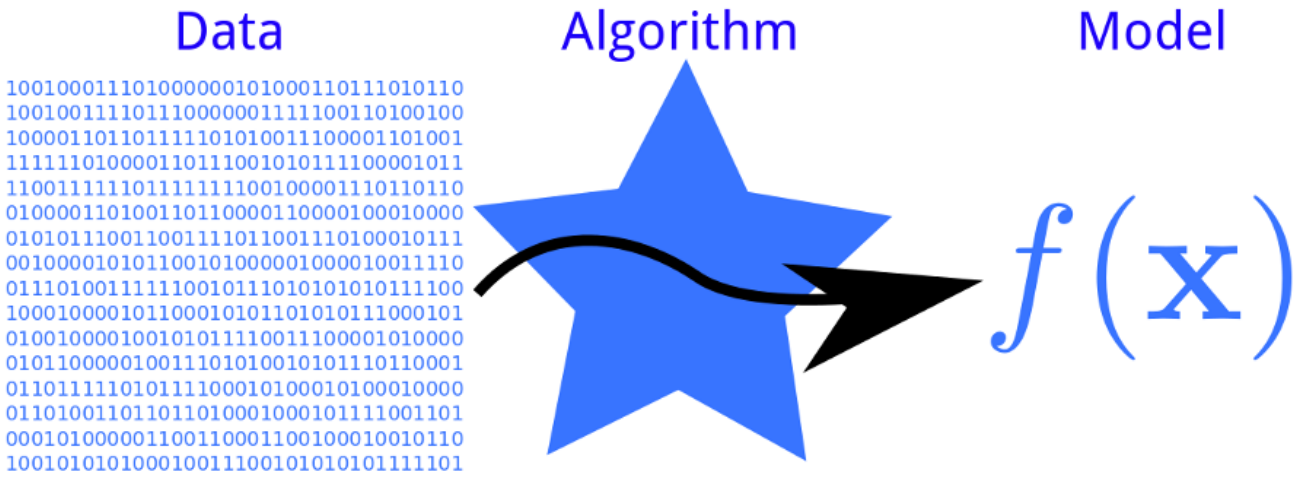
Introduction to Machine Learning



Lecture 1: Introduction

Iasonas Kokkinos

Iasonas.kokkinos@gmail.com

University College London

Course Requirements and Grading

Lab exercises (30%)

- **Python:** easy to get started
- Synthetic data (easy and fast to train)
- Hands-on experience with algorithms
- 3 deliverables, distributed evenly over the module
- 20-30% during practicals, 70-80% at home

Theory exercises (0/20)

- Close to the end (early December), getting you started for the exam.

Final exam (70%)

- Theory questions (judgement-oriented)
- Simulate running algorithms by hand

Deliverables and deadlines

Exact dates of deadlines: announced at least two weeks in advance

Strict policy (hard-wired in moodle)

- If moodle fails: deadline will be extended
- If your DSL connection fails: this is not our problem
- Take into account that uploading may take 1-2 minutes. Your assignment **must have finished uploading** before the exact deadline

Meeting hours

Office: 110B, 68-72 Gower street

Meetings hours: Tuesday, 11:00-12:00 am

Please, use this time

If you need more time for meetings, just ask!

Course support

- Moodle
  - Pointers to external references
  - Assignments
  - Discussion groups
  - Please, do **not** ask the Tas/myself in the discussion groups– send an email
- Questions regarding the assignments: email the TAs (feel free to CC me)
- Questions regarding the course:
  - come to the meeting hours
  - email if the answer can be short

Feedback

Please do it

- -during the lecture
- -in person
- -through the TAs (maybe?)
- -at the end (required to improve the course for next year)

Please, do not wait for the end

Last year: we did extra courses because of feedback

Still came late

Prerequisites

One-semester course in **all of these topics** **(not** any of them)

- Linear Algebra
- Calculus
- Probability
- Programming

Course objective: understand in detail core concepts

First week: make sure that you do have the background for the course

Lecture outline

Introduction to the course

Introduction to Machine Learning

Least squares

Machine Learning

Principles, methods, and algorithms for learning and prediction based on past evidence

Goal: Machines that perform a task based on experience, instead of explicitly coded instructions

Why?

- Crucial component of every intelligent/autonomous system
- Important for a system's adaptability
- Important for a system's generalization capabilities
- Attempt to understand human learning

Machine Learning variants

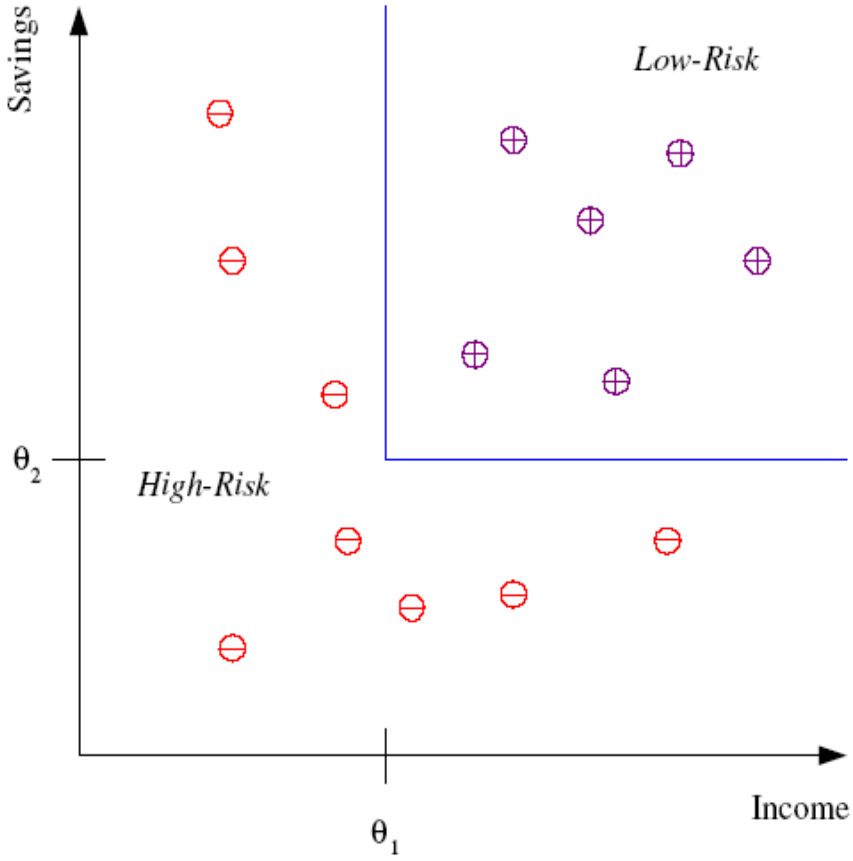
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised/semi-supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: sparse reward for a sequence of decisions

Classification

- Based on our experience, should we give a loan to this customer?
  - Binary decision: yes/no



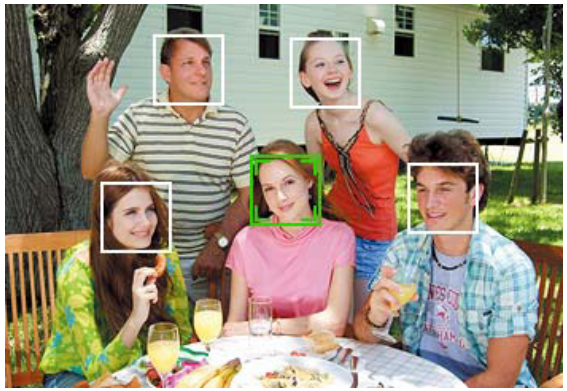
Decision boundary

Classification examples

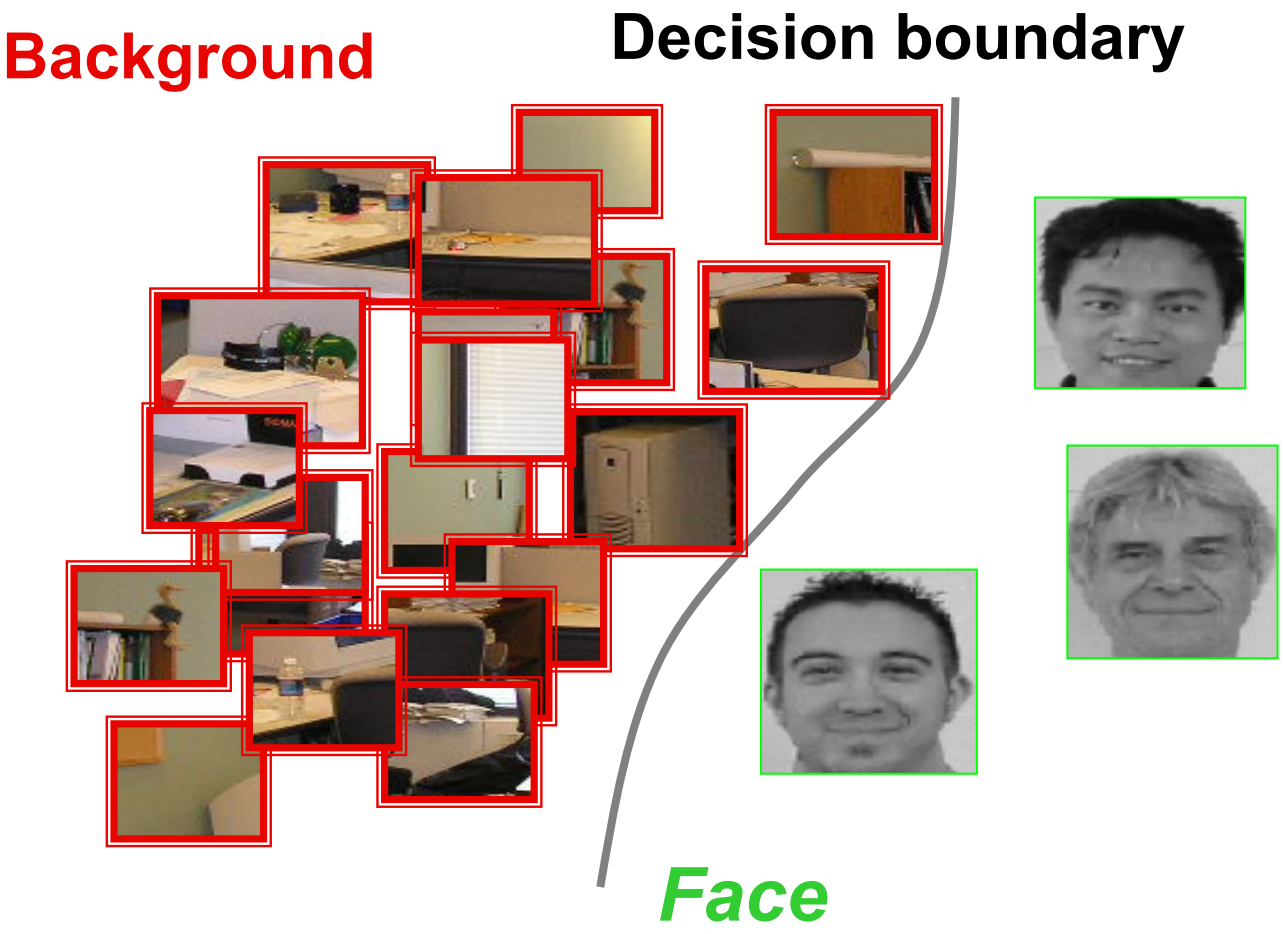
- Digit Recognition

- Spam Detection

- Face detection



`Faceness function': classifier



Test time: deploy the learned function

- Scan window over image
  - Multiple scales
  - Multiple orientations
- Classify window as either:
  - Face
  - Non-face

Machine Learning variants

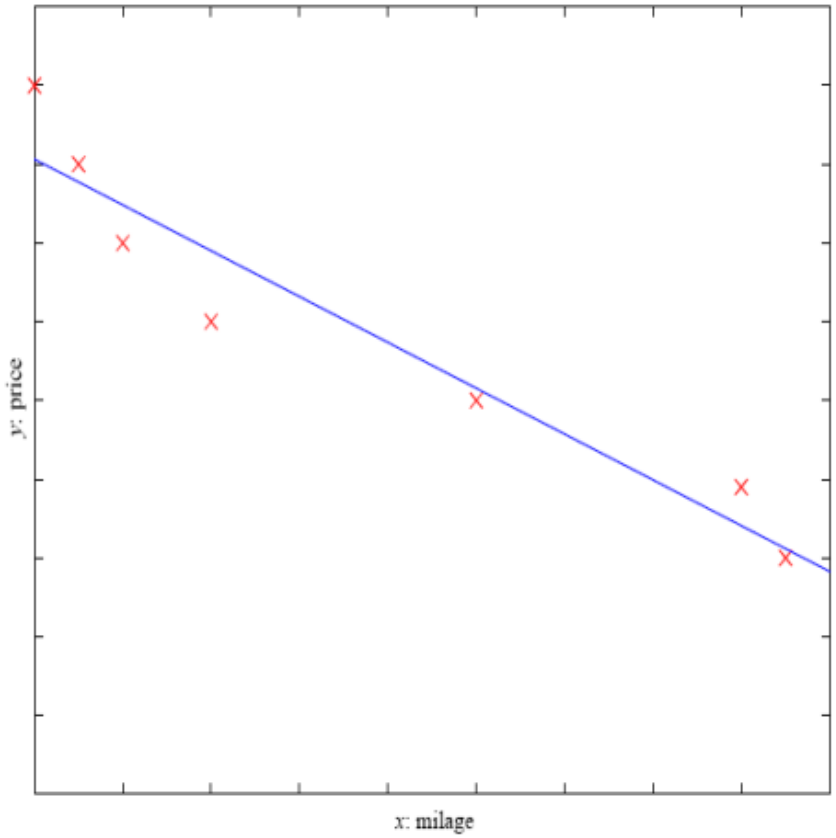
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

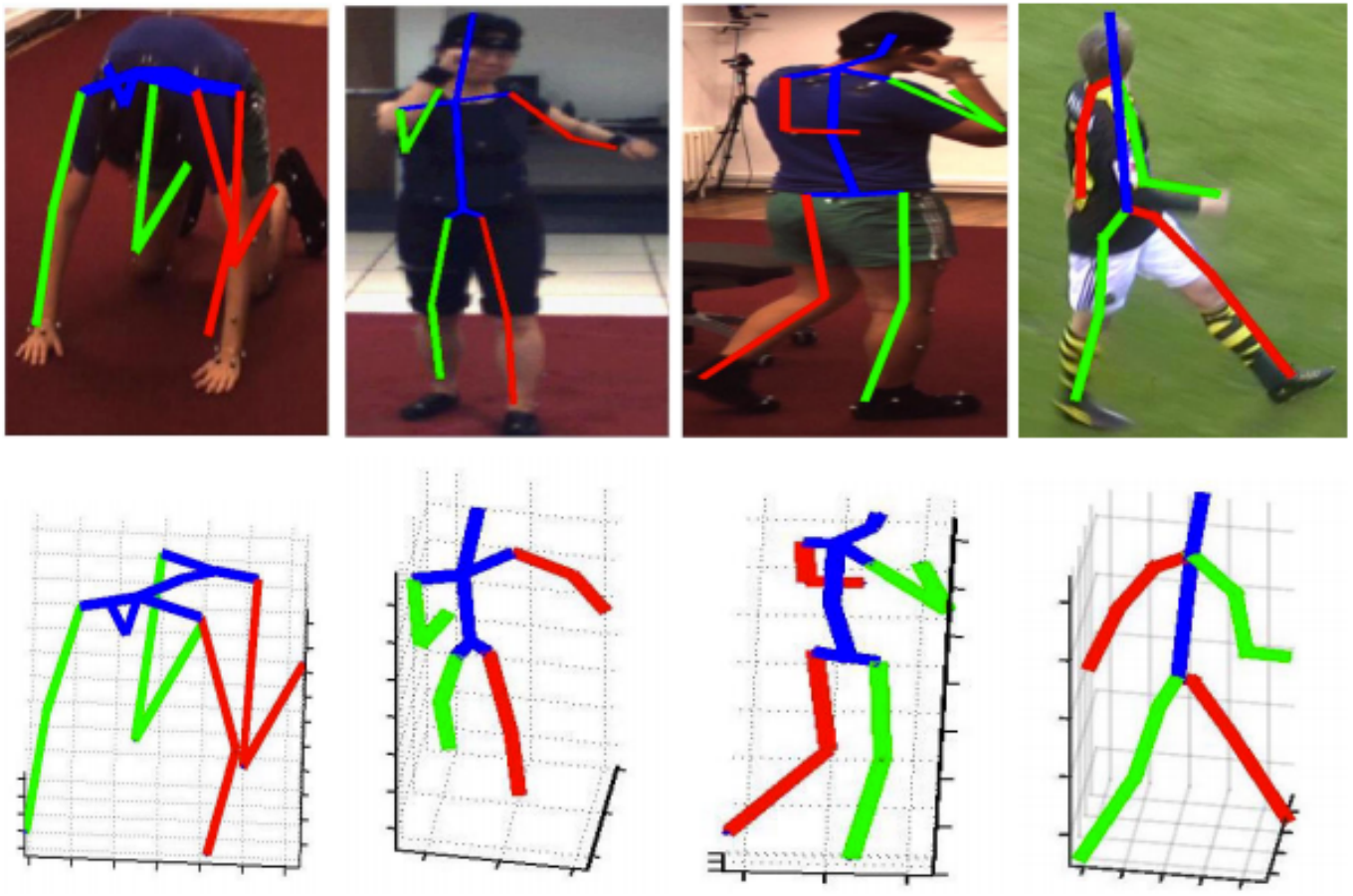
Regression

- Output: Continuous
  - E.g. price of a car based on years, mileage, condition,…



Computer vision example

- Human estimation: from image to vector-valued pose estimate



Machine Learning variants

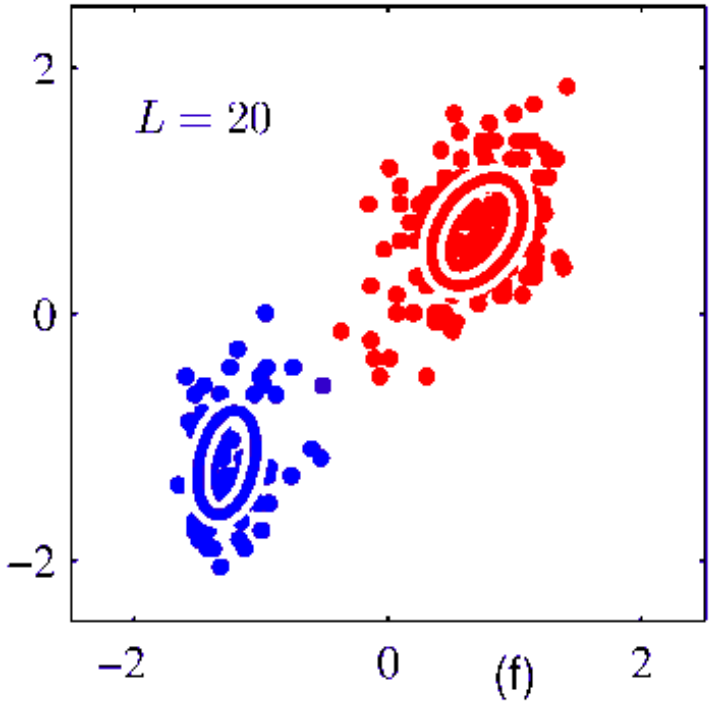
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

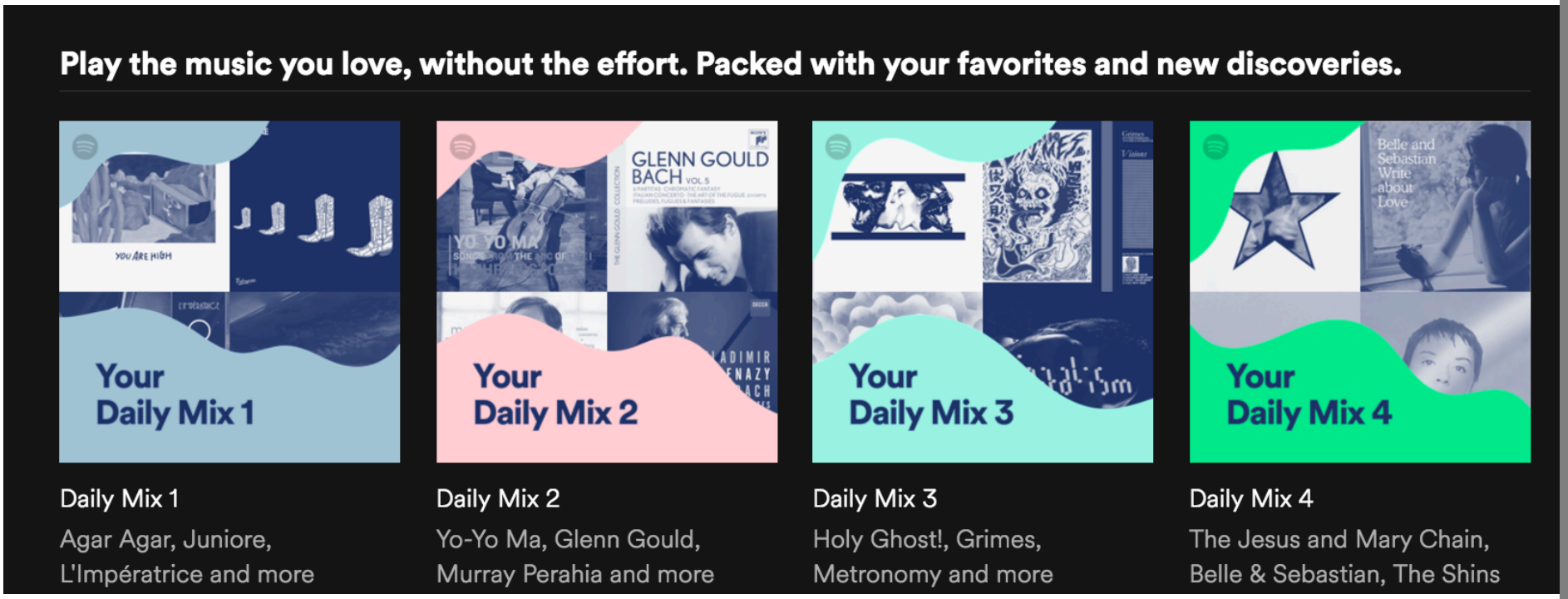
Clustering

- Break a set of data into coherent groups
  - Labels are `invented'



Clustering examples

- Spotify recommendations



Clustering examples

- Image segmentation

Machine Learning variants

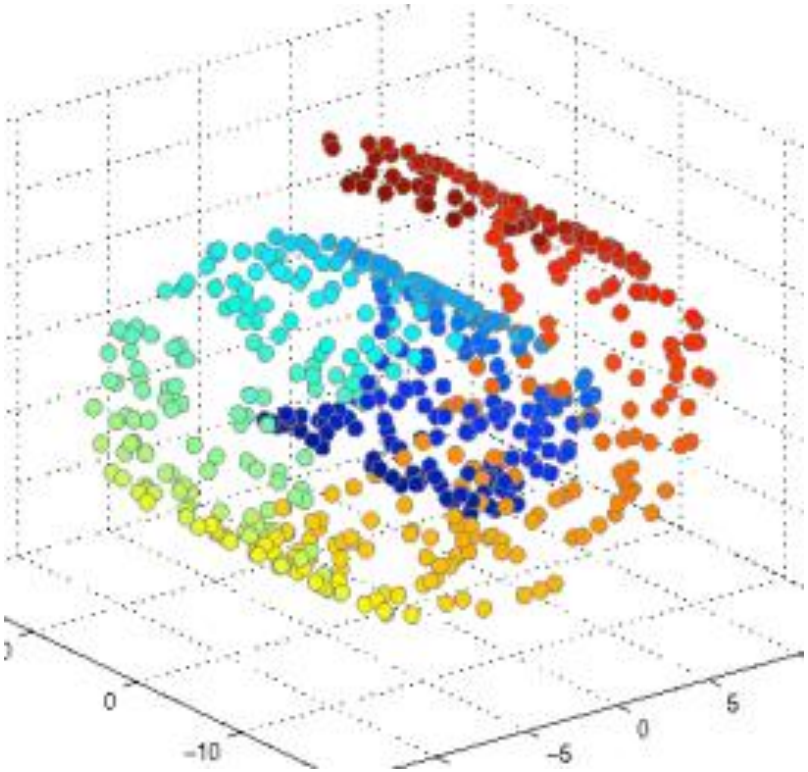
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

Dimensionality reduction & manifold learning

- Find a low-dimensional representation of high-dimensional data
  - Continuous outputs are `invented'



Example of nonlinear manifold: faces

Average of two faces is not a face

$\mathbf{x}_1$

$\frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2)$

$\mathbf{x}_2$

Moving along the learned face manifold

Trajectory along the “male” dimension

Trajectory along the “young” dimension

Lample et. al. Fader Networks, NIPS 2017

Machine Learning variants

- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised/semi supervised

  Partially supervised
- Reinforcement learning

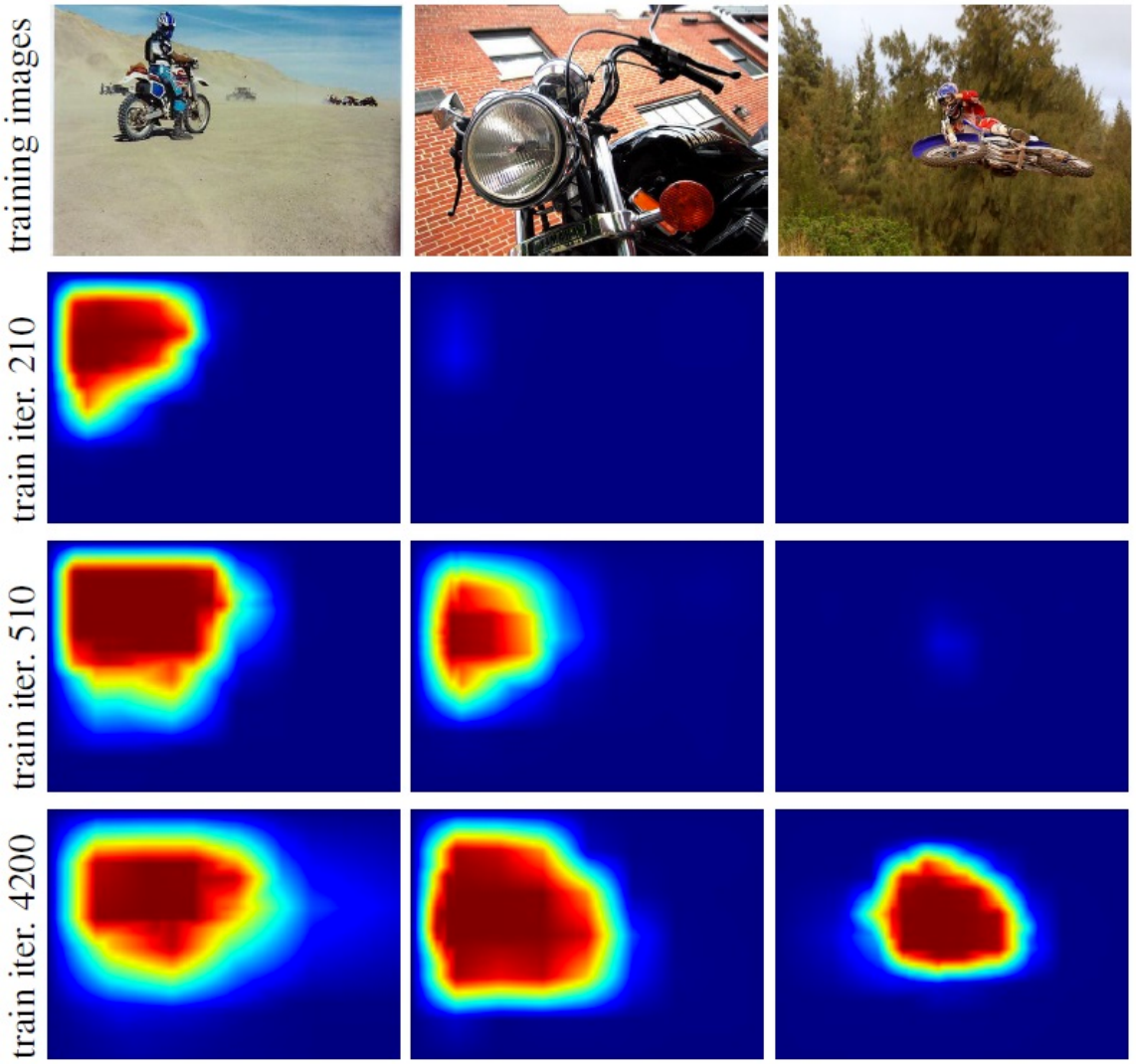
  Supervision: reward for a sequence of decisions

Weakly supervised learning: only part of the supervision signal

training images

Supervision signal: “motorcycle”

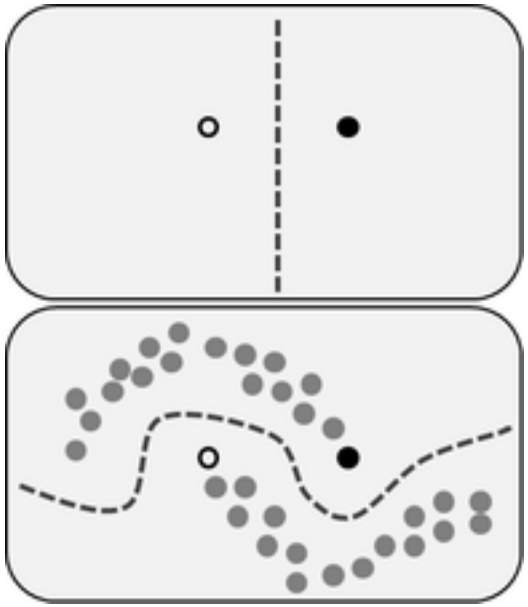
Weakly supervised learning: only part of the supervision signal



Supervision signal: “motorcycle”

Inferred localization information

Semi-supervised learning: only part of the data labelled



Labelled data

Labelled + unlabelled data

Machine Learning variants

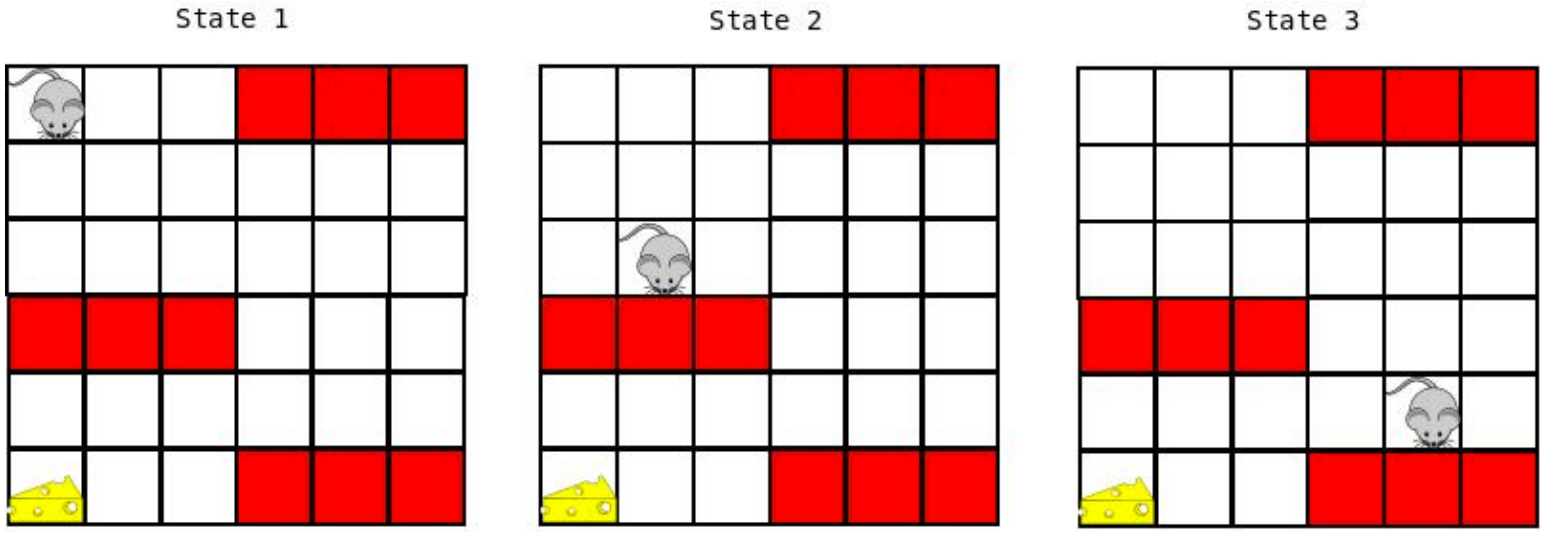
- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction
- Weakly supervised/semi supervised learning

  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

Reinforcement learning

- Agent interacts with environment repeatedly
  - Take actions, based on state
  - (occasionally) receive rewards
  - Update state
  - Repeat
- Goal: maximize cumulative reward



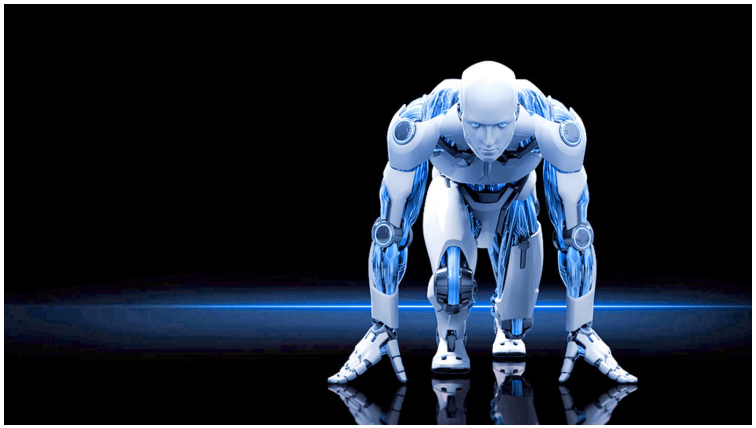
Reinforcement learning examples

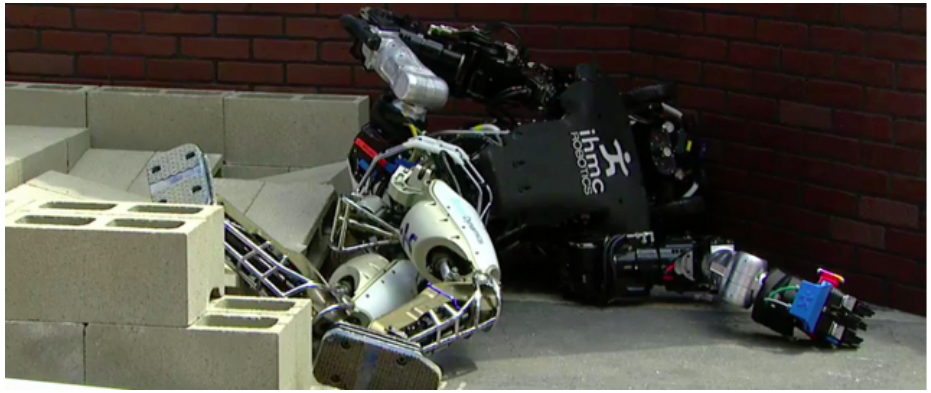
- Beat human champions in games

Backgammon, 90's

GO, 2015

- Robotics





Tentative Course Schedule

- 1st week: Introduction & linear regression.
- 2nd week: Regularization, Ridge Regression, Cross-Validation,
- 3rd week: Logistic Regression
- 4th week: Support Vector Machines
- 5th week: Ensemble Models (Adaboost, Random Forests)
- 6th week: Deep Learning (neural networks, backpropagation, SGD)
- 7th week: Deep Learning and applications
- 8th week: Unsupervised learning (K-means, PCA, Sparse Coding)
- 9th week: Probabilistic modelling (hidden variable models, EM)
- 10th week: Introduction to Reinforcement Learning

No rush - stop me whenever something is not clear!

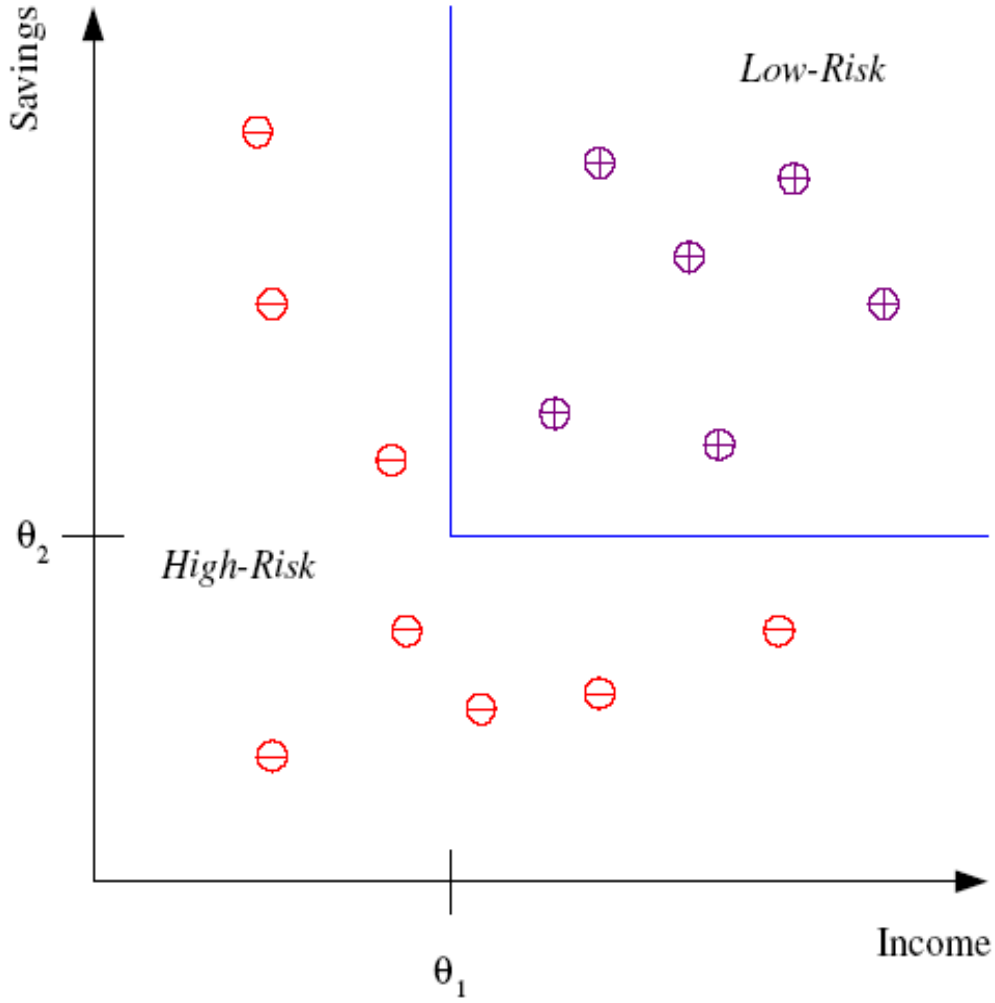
Focus of first part: supervised learning

- Supervised
  - Classification
  - Regression
- Unsupervised
  - Clustering
  - Dimensionality Reduction, Manifold Learning
- Weakly supervised

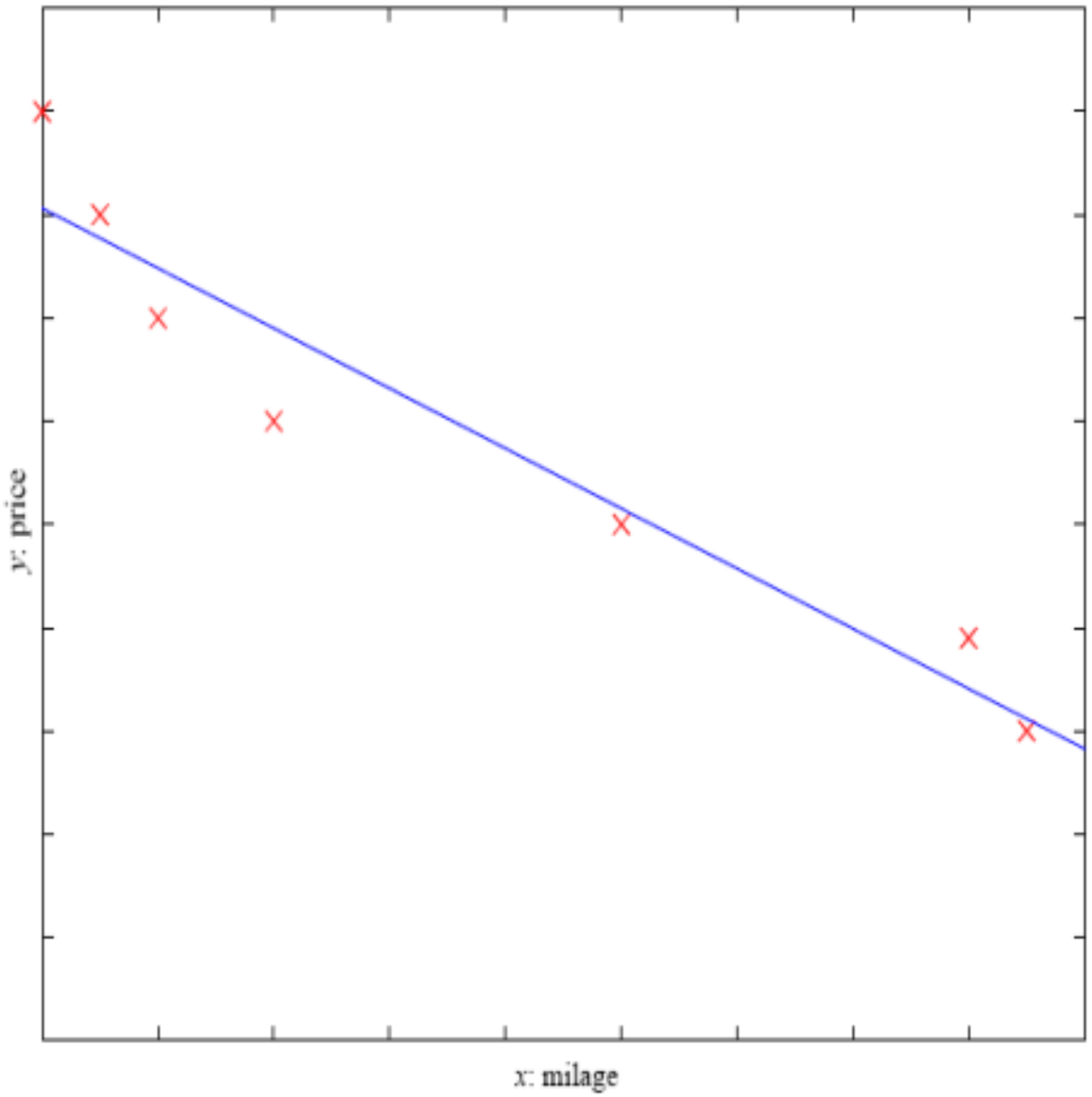
  Some data supervised, some unsupervised
- Reinforcement learning

  Supervision: reward for a sequence of decisions

Classification: yes/no decision



Regression: continuous output



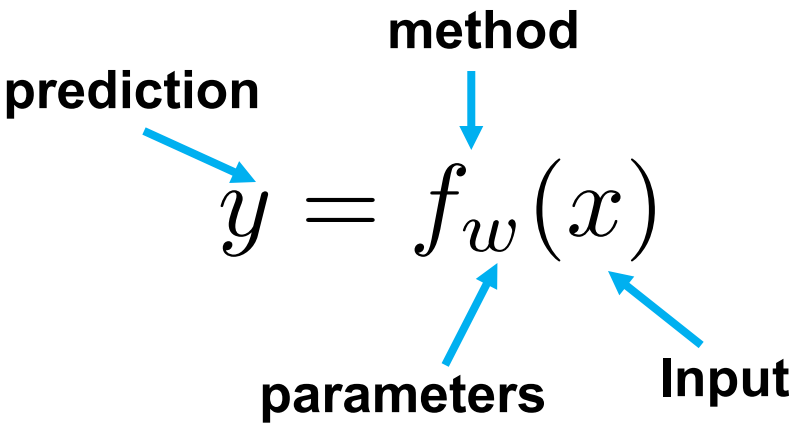
What we want to learn: a function

- Input-output mapping

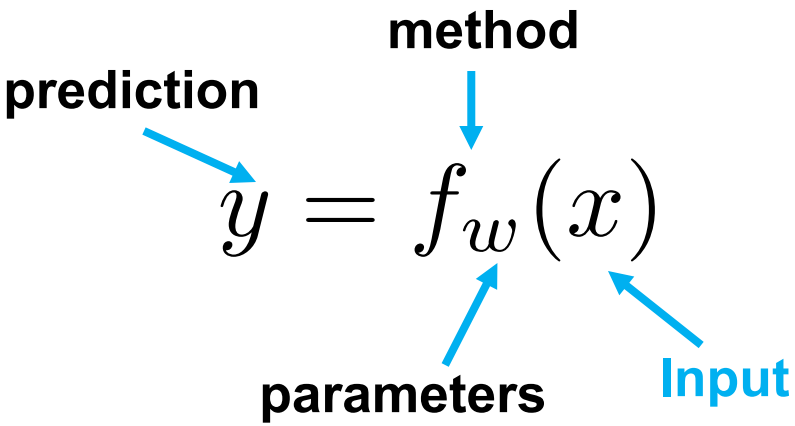
$$y = f_w(x)$$

What we want to learn: a function

- Input-output mapping



What we want to learn: a function

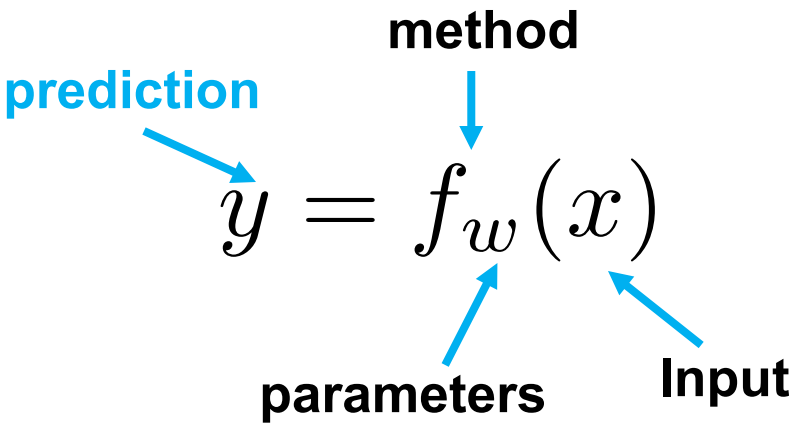


Calculus $x \in \mathbb{R}$

Vector calculus $\mathbf{x} \in \mathbb{R}^D$

Machine learning: can work also for discrete inputs, strings, trees, graphs,…

What we want to learn: a function



Classification: $y \in \{0, 1\}$

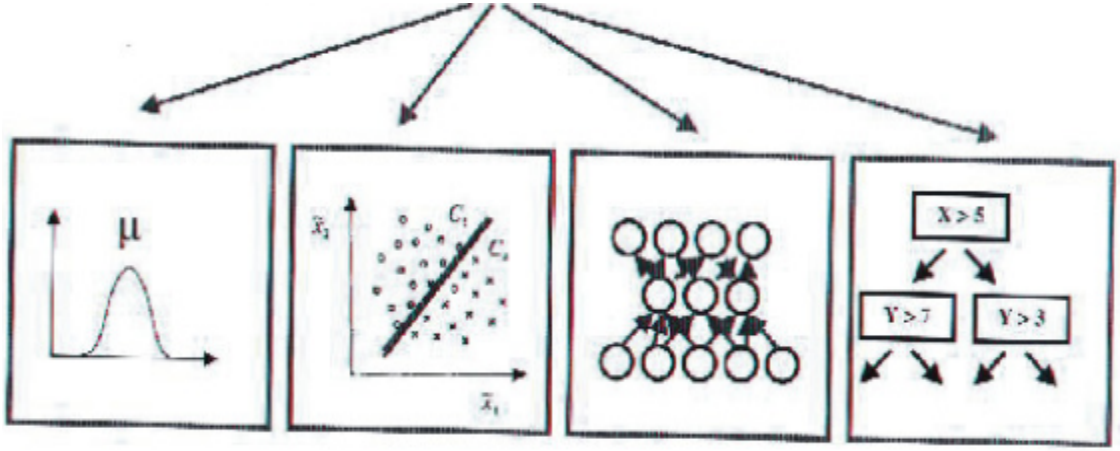
Regression: $y \in \mathbb{R}$

What we want to learn: a function

prediction

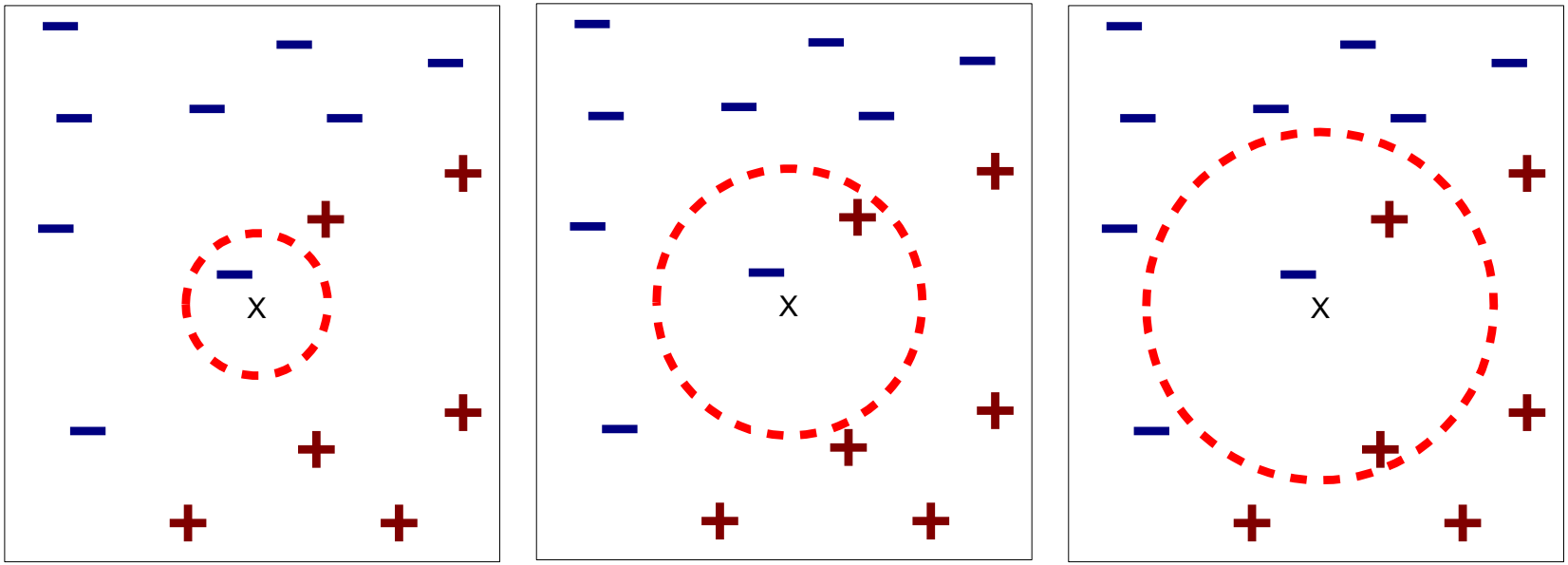
method

$$y = f_w(x)$$



Linear classifiers, neural networks, decision trees, ensemble models, probabilistic classifiers, …

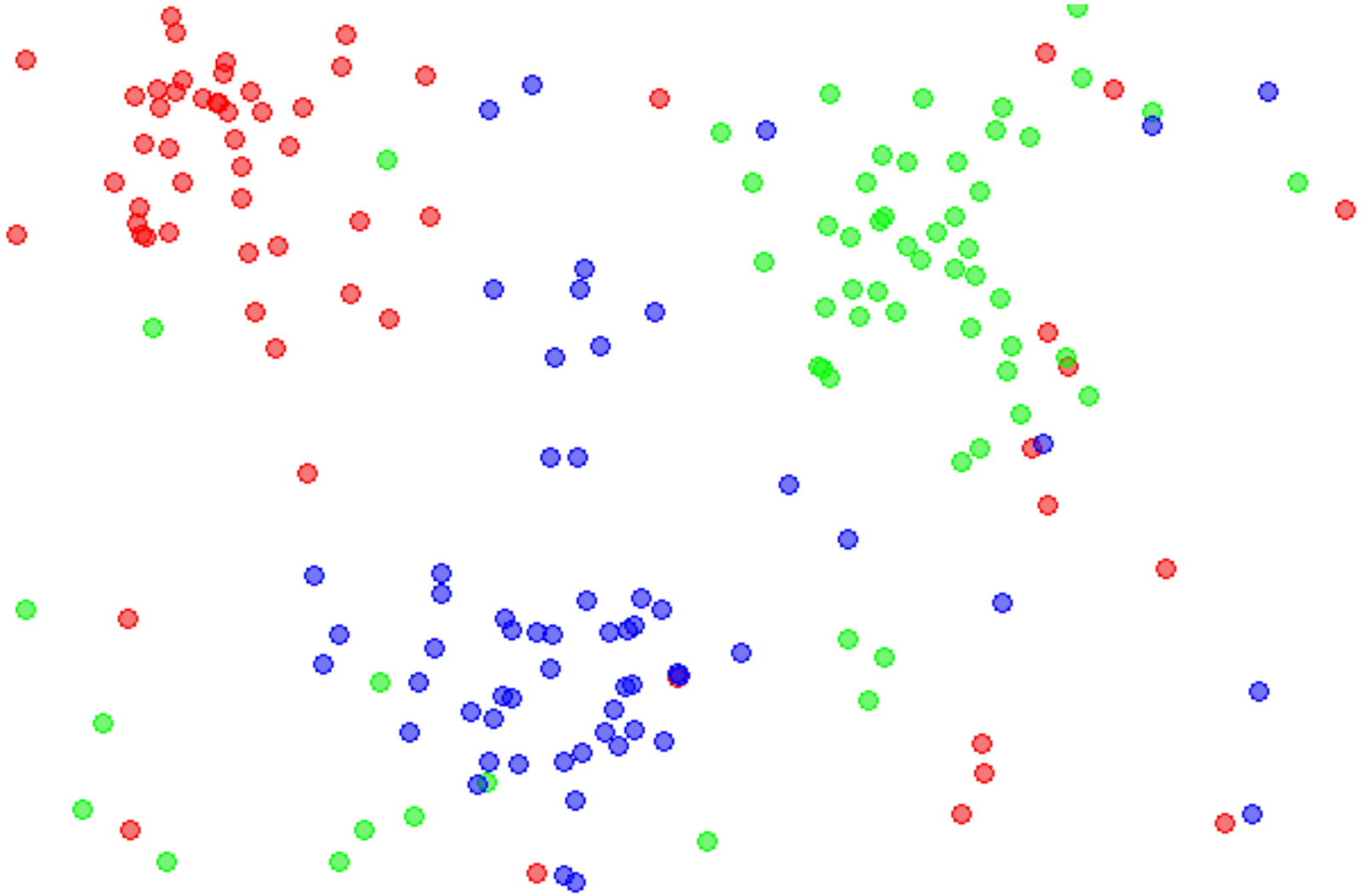
Example of method: K-nearest neighbor classifier



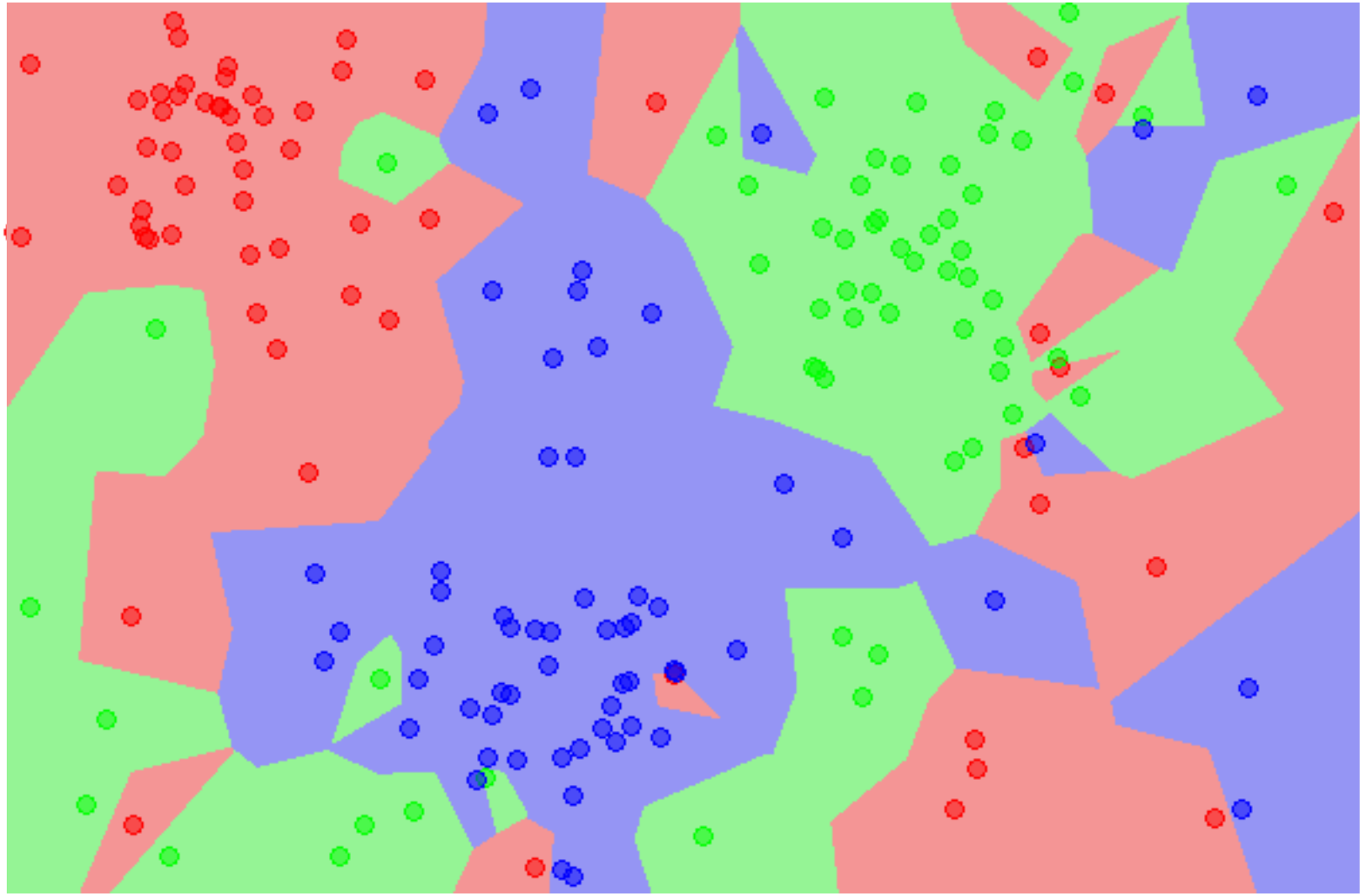
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- **Compute distance to other training records**
- **Identify *K* nearest neighbors**
- **Take majority vote**

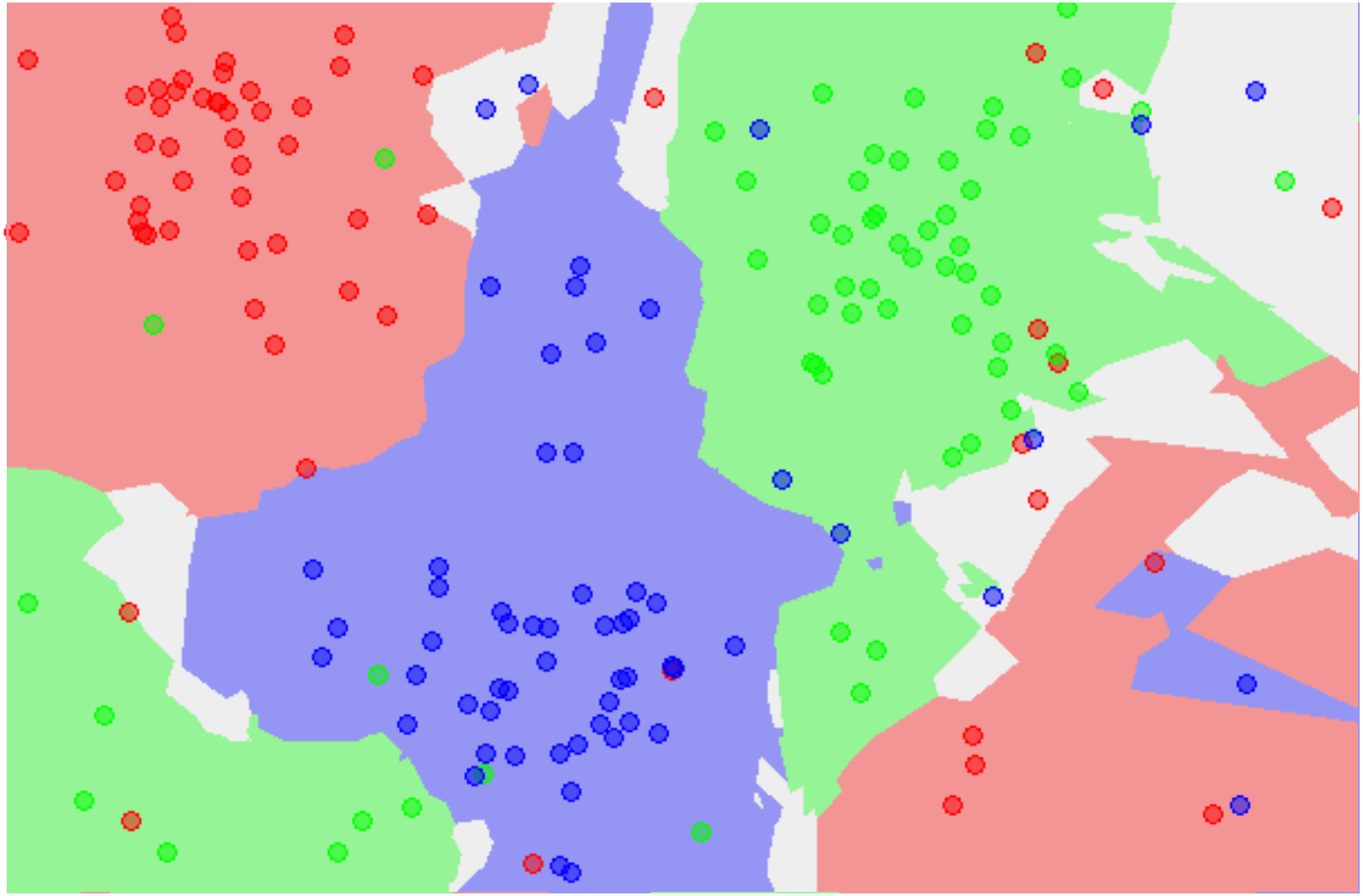
Training data for NN classifier (in R^2)



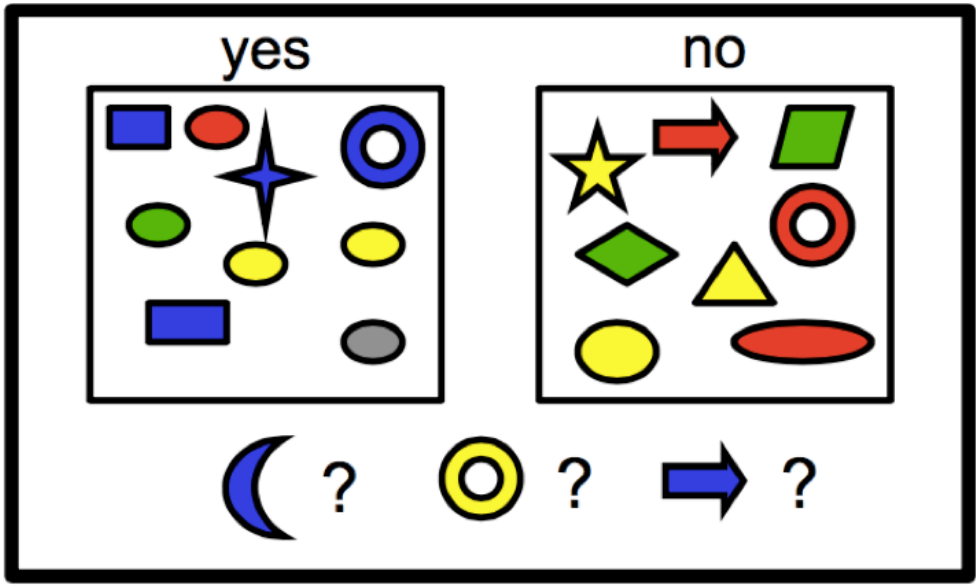
1-nn classifier prediction (in R^2)



3-nn classifier prediction



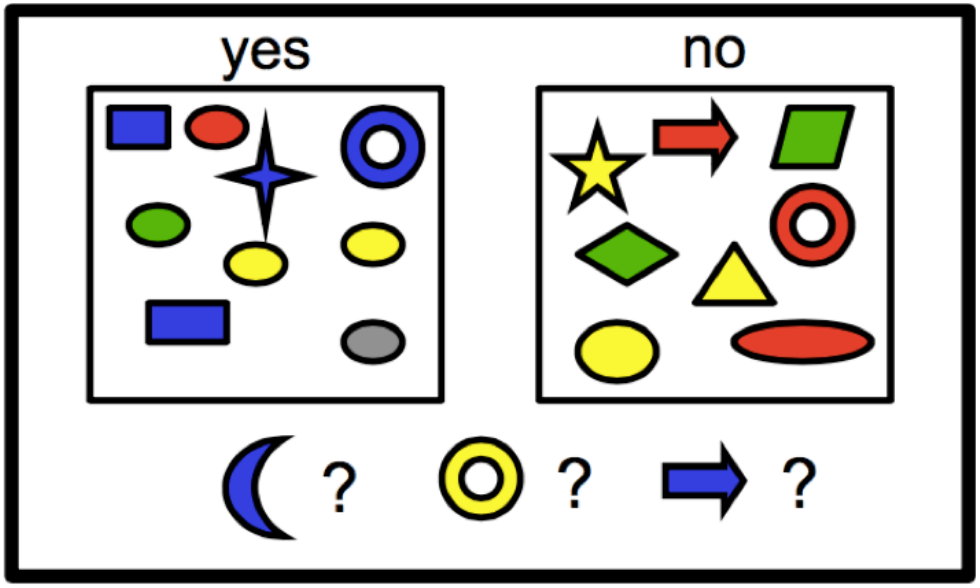
Method example: decision tree



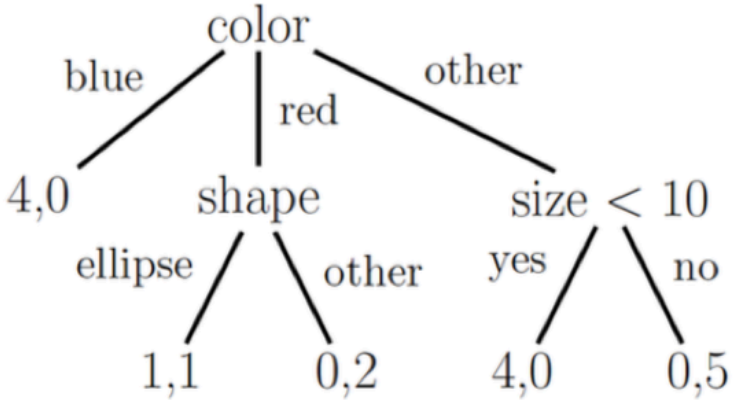
Features: color, shape, size

Machine learning: can work also for discrete inputs, strings, trees, graphs,…

Method example: decision tree

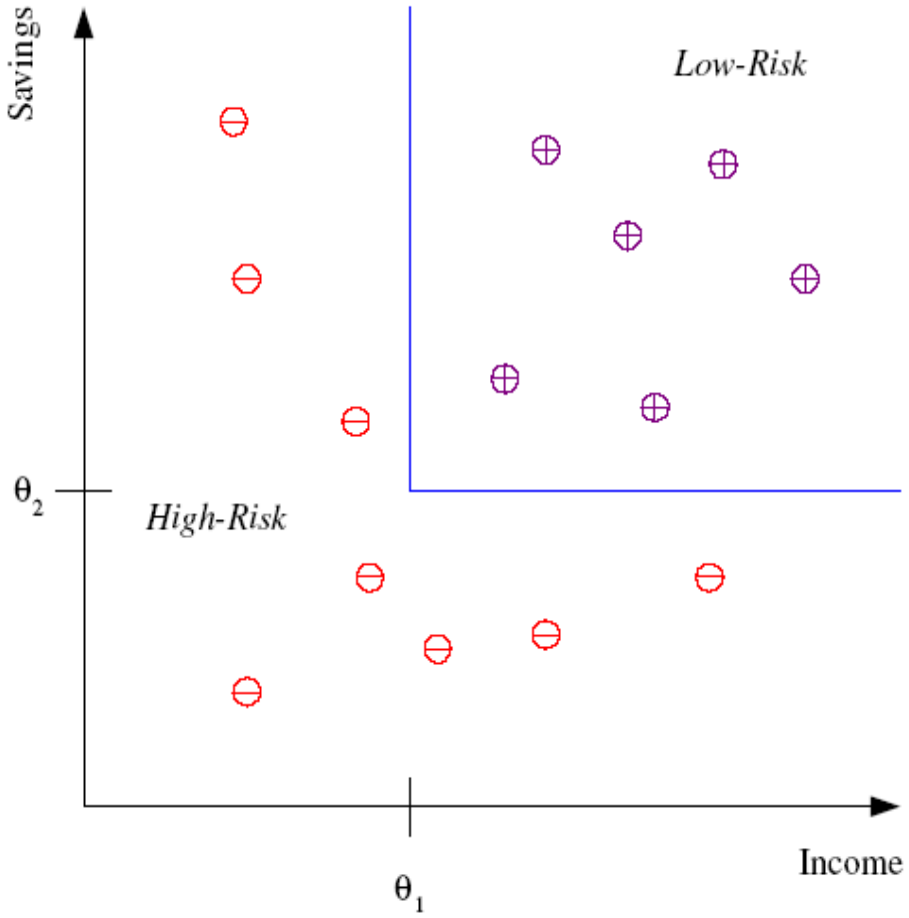


Features: color, shape, size

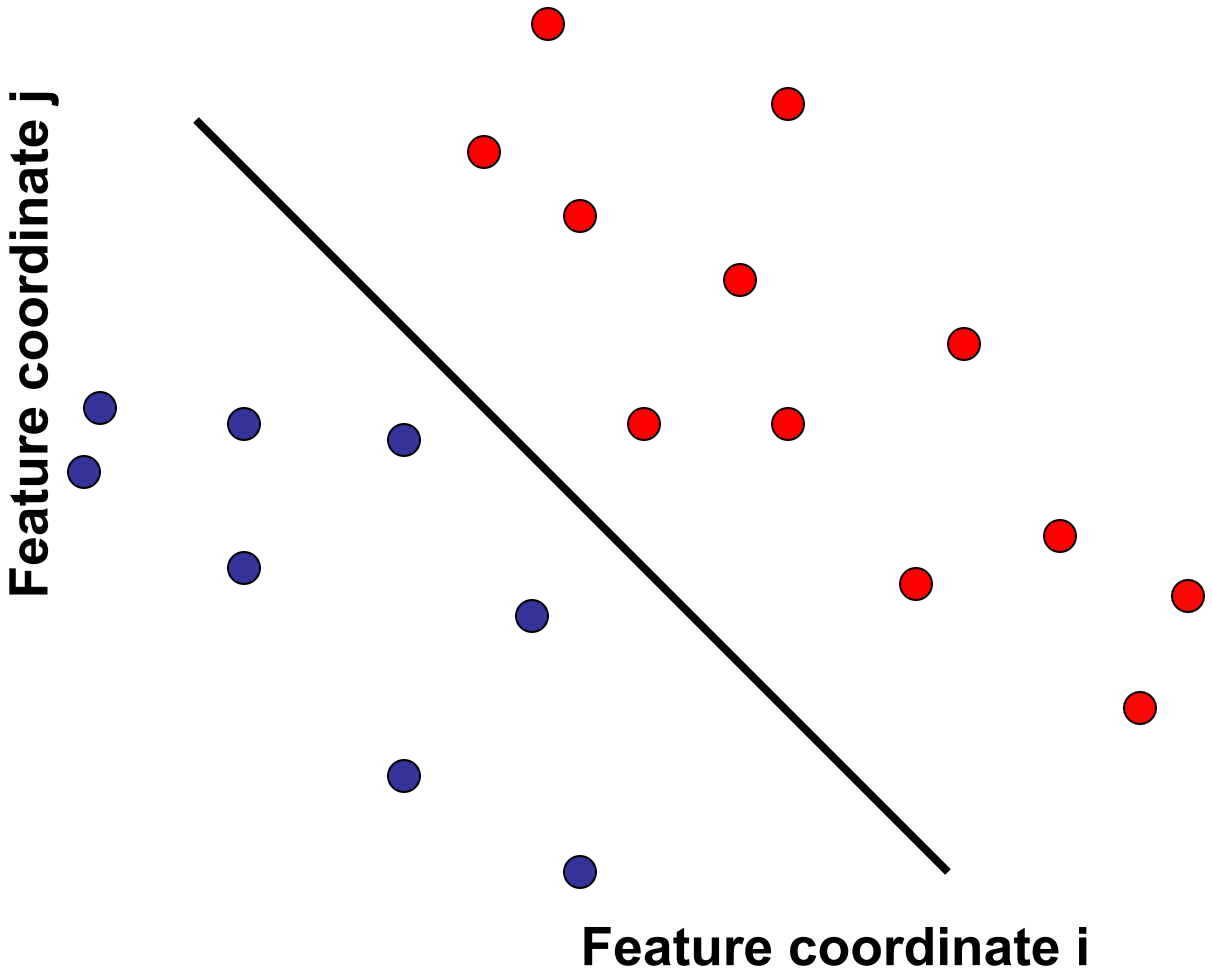


Method example: decision tree

What is the depth of the decision tree for this problem?

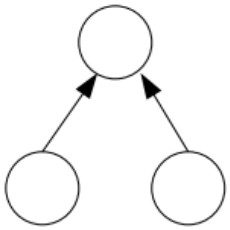


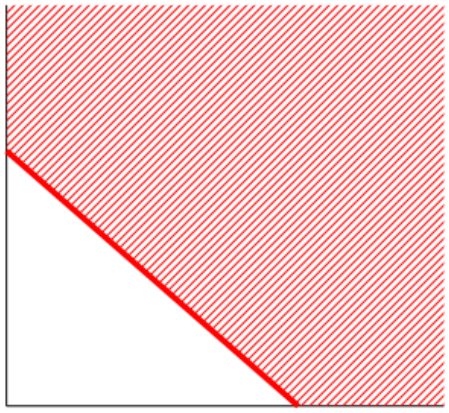
Method example: linear classifier



Method example: neural network

1 layer of trainable weights

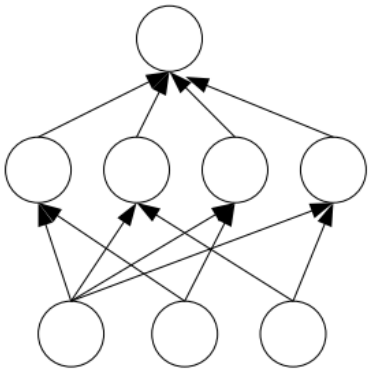


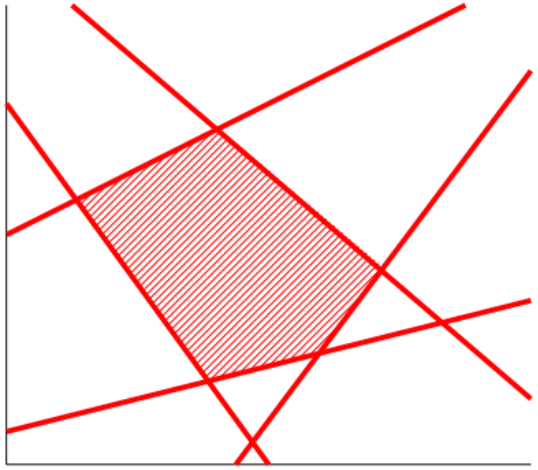


separating hyperplane

Method example: neural network

2 layers of trainable weights

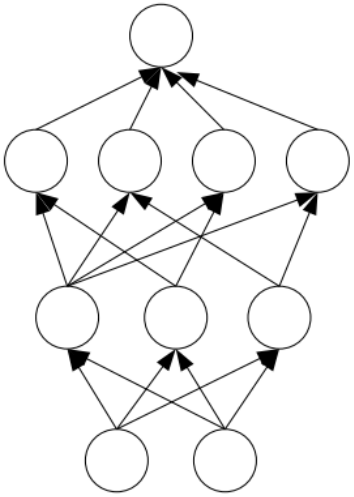


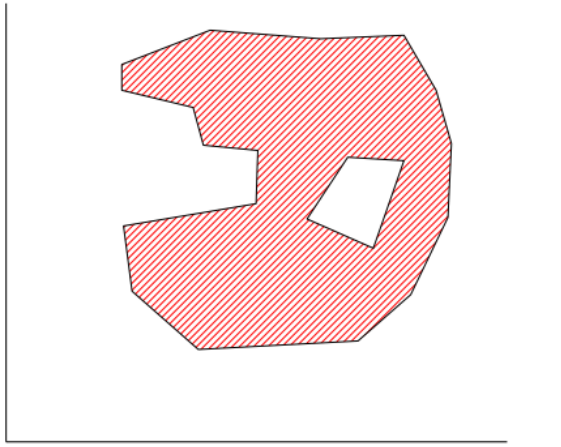


convex polygon region

Method example: neural network

3 layers of trainable weights





composition of polygons: convex regions

Lectures 1-7: all of that

- 1st week: Introduction & linear regression.
- 2nd week: Regularization, Ridge Regression, Cross-Validation,
- 3rd week: Logistic Regression
- 4th week: Support Vector Machines
- 5th week: Ensemble Models (Adaboost, Random Forests)
- 6th week: Deep Learning (neural networks, backpropagation, SGD)
- 7th week: Deep Learning and applications
- 8th week: Unsupervised learning (K-means, PCA, Sparse Coding)
- 9th week: Probabilistic modelling (hidden variable models, EM)
- 10th week: Introduction to Reinforcement Learning

No rush - stop me whenever something is not clear!

Lectures 1-7: all of that

- 1st week: Introduction & linear regression.
- 2nd week: Regularization, Ridge Regression, Cross-Validation,
- 3rd week: Logistic Regression
- 4th week: Support Vector Machines
- 5th week: Ensemble Models (Adaboost, Random Forests)
- 6th week: Deep Learning (neural networks, backpropagation, SGD)
- 7th week: Deep Learning and applications
- 8th week: Unsupervised learning (K-means, PCA, Sparse Coding)
- 9th week: Probabilistic modelling (hidden variable models, EM)
- 10th week: Introduction to Reinforcement Learning

No rush - stop me whenever something is not clear!

We have two centuries of material to cover!

https://en.wikipedia.org/wiki/Least_squares

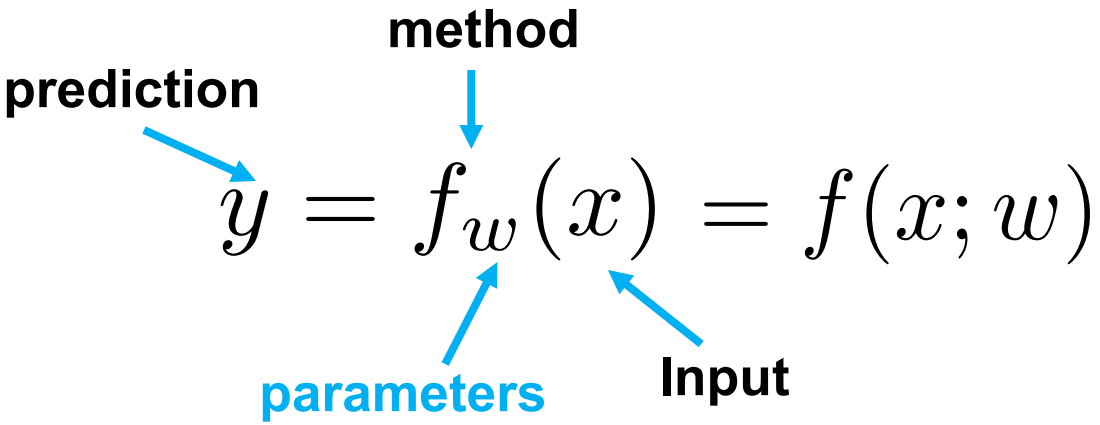
The first clear and concise exposition of the method of least squares was published by Legendre in **1805.**

The technique is described as an **algebraic procedure** for **fitting linear equations to data** and Legendre demonstrates the new method by analyzing the same data as Laplace for the shape of the earth.

The value of Legendre's method of least squares was immediately recognized by leading astronomers and geodesists of the time

What we want to learn: a function

- Input-output mapping



$$w \in \mathbb{R}$$

$$\mathbf{w} \in \mathbb{R}^K$$

Assumption: linear function

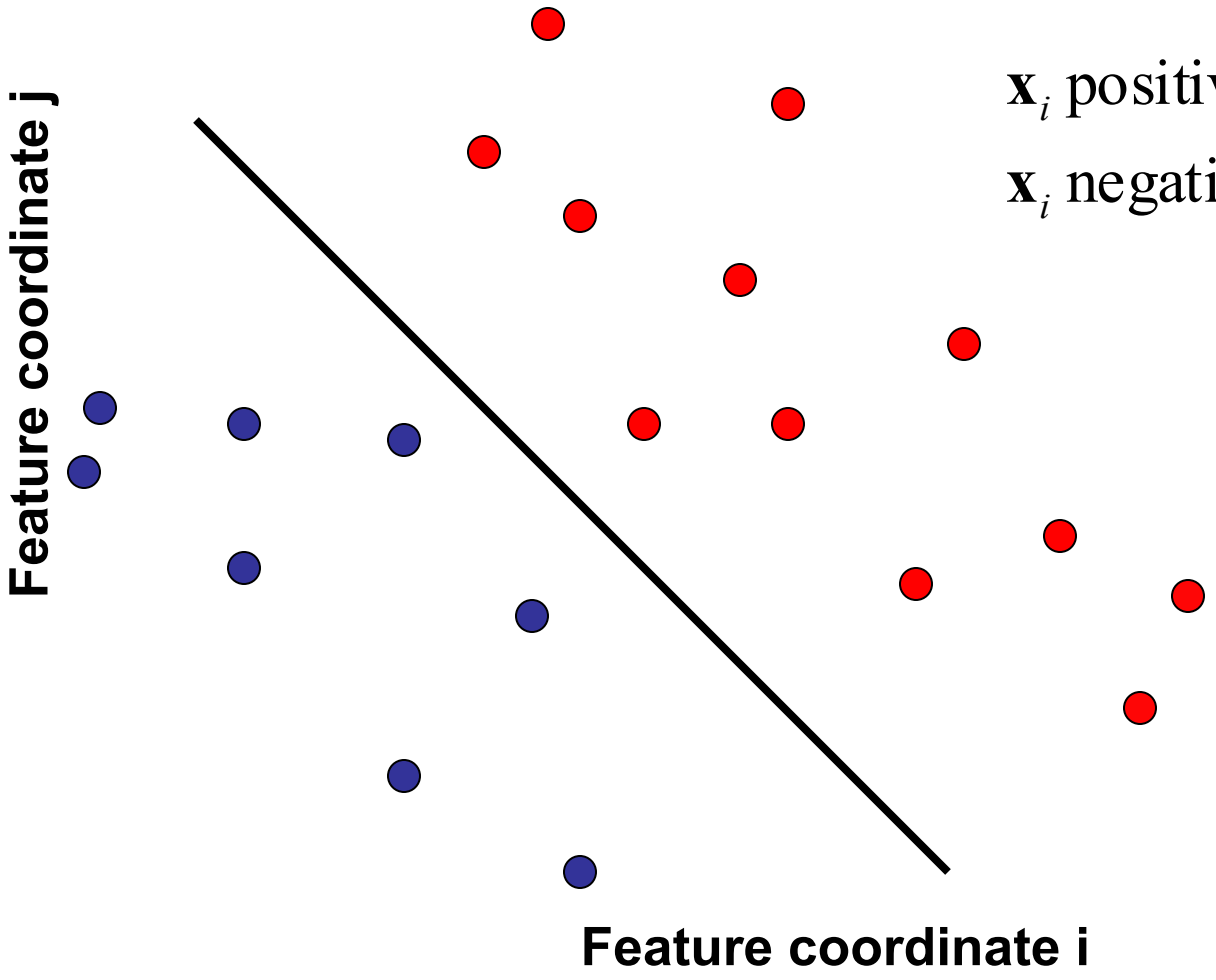
$$y = f_{\mathbf{w}}(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

Inner product:

$$\mathbf{w}^T \mathbf{x} = \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{d=1}^{D} \mathbf{w}_d \mathbf{x}_d$$

$$\mathbf{x} \in \mathbb{R}^D, \mathbf{w} \in \mathbb{R}^D$$

Reminder: linear classifier



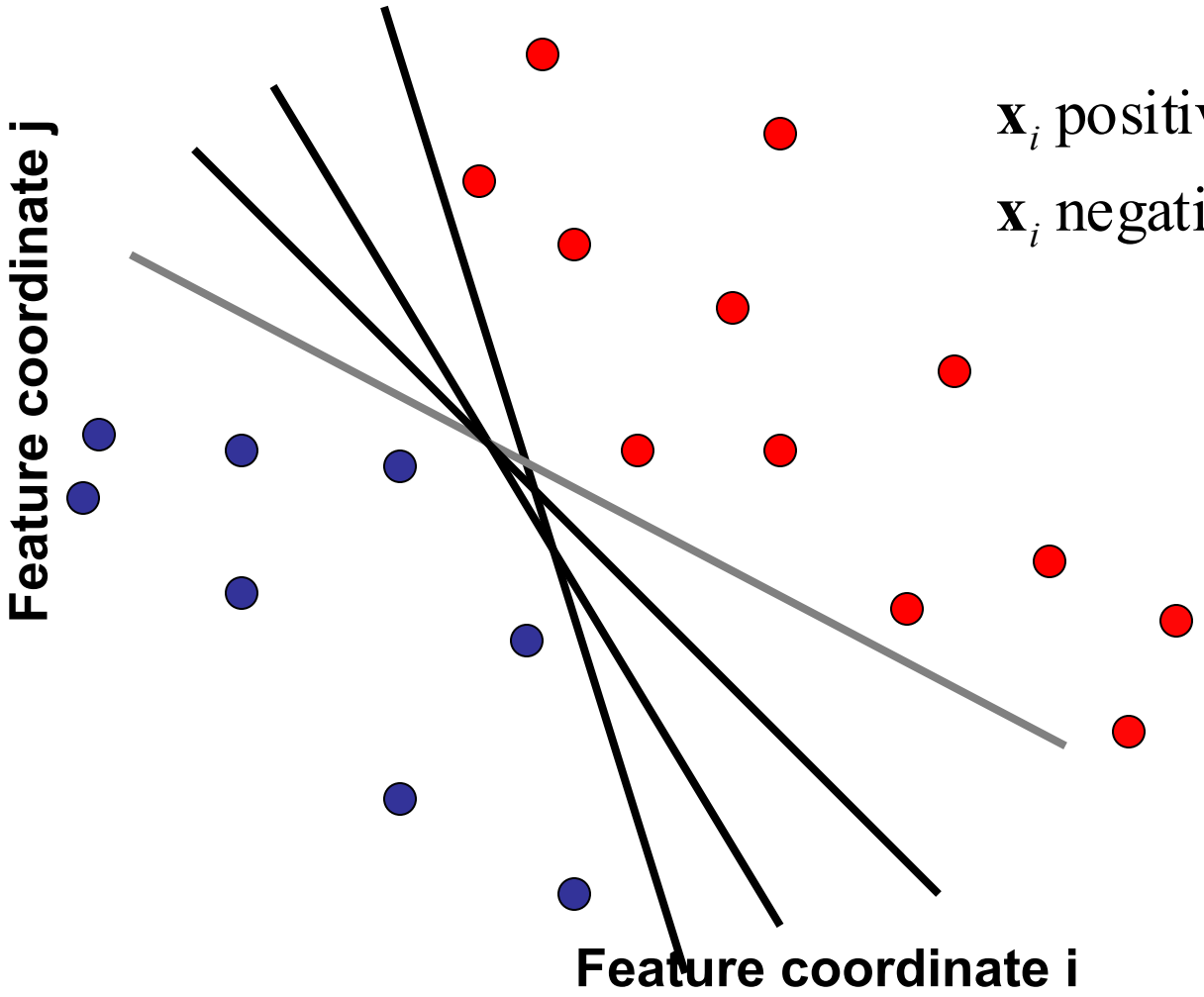
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$$

Question: which one?



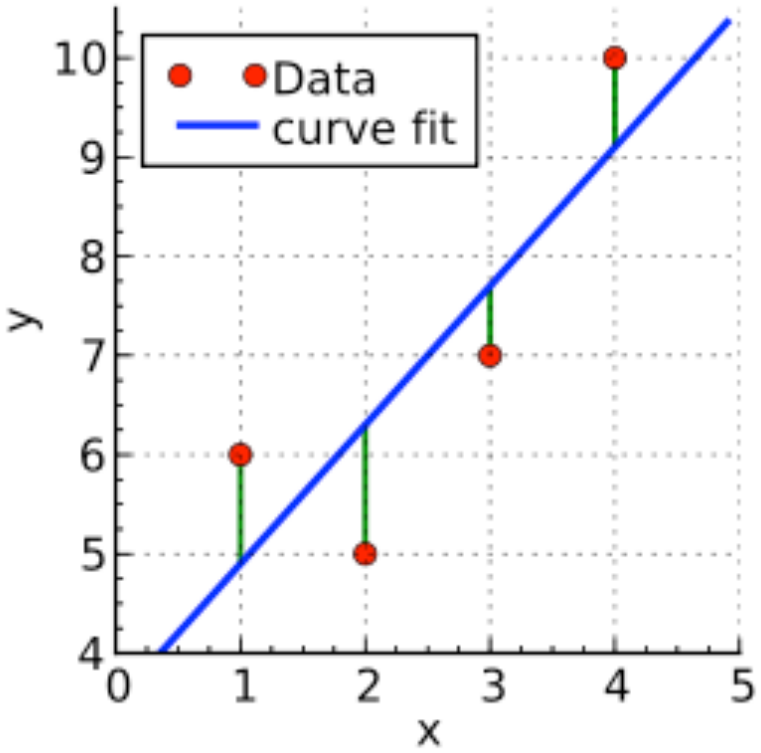
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

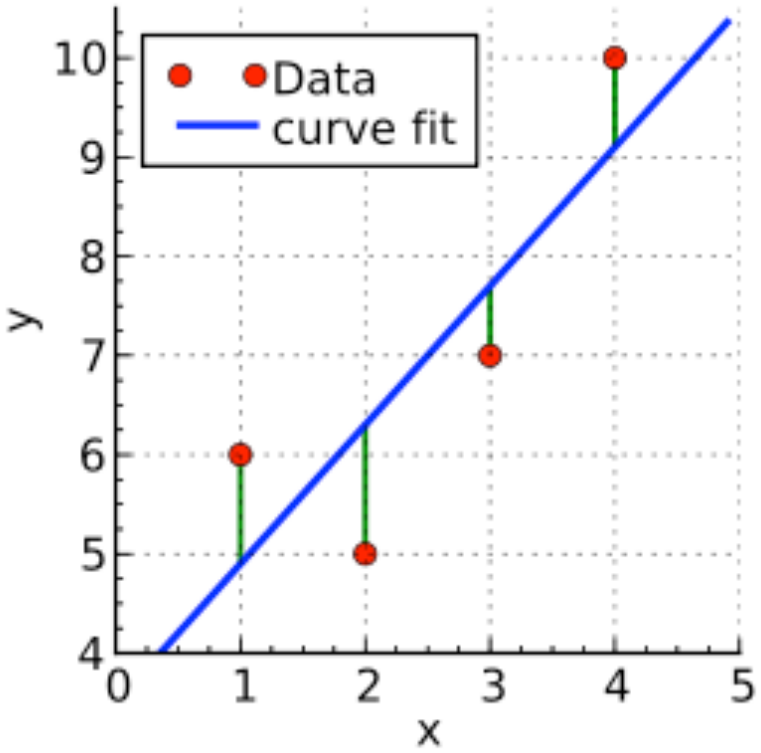
Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$$

Linear regression in 1D



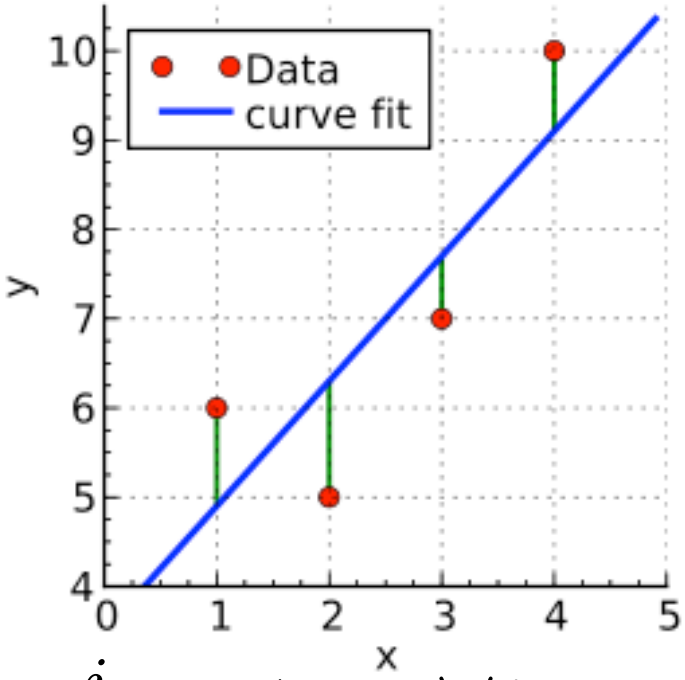
Linear regression in 1D



Training set: input–output pairs $\mathcal{S} = \{(x^i, y^i)\}, \quad i = 1 \ldots, N$

$$x^i \in \mathbb{R}, \quad y^i \in \mathbb{R}$$

Linear regression in 1D



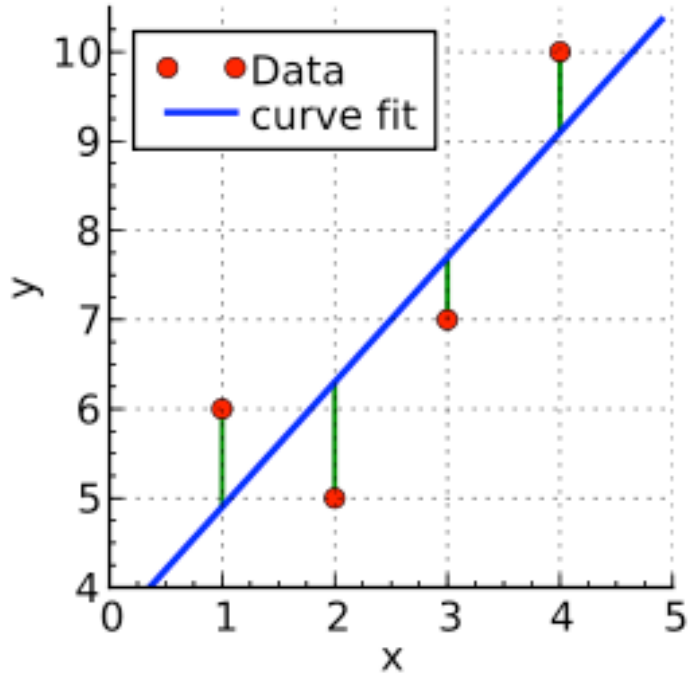
$$y^i = w_0 + w_1 x_1^i + \epsilon^i$$

$$= w_0 x_0^i + w_1 x_1^i + \epsilon^i, \quad x_0^i = 1, \quad \forall i$$

$$= \mathbf{w}^T \mathbf{x}^i + \epsilon^i$$

Sum of squared errors criterion

$$y^i = \mathbf{w}^T \mathbf{x}^i + \epsilon^i$$



Loss function: sum of squared errors

$$L(\mathbf{w}) = \sum_{i=1}^{N} (\epsilon^i)^2$$

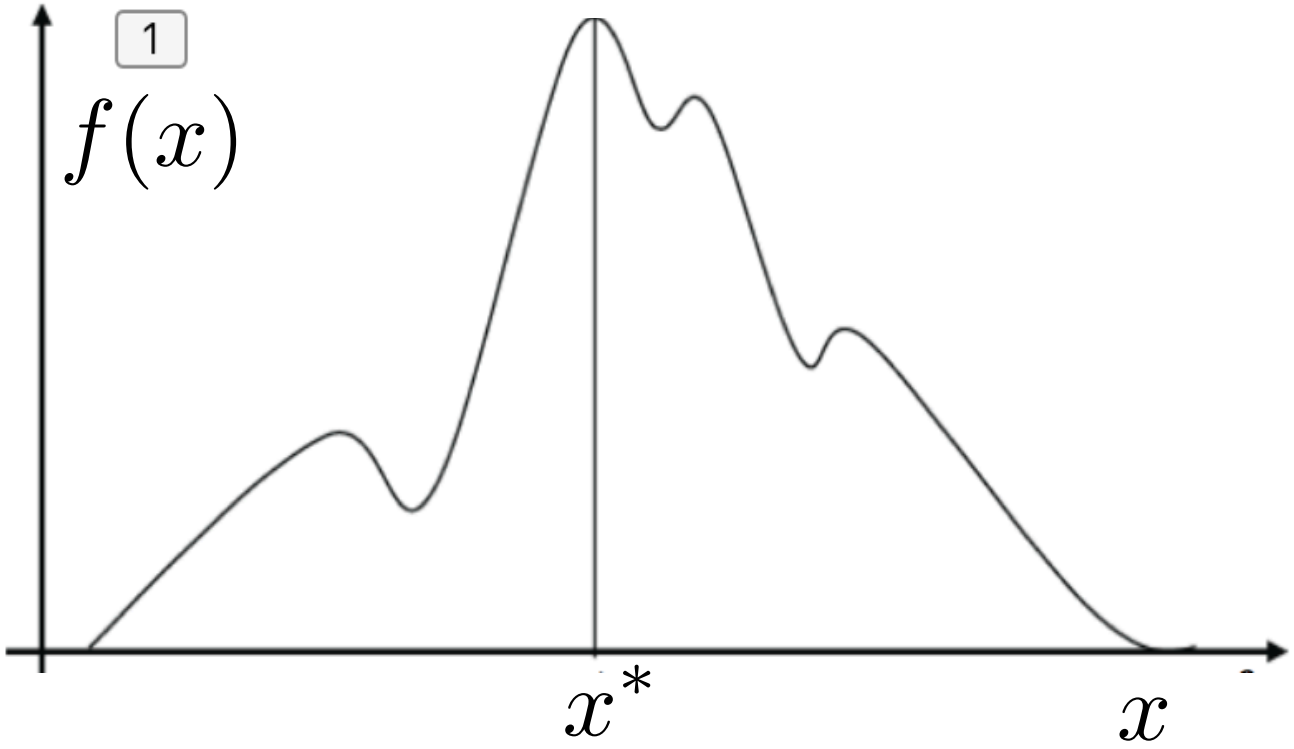
Expressed as a function of two variables:

$$L(w_0, w_1) = \sum_{i=1}^{N} \left[y^i - \left(w_0 x_0^i + w_1 x_1^i \right) \right]^2$$

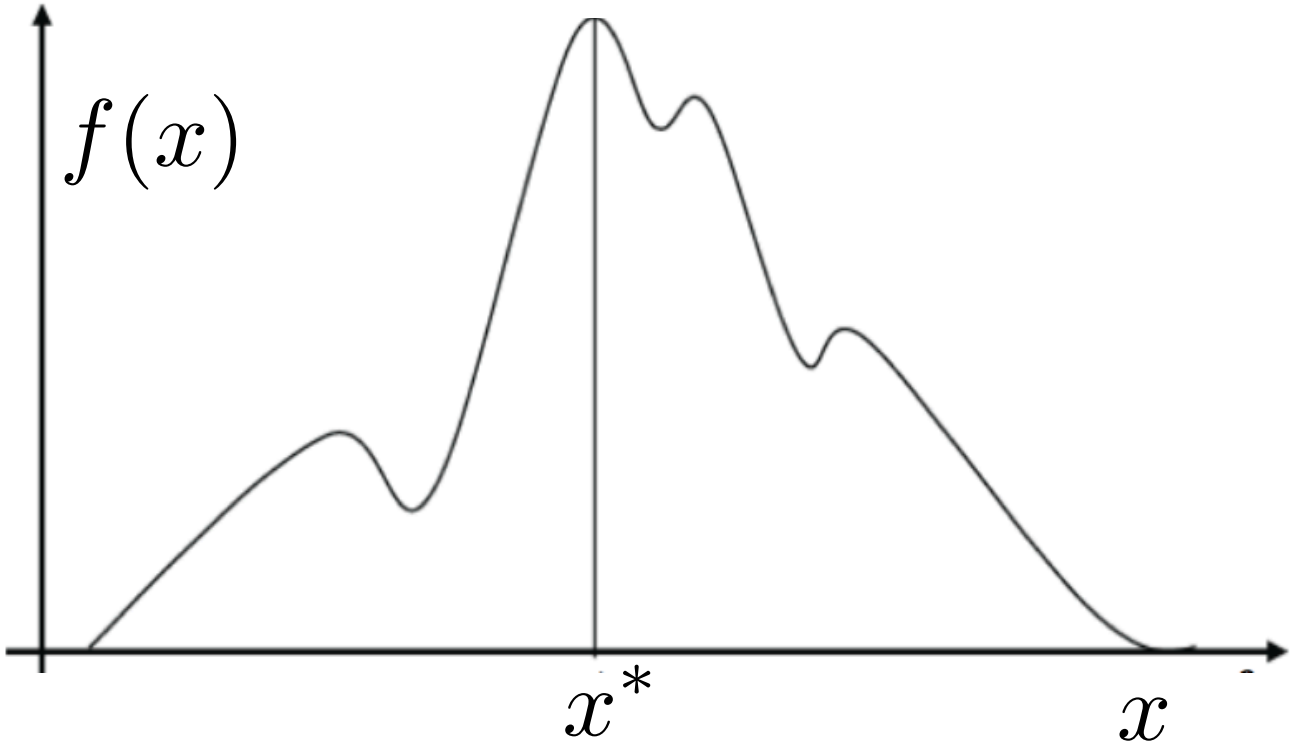
Question: what is the best (or least bad) value of w?

Answer: least squares

Calculus 101

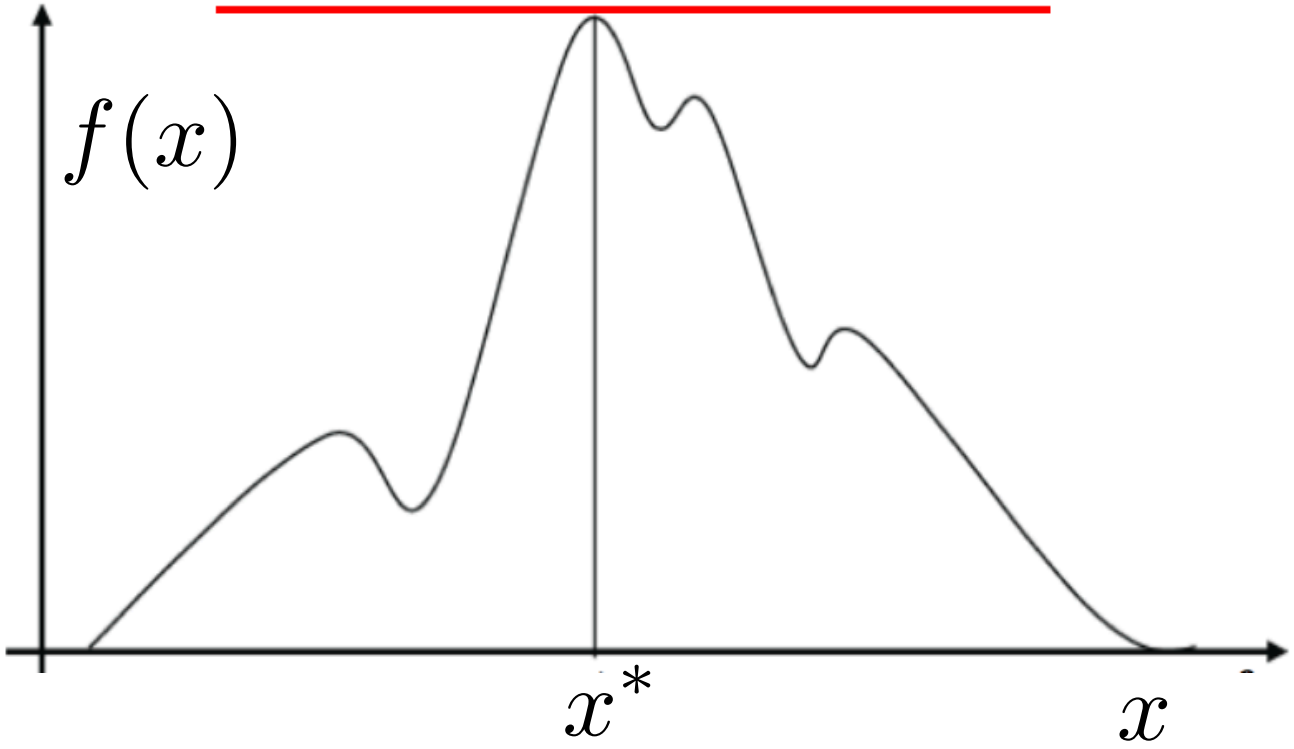


Calculus 101



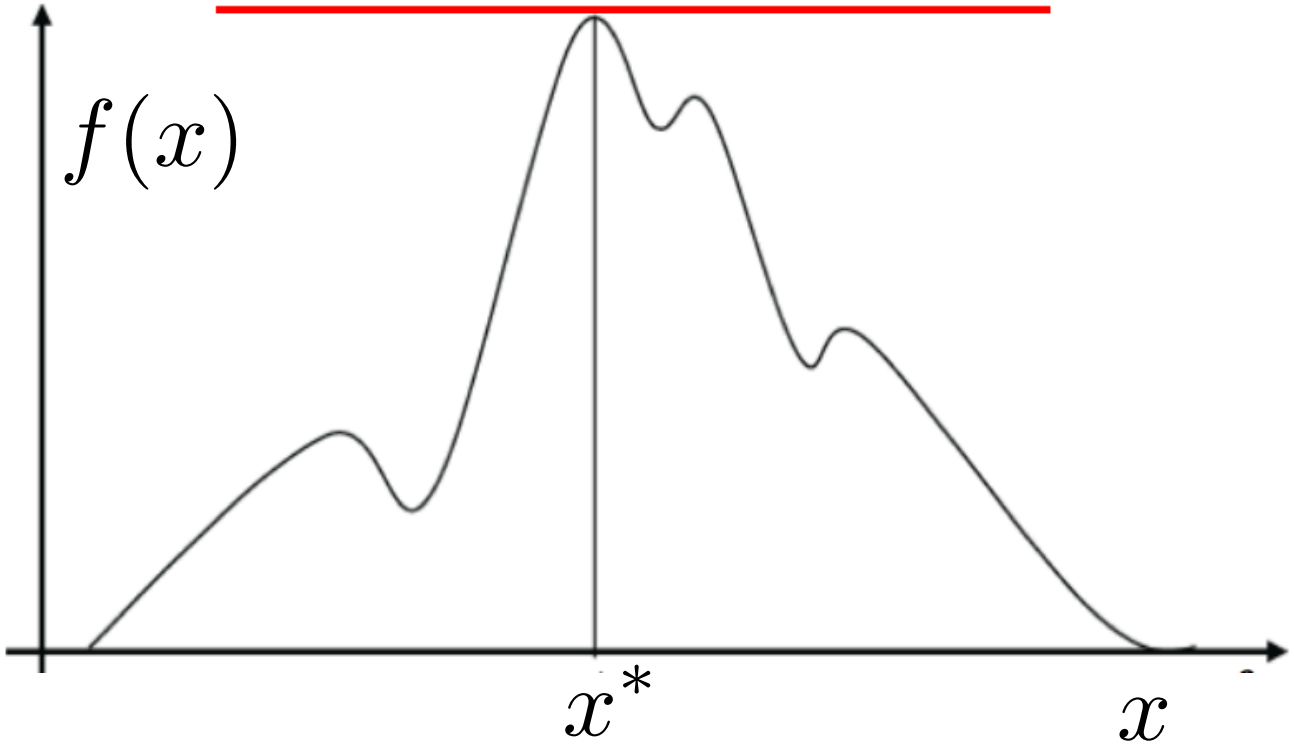
$$x^* = \text{argmax}_x f(x)$$

Condition for maximum: derivative is zero



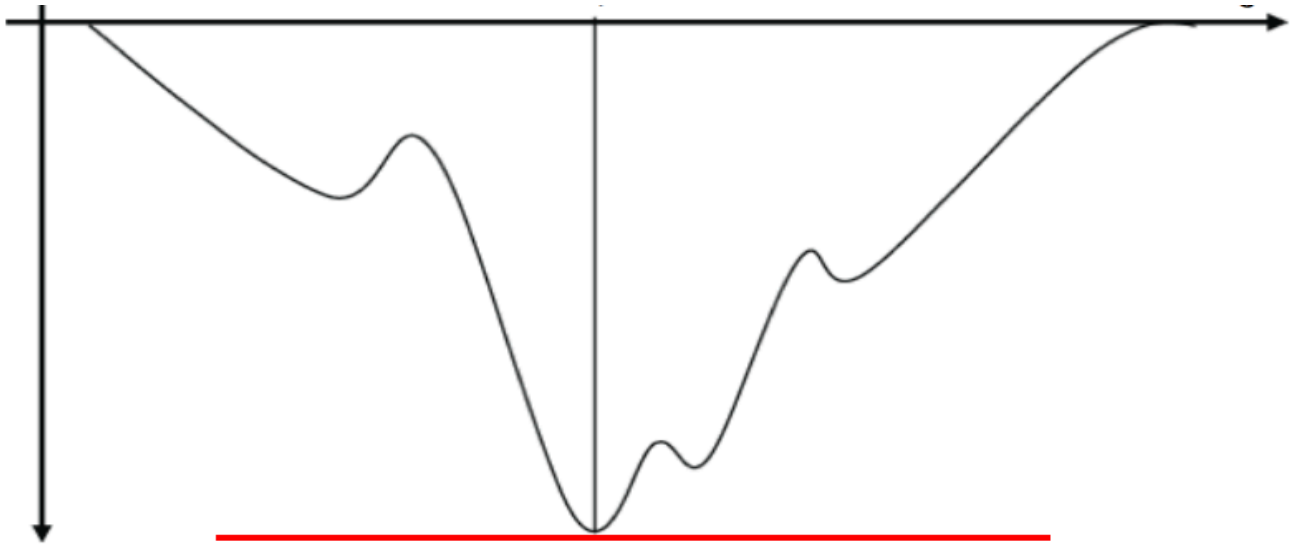
$$x^* = \text{argmax}_x f(x)$$

Condition for maximum: derivative is zero



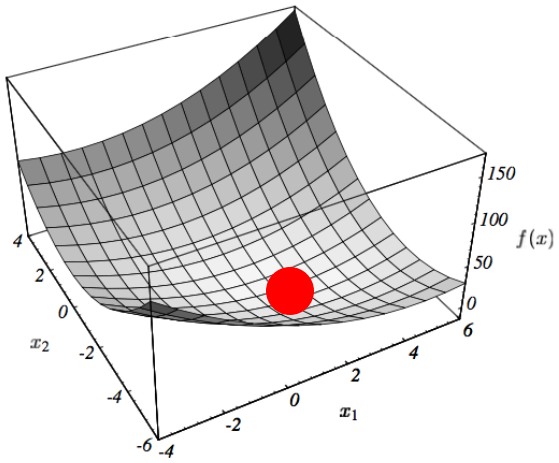
$$x^* = \text{argmax}_x f(x) \quad \rightarrow \quad f'(x^*) = 0$$

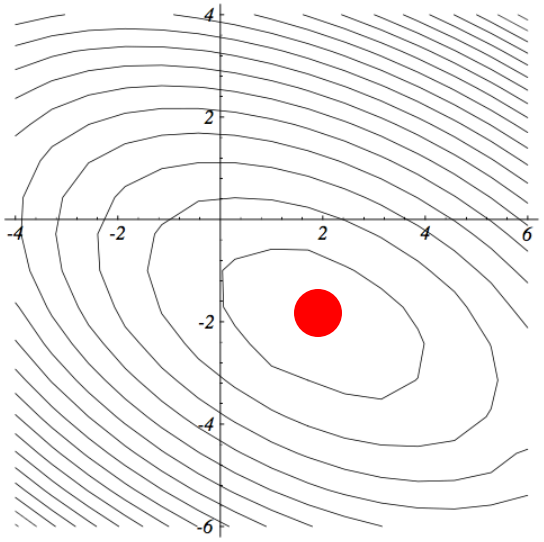
Condition for minimum: derivative is zero

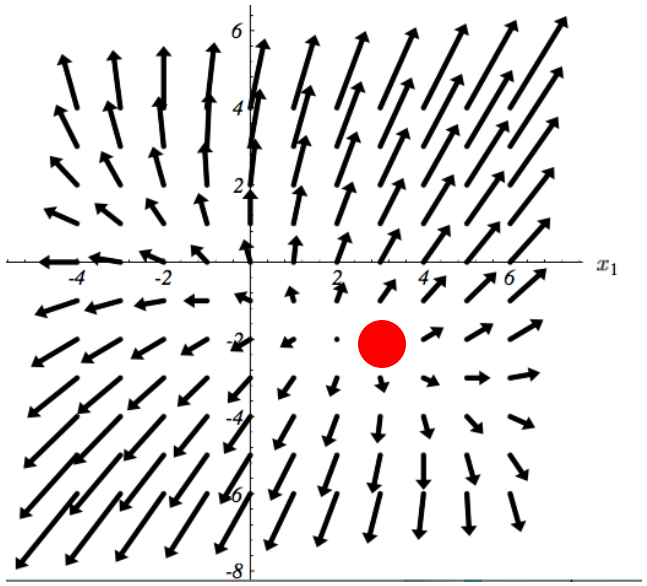


$$x^* = \text{argmin}_x f(x) \quad \rightarrow \quad f'(x^*) = 0$$

Vector calculus 101







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

2D function graph

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

isocontours

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

gradient field

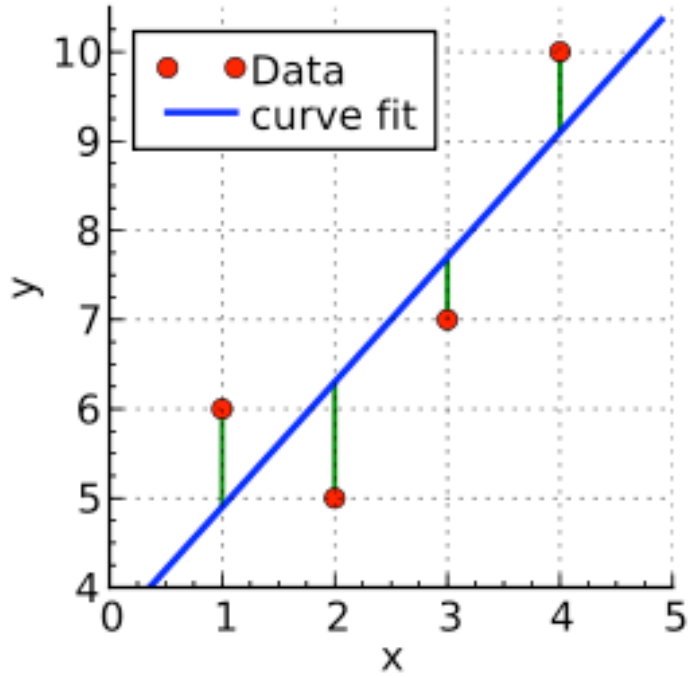
● **at minimum of function:** $\nabla f(\mathbf{x}) = \mathbf{0}$

Back to least squares..

$$y^i = \mathbf{w}^T \mathbf{x}^i + \epsilon^i$$

Loss function: sum of squared errors

$$L(\mathbf{w}) = \sum_{i=1}^{N} (\epsilon^i)^2$$



Expressed as a function of two variables:

$$L(w_0, w_1) = \sum_{i=1}^{N} \left[y^i - \left(w_0 x_0^i + w_1 x_1^i \right) \right]^2$$

training sample

feature dimension

Question: what is the best (or least bad) value of w?

Answer: least squares

Fitting a line

$$L(w_0, w_1) = \sum_{i=1}^{N} \left[y^i - \left(w_0 x_0^i + w_1 x_1^i \right) \right]^2$$

$$\frac{\partial L(w_0, w_1)}{\partial w_0} = \sum_{i=1}^{N} \frac{\partial \left[y^i - \left(w_0 x_0^i + w_1 x_1^i \right) \right]^2}{\partial w_0}$$

$$= \sum_{i=1}^{N} 2 \left[y^i - \left(w_0 x_0^i + w_1 x_1^i \right) \right] (-x_0^i)$$

$$= -2 \sum_{i=1}^{N} \left(y^i x_0^i - w_0 x_0^i x_0^i - w_1 x_1^i x_0^i \right)$$

$$\frac{\partial L(w_0, w_1)}{\partial w_0} = 0_N \Leftrightarrow$$

$$\sum_{i=1}^{N} y^i x_0^i = w_0 \sum_{i=1}^{N} x_0^i x_0^i + w_1 \sum_{i=1}^{N} x_1^i x_0^i$$

Fitting a line, continued

$$\frac{\partial L(w_0, w_1)}{\partial w_0} = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{N} y^i x_0^i = w_0 \sum_{i=1}^{N} x_0^i x_0^i + w_1 \sum_{i=1}^{N} x_1^i x_0^i$$

$$\frac{\partial L(w_0, w_1)}{\partial w_1} = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{N} y^i x_1^i = w_0 \sum_{i=1}^{N} x_0^i x_1^i + w_1 \sum_{i=1}^{N} x_1^i x_1^i$$

2 linear equations, 2 unknowns

Fitting a line, continued

$$\sum_{i=1}^{N} y^i x_0^i = w_0 \sum_{i=1}^{N} x_0^i x_0^i + w_1 \sum_{i=1}^{N} x_1^i x_0^i$$

$$\sum_{i=1}^{N} y^i x_1^i = w_0 \sum_{i=1}^{N} x_0^i x_1^i + w_1 \sum_{i=1}^{N} x_1^i x_1^i$$

2x2 system of equations:

$$\begin{bmatrix} \sum_{i=1}^{N} y^i x_0^i \\ \sum_{i=1}^{N} y^i x_1^i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_0^i x_0^i & \sum_{i=1}^{N} x_0^i x_1^i \\ \sum_{i=1}^{N} x_0^i x_1^i & \sum_{i=1}^{N} x_1^i x_1^i \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

That's it!

Fitting a line, continued

2x2 system of equations:

$$\begin{bmatrix} \sum_{i=1}^{N} y^i x_0^i \\ \sum_{i=1}^{N} y^i x_1^i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_0^i x_0^i & \sum_{i=1}^{N} x_0^i x_1^i \\ \sum_{i=1}^{N} x_0^i x_1^i & \sum_{i=1}^{N} x_1^i x_1^i \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

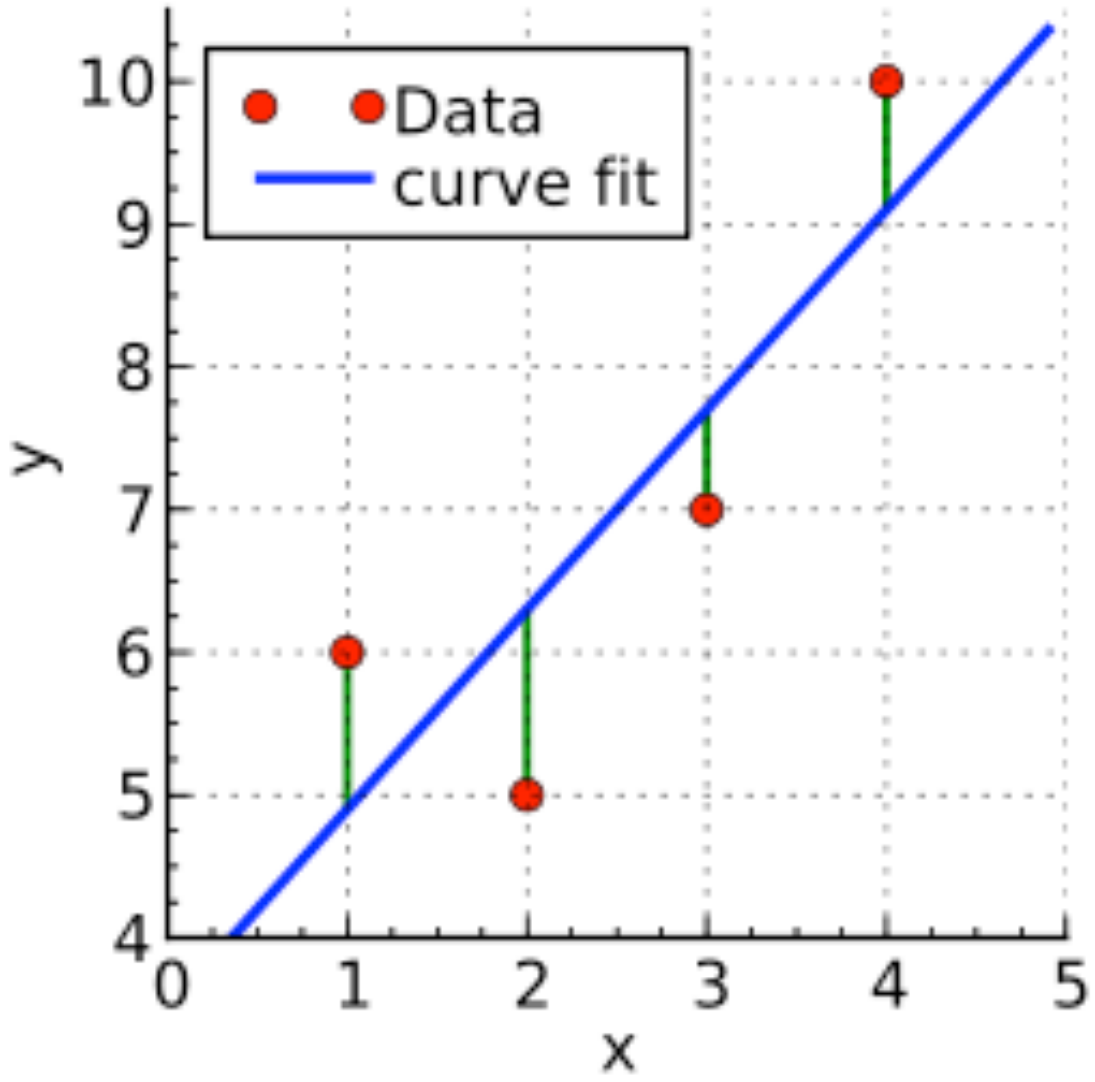
Or, without summations:

$$\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{X}\mathbf{w}$$

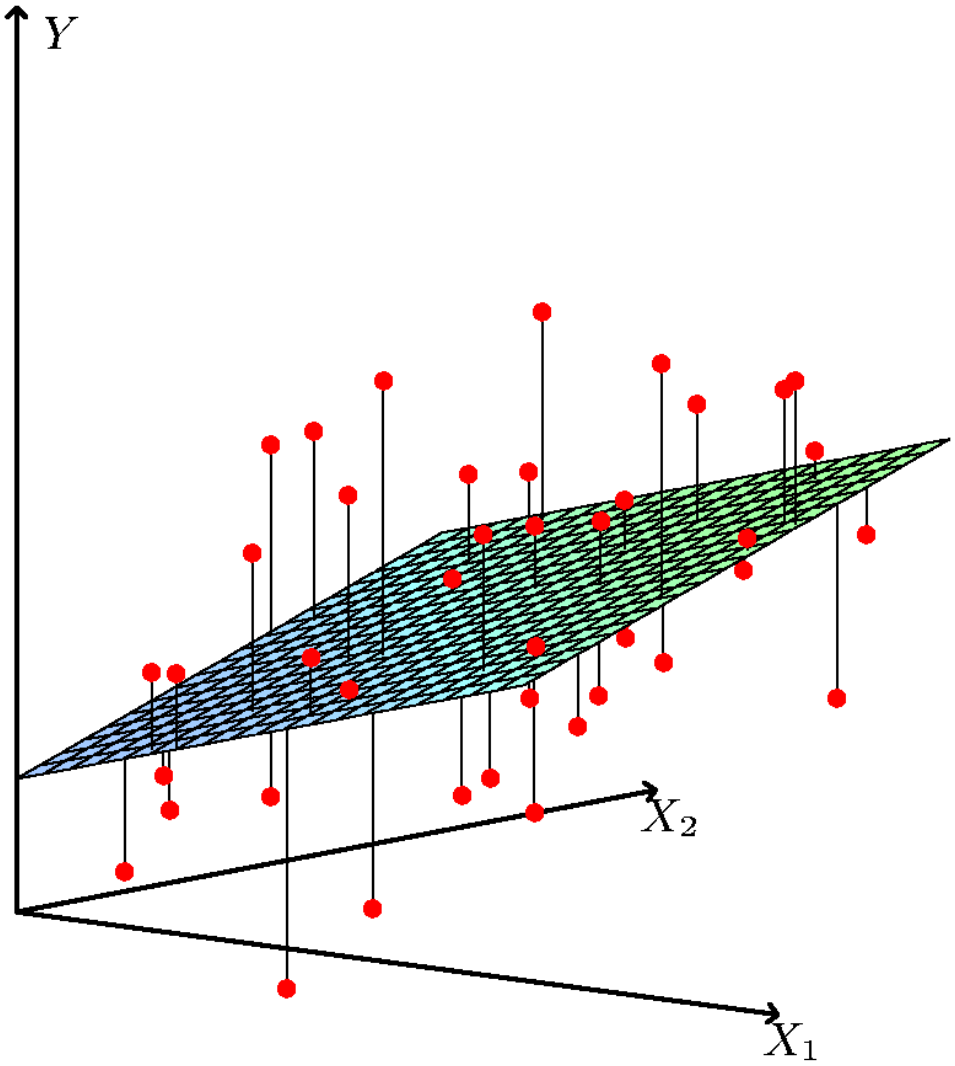
$$\mathbf{y} = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_0^1 & x_1^1 \\ \vdots & \vdots \\ x_0^N & x_2^N \end{bmatrix}$$

Solution: $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Linear regression in 1D



Linear regression in 2D (or ND)



Least squares solution for linear regression

D: problem dimension

N: training set size

$$\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} = \begin{bmatrix} x_1^1 & \dots & x_D^1 \\ x_1^2 & \dots & x_D^2 \\ \vdots & & \\ x_1^N & \dots & x_D^N \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} + \begin{bmatrix} \epsilon^1 \\ \epsilon^2 \\ \vdots \\ \epsilon^N \end{bmatrix}$$

Nx1 **NxD** **Dx1** **Nx1**

Least squares solution for linear regression

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

Least squares solution for linear regression

Loss function: $L(\mathbf{w}) = \sum_{i=1}^{N} (y^i - \mathbf{w}^T \mathbf{x}^i)^2 = \sum_{i=1}^{N} (\epsilon^i)^2$

$$L(\mathbf{w}) = \begin{bmatrix} \epsilon^1 & \epsilon^2 & \dots & \epsilon^N \end{bmatrix} \begin{bmatrix} \epsilon^1 \\ \epsilon^2 \\ \vdots \\ \epsilon^N \end{bmatrix}$$

Least squares solution for linear regression

Loss function: $L(\mathbf{w}) = \sum_{i=1}^{N} (y^i - \mathbf{w}^T \mathbf{x}^i)^2 = \sum_{i=1}^{N} (\epsilon^i)^2$

$$L(\mathbf{w}) = \begin{bmatrix} \epsilon^1 & \epsilon^2 & \dots & \epsilon^N \end{bmatrix} \begin{bmatrix} \epsilon^1 \\ \epsilon^2 \\ \vdots \\ \epsilon^N \end{bmatrix}$$

$$L(\mathbf{w}) = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \qquad \mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$