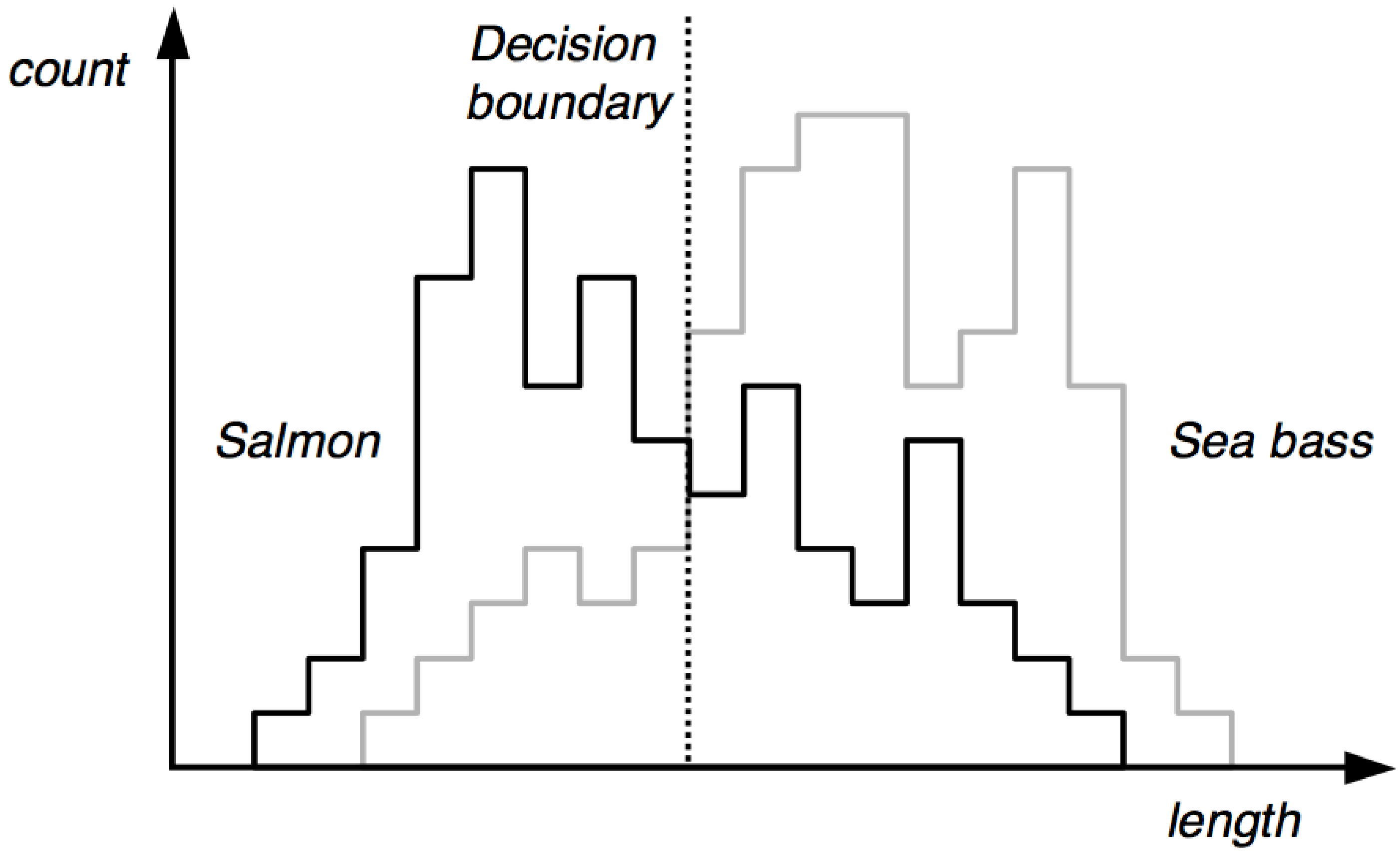
- What kind of information can distinguish one species from the other?
  - Length, width, weight, number and shape of fins, tail shape, etc.
- What can cause problems during sensing?
  - Lighting conditions, position of fish on the conveyor belt, camera noise, etc.
- What are the steps in the process?
  - Capture image -> isolate fish -> take measurements -> make decision

An example: our system

- **Sensor**
  - The camera captures an image as a new fish enters the sorting area
- **Preprocessing**
  - Adjustments for average intensity levels
  - Segmentation to separate fish from background
- **Feature Extraction**
  - Assume a fisherman told us that a sea bass is generally longer than a salmon. We can use **length** as a feature and decide between sea bass and salmon according to a threshold on length.



An example: features

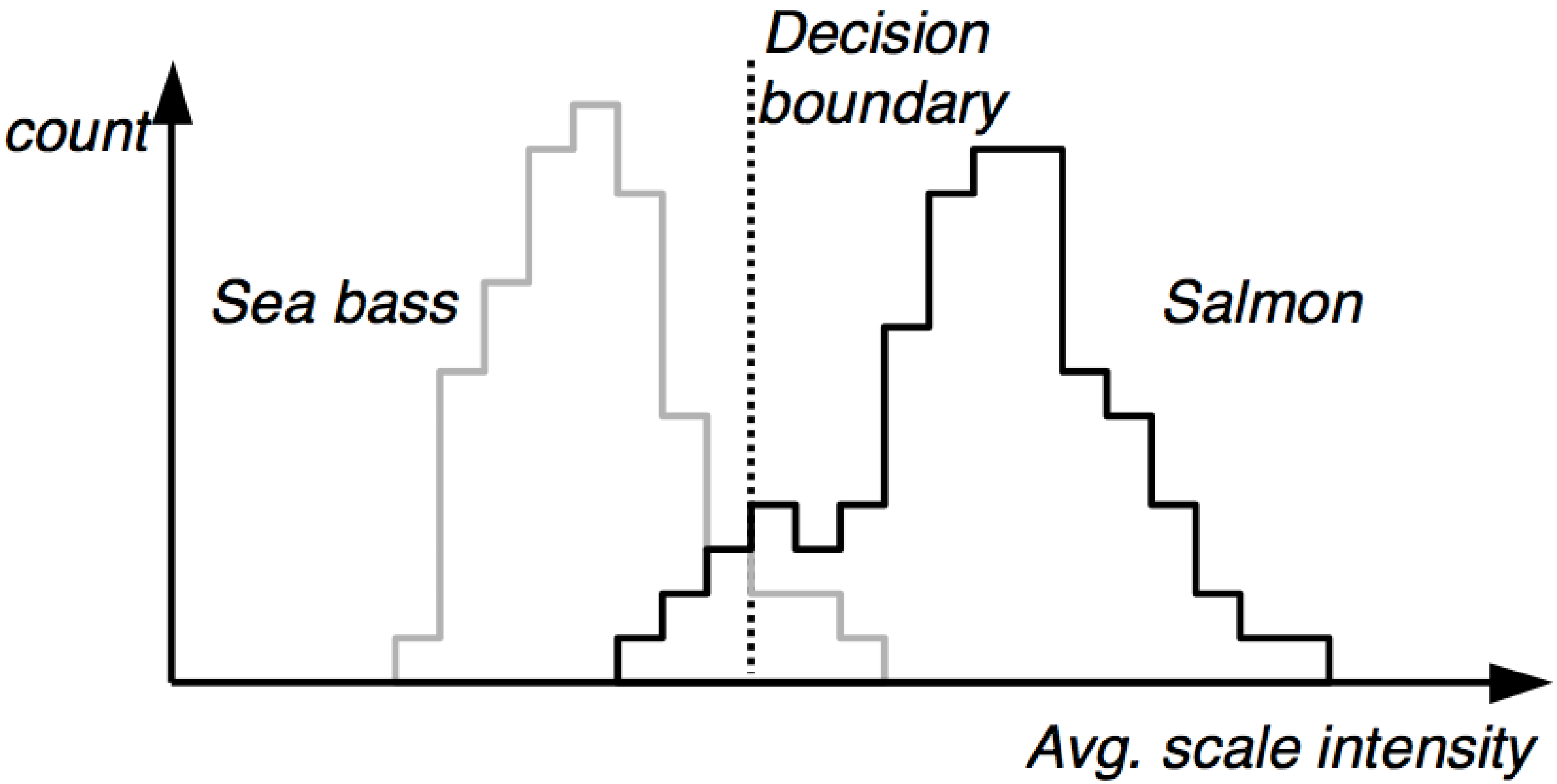


We estimate the system's probability of error and obtain a discouraging result of 40%. Can we improve this result?

An example: features

- Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold
- Committed to achieve a higher recognition rate, we try a number of features
  - Width, Area, Position of the eyes w.r.t. mouth...
  - only to find out that these features contain no discriminatory information
- Finally we find a “good” feature: **average intensity of the fish scales**

An example: features



Histogram of the lightness feature for two types of fish in **training samples**. It looks easier to choose the threshold but we still can not make a perfect decision.

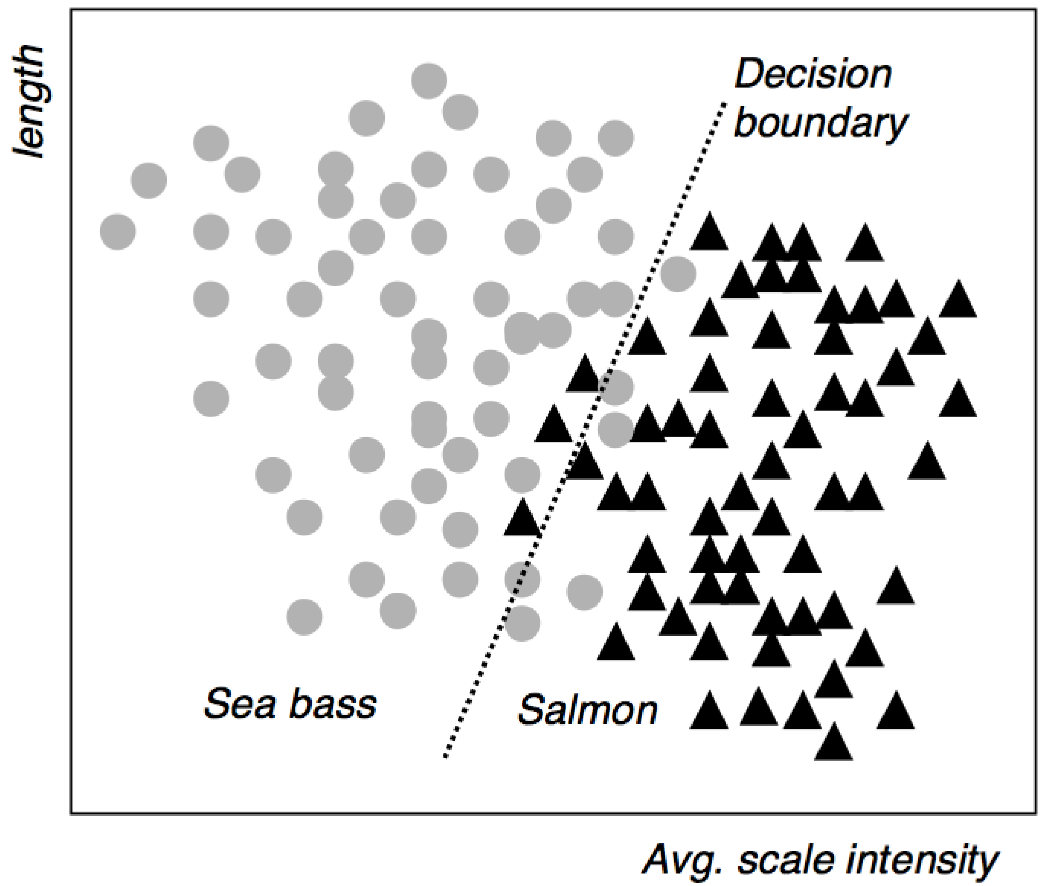
An example: multiple features

- We can use two features in our decision:
  - lightness: $\boldsymbol{x}_1$
  - length: $\boldsymbol{x}_2$
- Each fish image is now represented as a point (feature vector)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in a two-dimensional **feature space**.

An example: multiple features

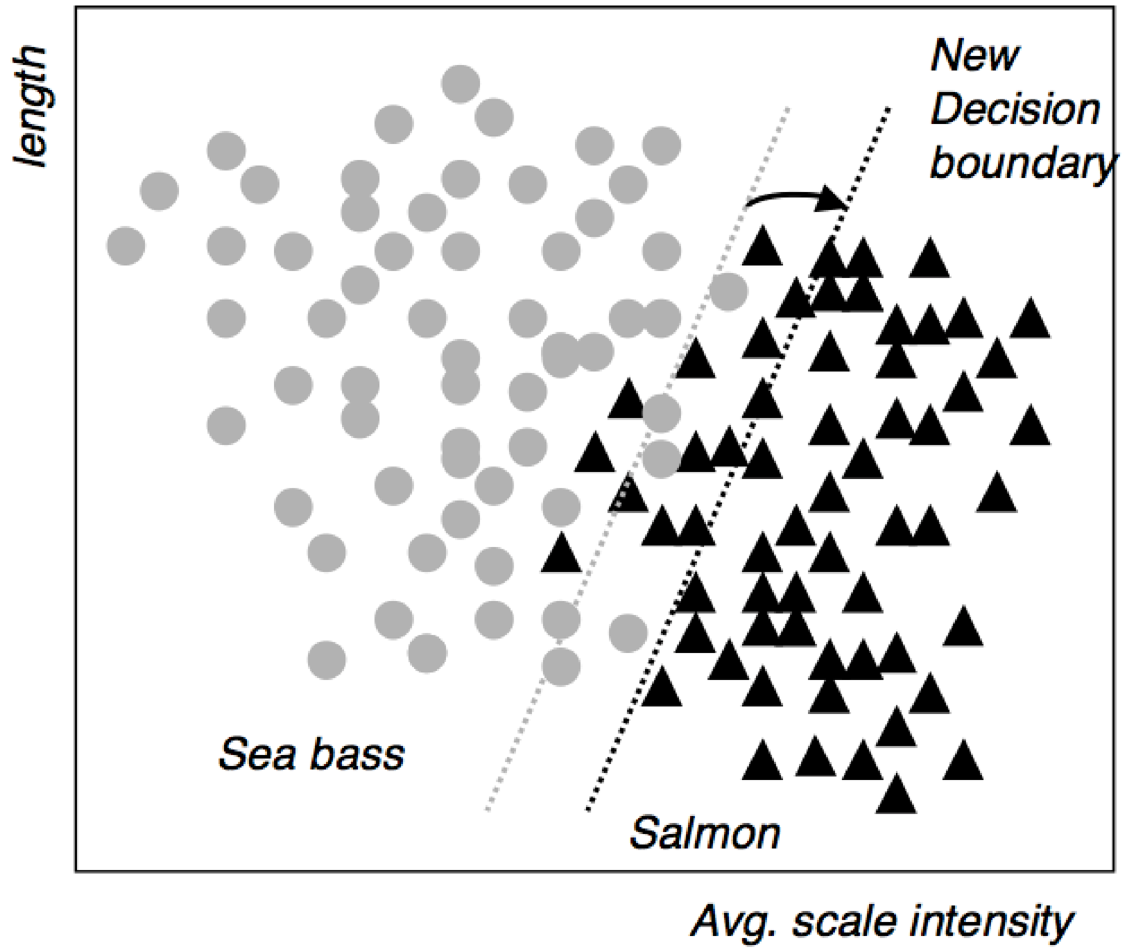


Scatter plot of lightness and length features for training samples. We can compute a **decision boundary** to divide the feature space into two regions with a classification rate of 95.7%.

An example: cost of error

- We should also consider **costs of different errors** we make in our decisions.
- For example, if the fish packing company knows that:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?

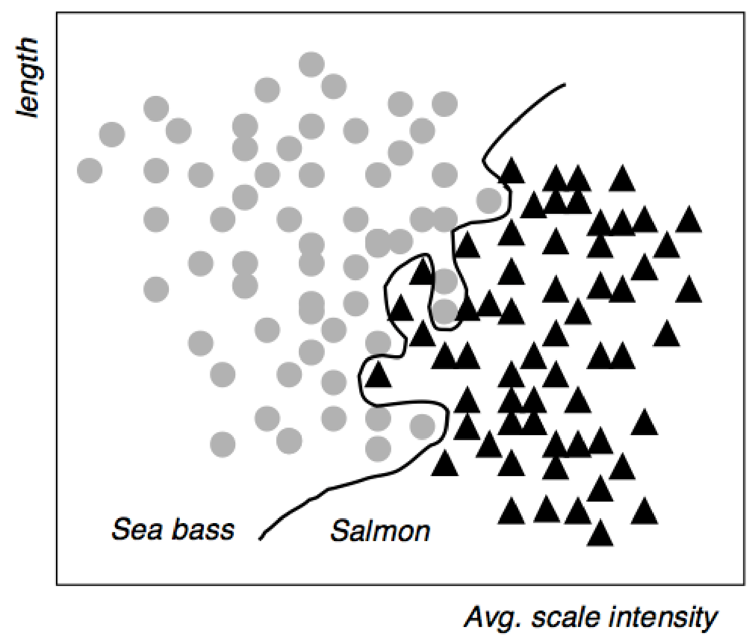
An example: cost of error



We could intuitively shift the decision boundary to minimize an alternative cost function

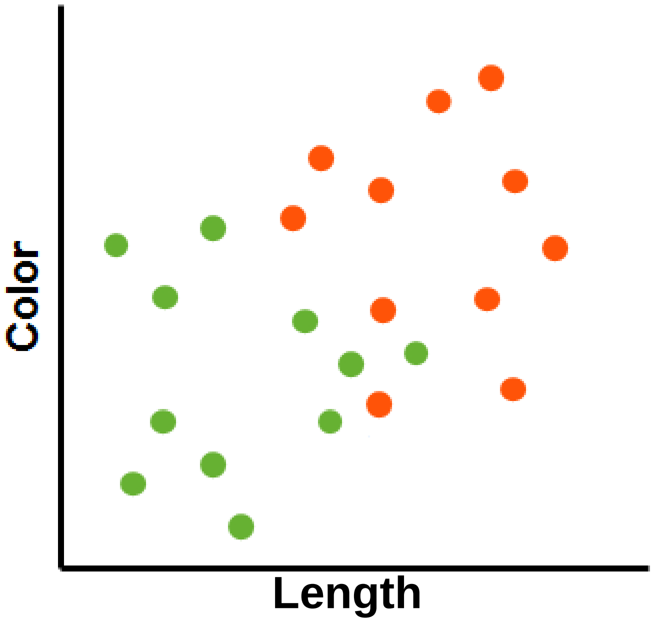
An example: generalization

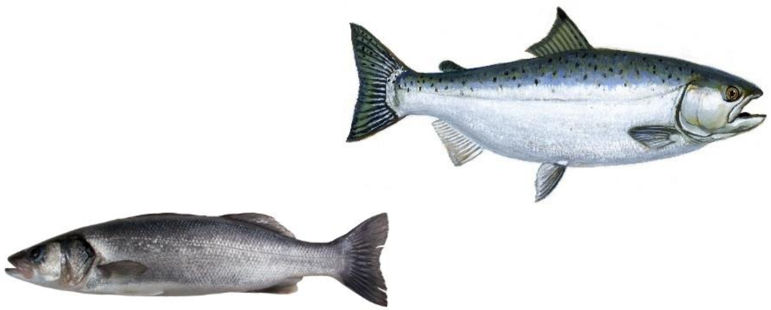
- **The issue of generalization**
  - The recognition rate of our linear classifier (95.7%) met the design specifications, but we still think we can improve the performance of the system
  - We then design a classifier that obtains an impressive classification rate of 99.9975% with the following decision boundary



Data Driven Design

- When to use?
  - Difficult to reason about a generic rule that solves the problem
  - Easy to collect examples (with the solution)



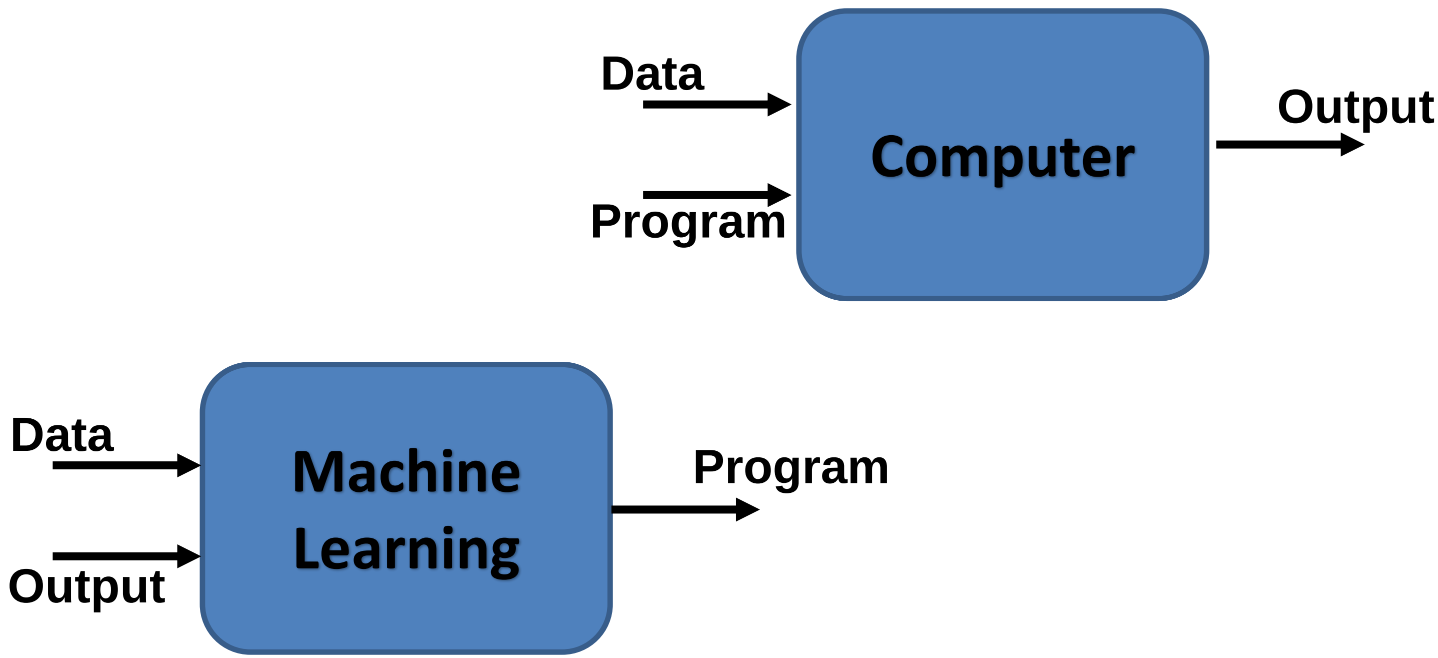


Data Driven Design

- There is little or no domain theory
- Thus the system will learn (i.e., generalize) from training data the general input-output function
  - Programming computers to use example data or past experience
- The system produces a program that implements a function that assigns the decision to any observation (and not just the input-output patterns of the training data)

What is Machine Learning?

- Automating the Automation



Data Driven Design

- A good learning program learns something about the data beyond the specific cases that have been presented to it
  - Indeed, it is trivial to just store and retrieve the cases that have been seen in the past
    - This does not address the problem of how to handle new cases, however
- Over-fitting a model to the data means that instead of general properties of the population we learn idiosyncracies (i.e., non-representative properties) of the sample.

DISTINCT LEARNING PROBLEMS

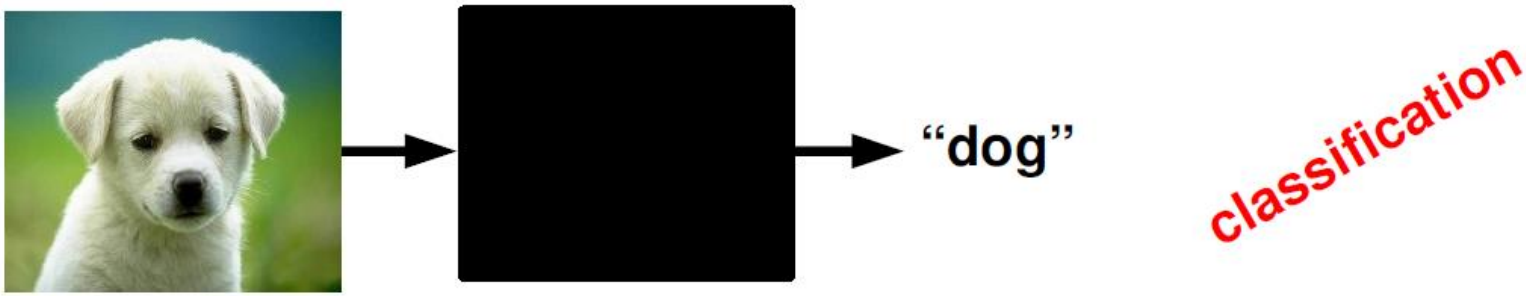
Taxonomy of the Learning Settings

Goals and available data dictate the type of learning problem

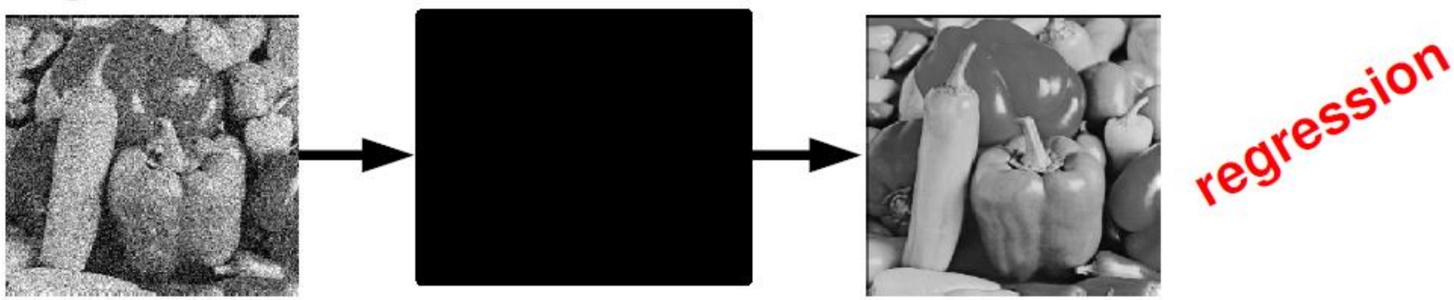
- Supervised Learning
  - Classification
    - Binary
    - Multiclass
      - Nominal
      - Ordinal
  - Regression
  - Ranking
  - Counting
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- etc.

Supervised Learning: Examples

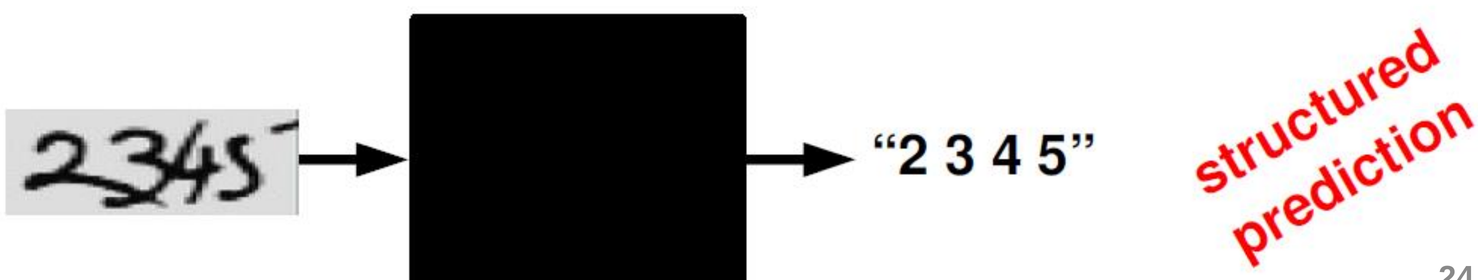
Classification



Denoising



OCR



Classification/Regression

$$y = f(\mathbf{x})$$

output **prediction function** **feature vector**

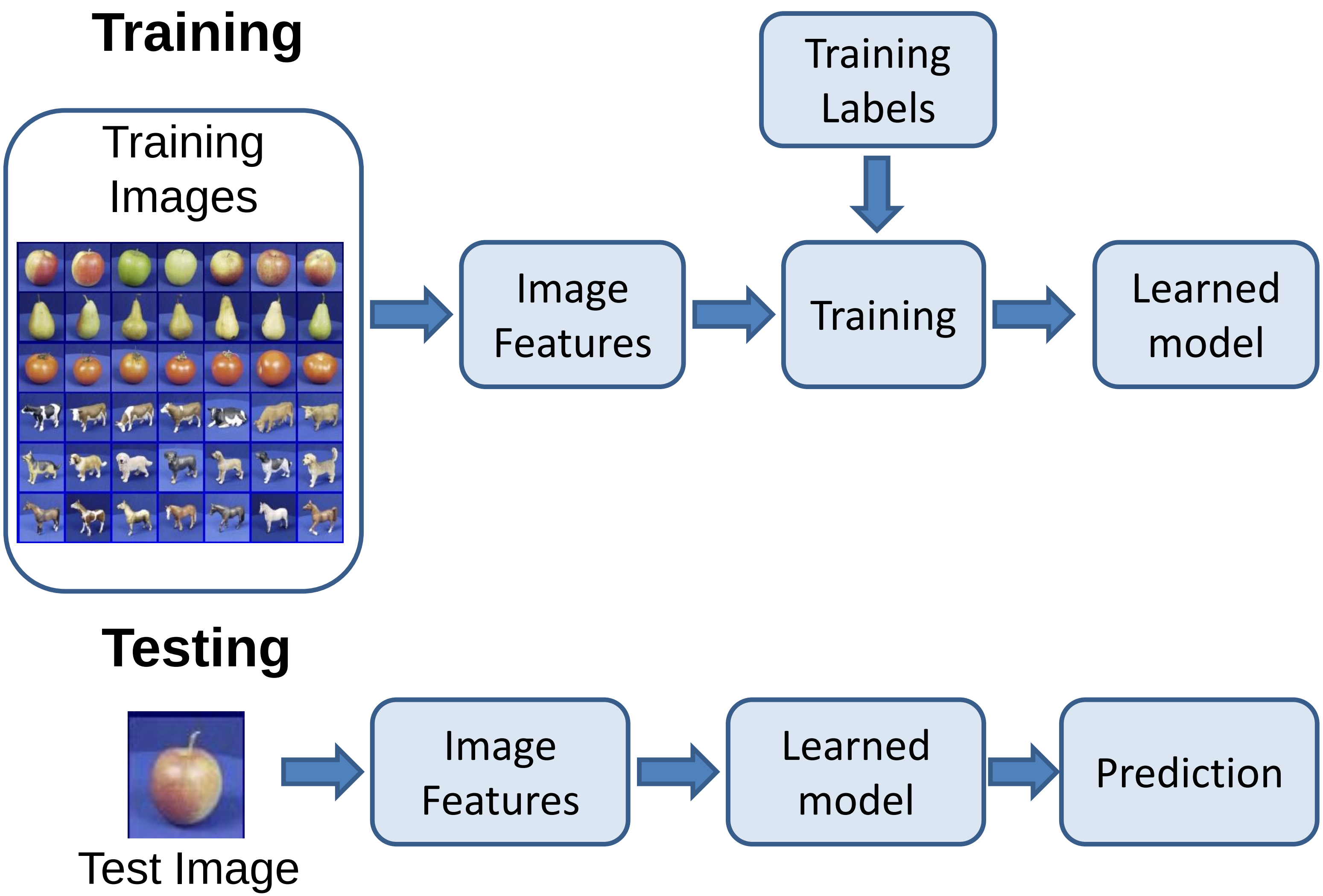
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Regression

- Predicting house price
  - Output: price (a scalar)
  - Inputs: size, orientation, localization, distance to key services, etc.

- Given a collection of labelled examples (= houses with known price), come up with a function that will predict the price of new examples (houses).

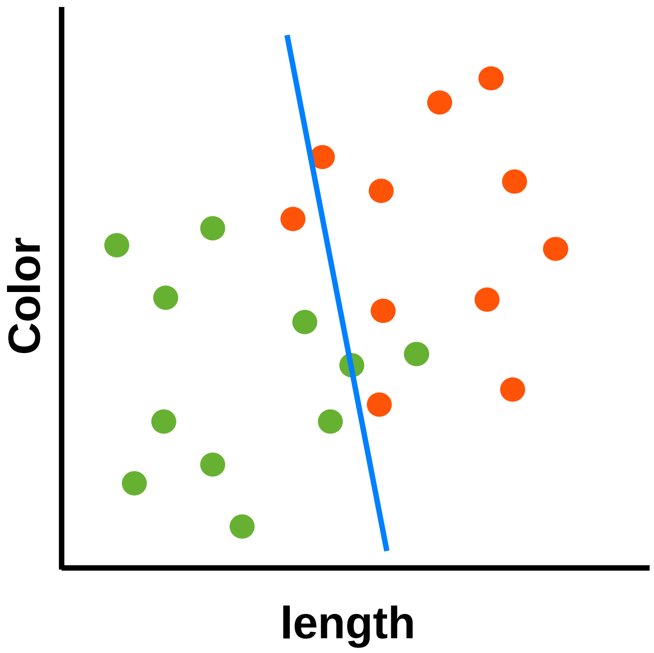
Supervised Learning in computer vision



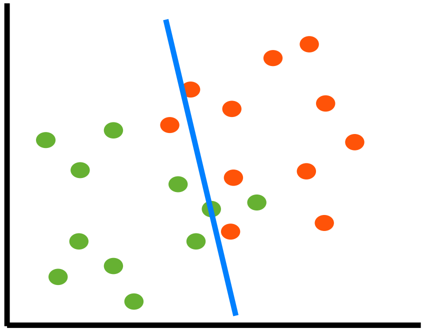
… but with common traits

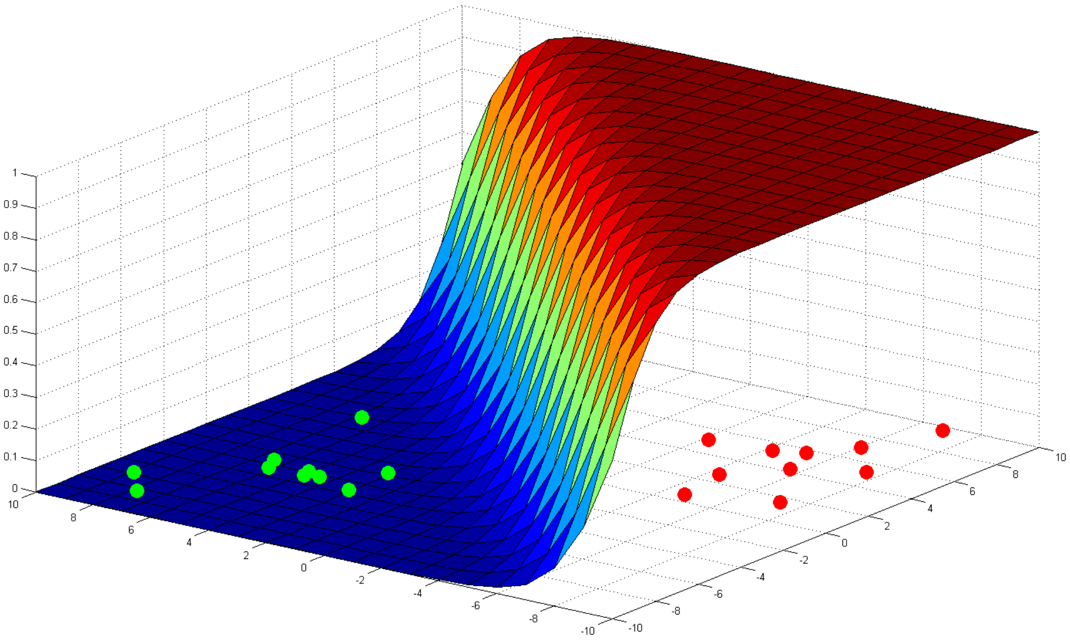
FOR THE SAME PROBLEM, DIFFERENT SOLUTIONS

Design of a Classifier

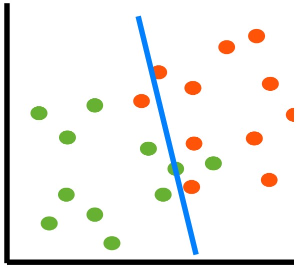


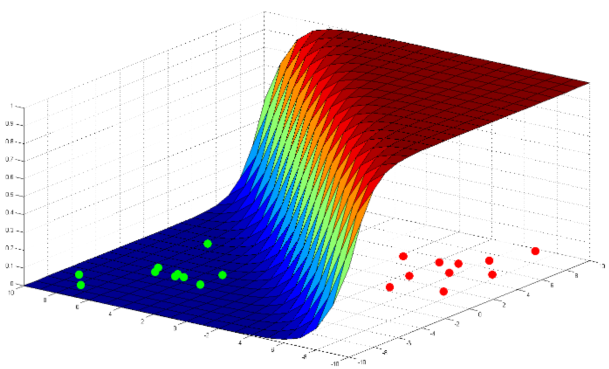
Design of a Classifier

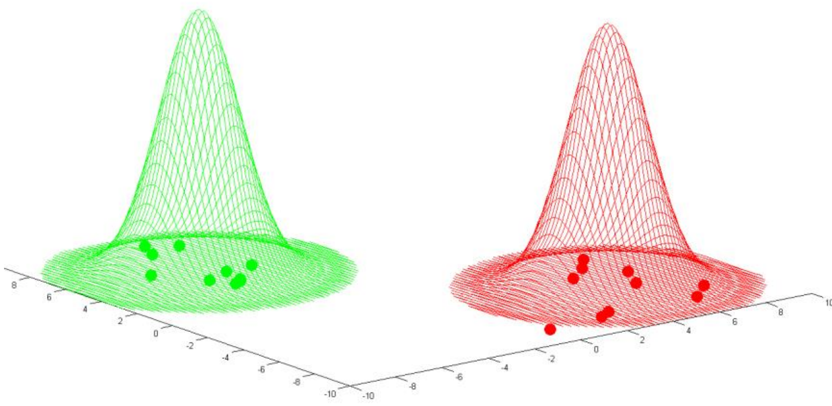




Design of a Classifier







Taxonomy of the Learning Tools

Classifier

no computation of posterior probabilities
(probability of certain class given the data)

computation of posterior probabilities

Discriminant function

Properties
- directly map each x onto a class label

Tools
- Least Square Classification
- Fisher's Linear Discriminant
- SVM
- Etc.

Probabilistic Discriminative Models

Properties
- Model posterior probabilities ($p(C_k|x)$) directly

Tools
- Logistic Regression

Probabilistic Generative Models

Properties
- model class priors ($p(C_k)$) & class-conditional densities ($p(x|C_k)$)
- use to compute posterior probabilities ($C_k|x$))

Tools
- Bayes

Pros and Cons of the three approaches

- Discriminant Functions are the most simple and intuitive approach to classify data, but do not allow to
  - compensate for class priors (e.g. class 1 is a very rare disease)
  - minimize risk (e.g. classifying sick person as healthy more costly than classifying healthy person as sick)
  - implement reject option (e.g. person cannot be classified as sick or healthy with a sufficiently high probability)

Pros and Cons of the three approaches

- Generative models provide a probabilistic model of *all* variables that allows to synthesize new data and to do novelty detection but
  - generating all this information is computationally expensive and complex and is not needed for a simple classification decision

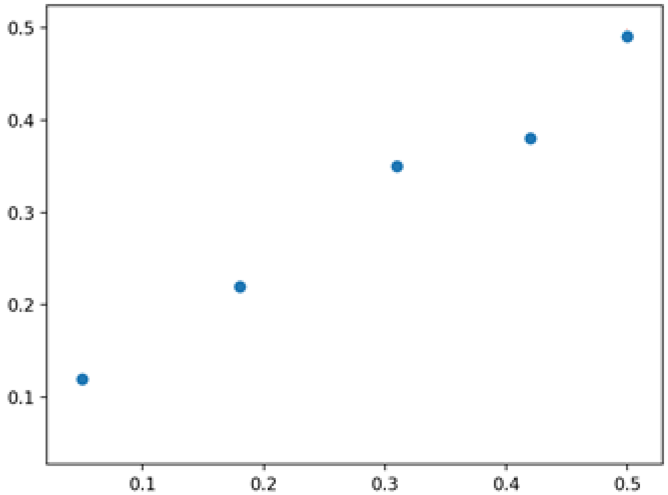
- Discriminative models provide a probabilistic model for the target variable (classes) conditional on the observed variables
  - this is usually sufficient for making a well-informed classification decision without the disadvantages of the simple Discriminant Functions

DIFFERENT SOLUTIONS BUT WITH COMMON INGREDIENTS

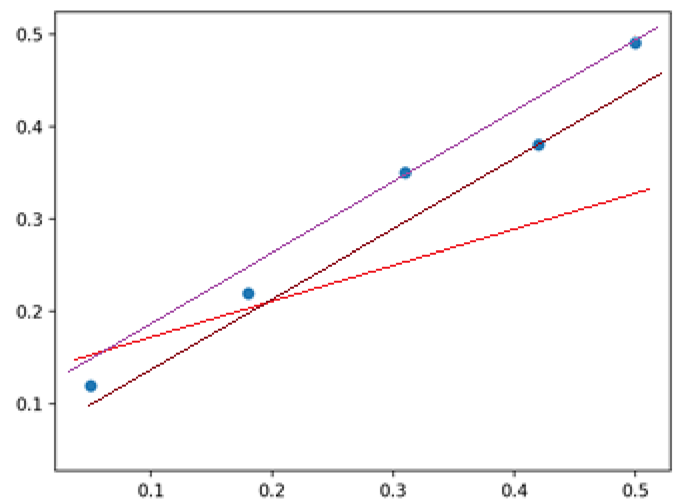
Common steps

- The learning of a model from the data entails:
  - **Model representation**
  - **Evaluation**
  - **Optimization**

Linear Regression

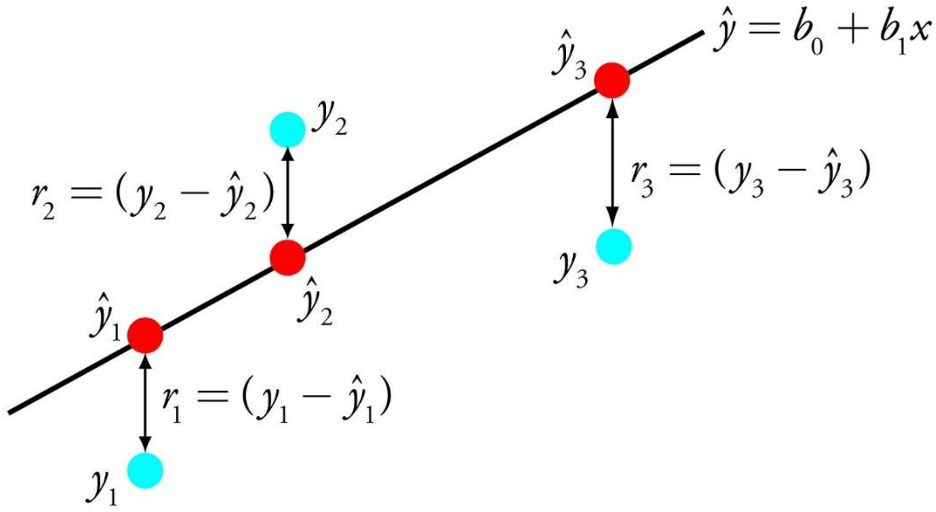


- Model Representation



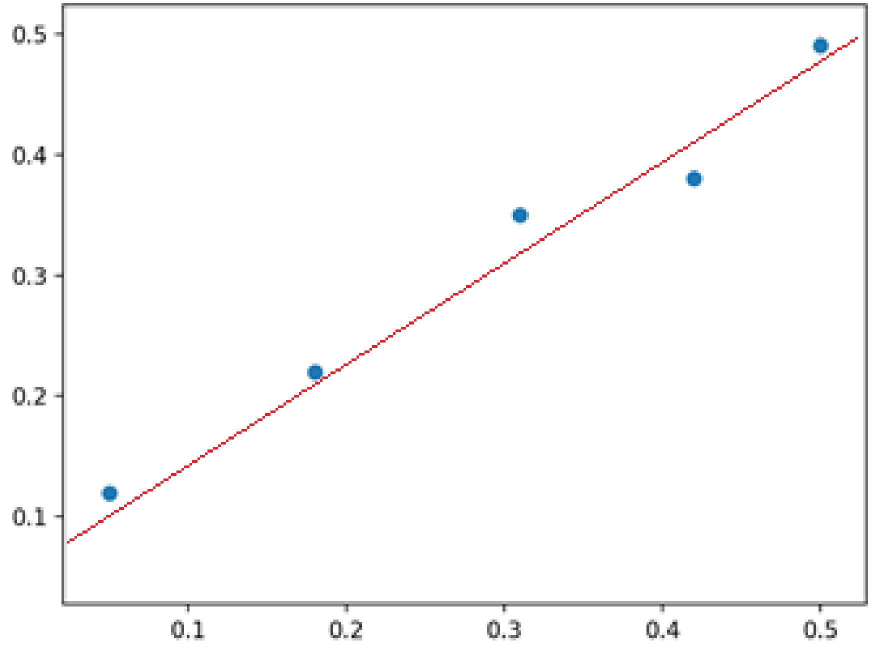
Linear Regression

- Evaluation



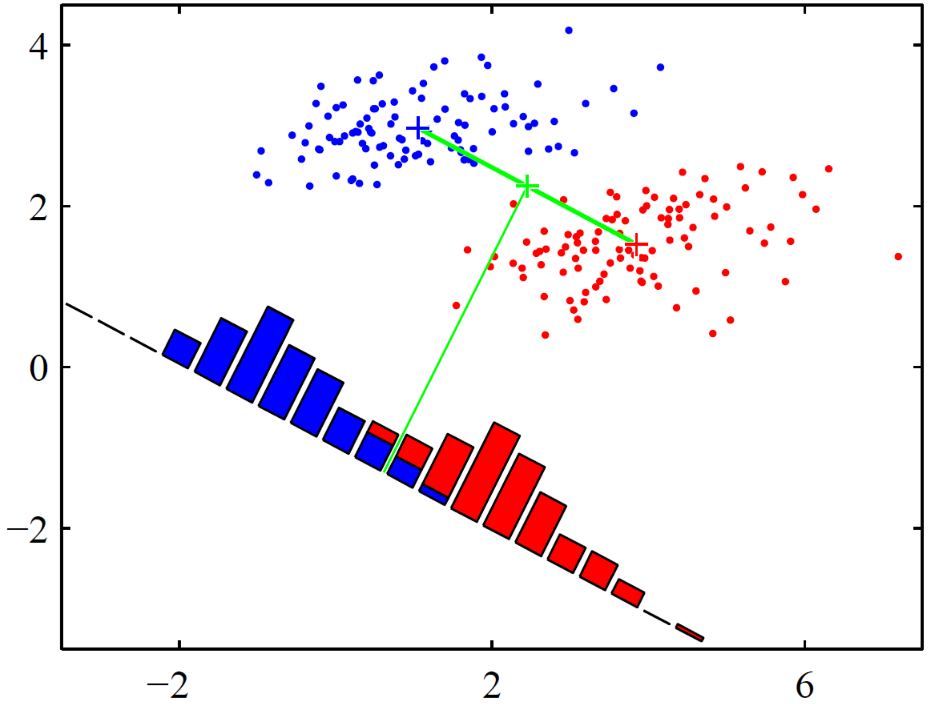
Linear Regression

- Optimization: finding the model that maximizes our measure of quality

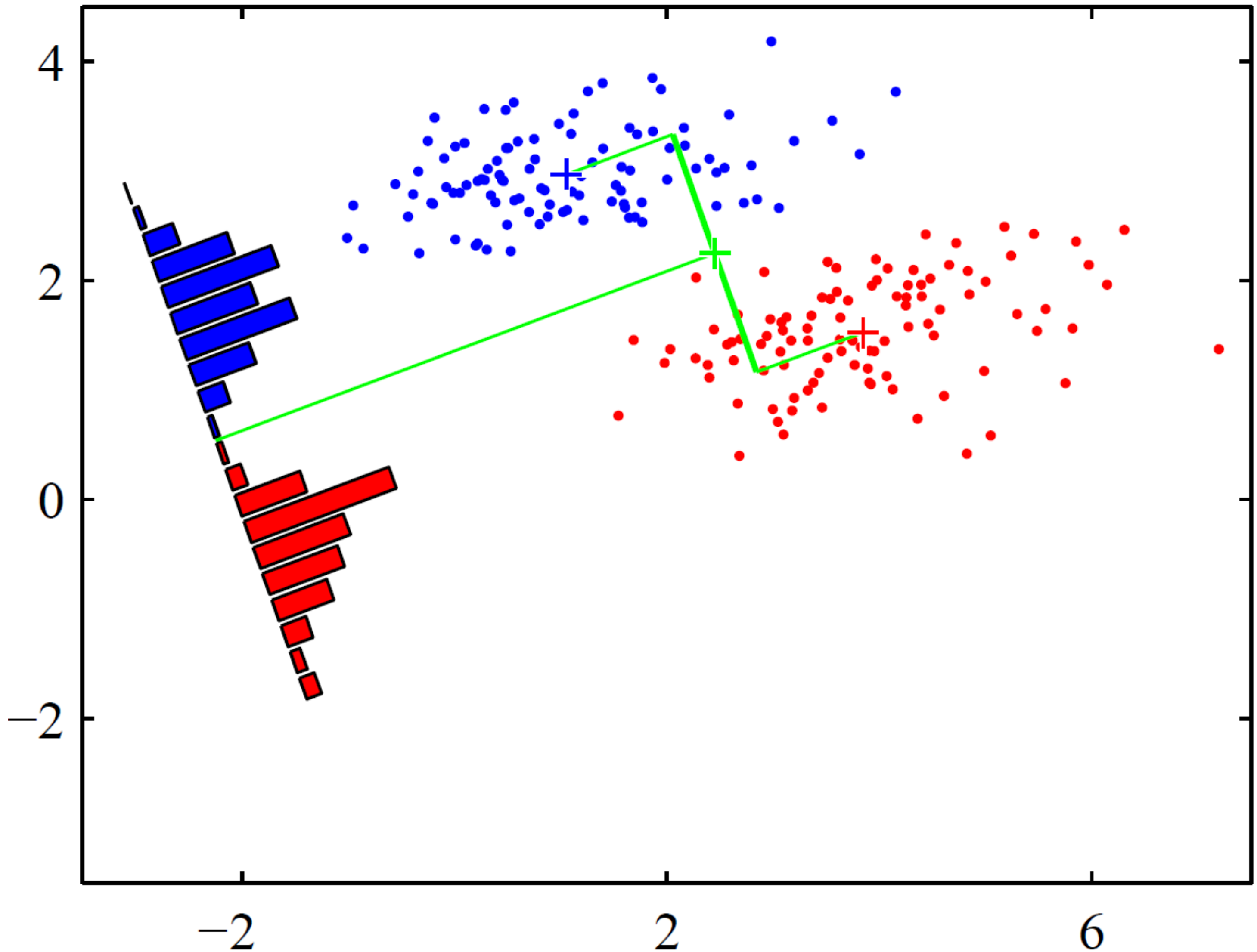


Let's design a classifier

- Use the (hyper-)plane orthogonal to the line joining the means
  - project the data in the direction given by the line joining the class means



Let’s design a classifier



Fisher's linear discriminant

- Every algorithm has three components:
  - **Model representation**
  - **Evaluation**
  - **Optimization**
- Model representation: class of linear models
- Evaluation: find the direction **w** that maximizes J(**w**)= $\frac{(m_2-m_1)^2}{s_1^2+s_2^2}$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

- Optimization

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Hyper parameters / user defined parameters

AVOIDING OVERFITTING AND DATA MEMORIZATION

Regularization

- To build a machine learning algorithm we specify model family, a cost function and optimization procedure
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error
  - There are many regularization strategies
- Regularization works by trading increased bias for reduced variance. An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.