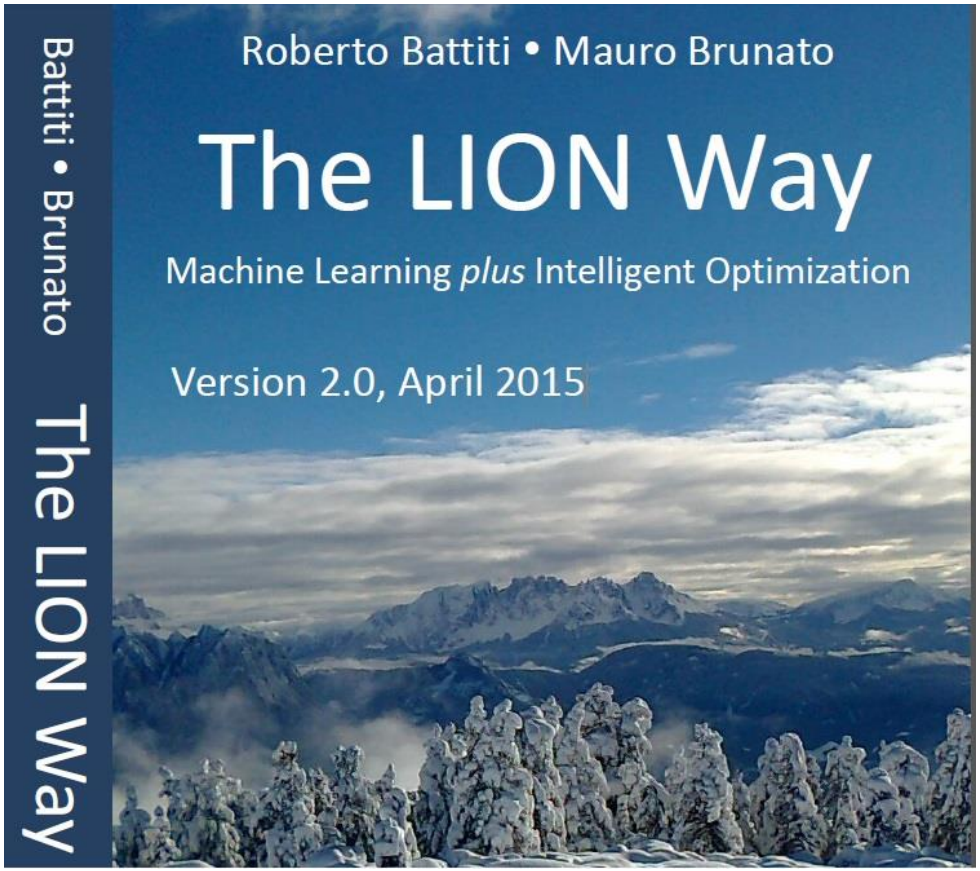
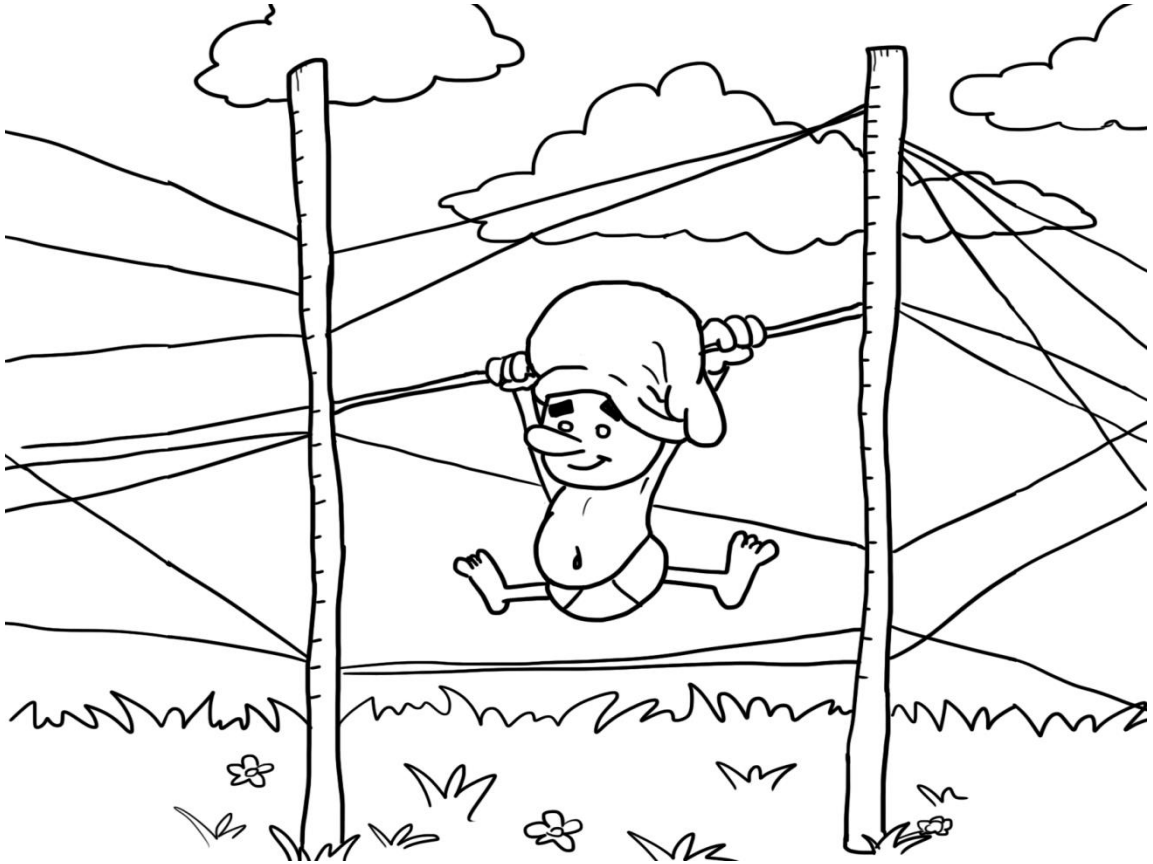


ROBERTO BATTITI, MAURO BRUNATO.
The LION Way: Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy,
Apr 2015

http://intelligent-optimization.org/LIONbook

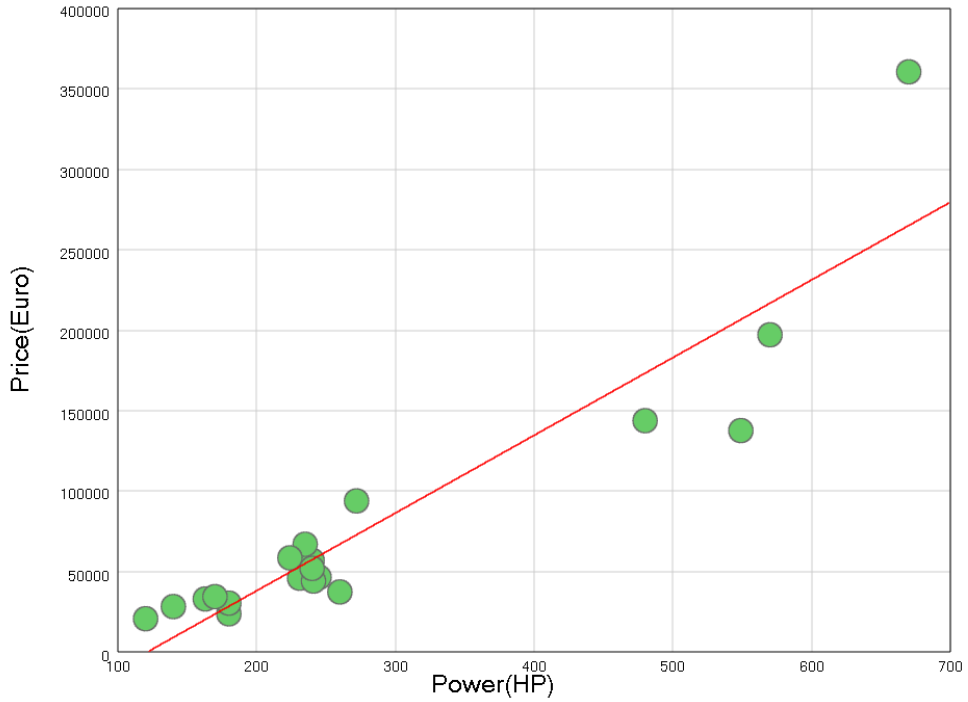
Chap.4 Linear models

Most right-handed people are linear thinking, think inside the box.



Linear models

- Just below the mighty power of optimization lies the awesome power of linear algebra.



Data about price and power of different car models. A **linear model (fit)** is shown.

Linear regression

- A linear dependence of the output from the input features

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d.$$

- The model is simple, can be easily trained,
- The computed **weights** in the linear summation provide a direct explanation of the importance of the various attributes

Best (linear) fit → optimization

- Errors can be present (every physical quantity can be measured only with a finite precision)

$$y_i = \boldsymbol{w}^T \cdot \boldsymbol{x}_i + \epsilon_i,$$

- Determine optimal weight vector **w** so that: $\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \cdot \boldsymbol{x}$

approximates as closely as possible the experimental data

- minimizes the sum of the squared errors (least squares approximation):

$$\text{ModelError}(\boldsymbol{w}) = \sum_{i=1}^{\ell} (\boldsymbol{w}^T \cdot \boldsymbol{x}_i - y_i)^2.$$

Least-squares

- In the unrealistic case of zero measurement errors and a perfect linear model, one is left with **a set of linear equations**:

$$\boldsymbol{w}^T \cdot \boldsymbol{x}_i = y_i,$$

- In all real-world cases measurement errors are present, and the number of measurements (xi; yi) can be much larger than the input dimension. Therefore one needs to search for an approximated solution, for weights **w** obtaining the lowest possible value of the error.
- How? Trust optimization (this case is standard! Use *pseudo-inverse* in linear algebra, see later)

A trick for nonlinear dependencies

- f(**x**) = $\mathbf{w}^T$ **x** is too restrictive, In particular, it assumes that f(0) = 0
- f(**x**) = w_0 + $\mathbf{w}^T$ **x** (affine model) by adding a dimension and fixing the value to 1:

  x = (1; x1; …; xd)
- Linear model on nonlinear features $\phi(\boldsymbol{x})$.

$$\phi_1, \ldots, \phi_n : \mathbb{R}^d \to \mathbb{R}^n$$

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^T \cdot \phi(\boldsymbol{x}).$$

A trick for nonlinear dependencies

- E.g., a quadratic model with tow inputs is linear in the following features

$$\phi_1(\boldsymbol{x}) = 1, \quad \phi_2(\boldsymbol{x}) = x_1, \quad \phi_3(\boldsymbol{x}) = x_2,$$

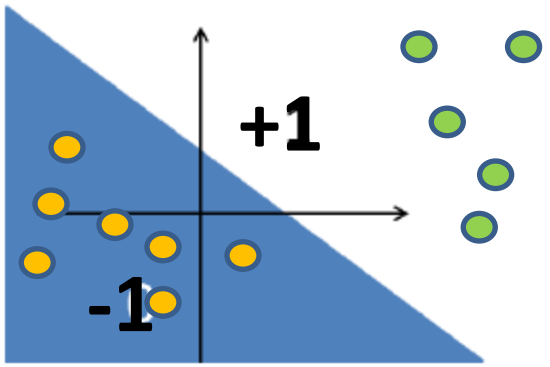
$$\phi_4(\boldsymbol{x}) = x_1 x_2, \quad \phi_5(\boldsymbol{x}) = x_1^2, \quad \phi_6(\boldsymbol{x}) = x_2^2.$$

Linear models for classification

- Let the outcome variable be two-valued (e.g., +/- 1). Linear functions can be used as **discriminants**
- **hyperplane** defined by a vector **w** separating the two classes

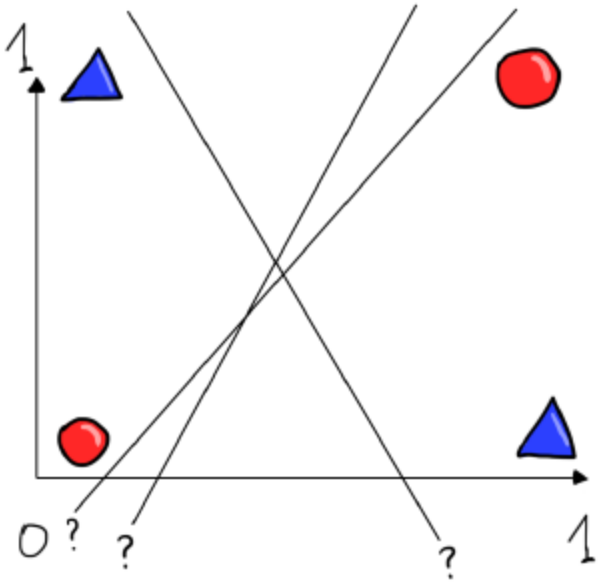
(examples marked as red / orange)

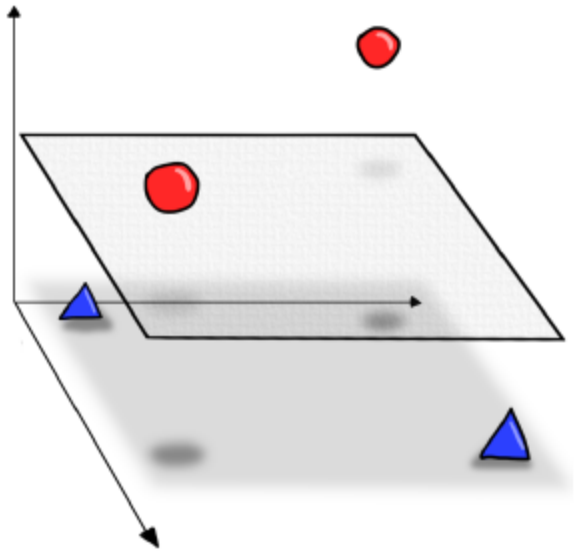
$$y = \begin{cases} +1 & \text{if } \boldsymbol{w}^T \cdot \boldsymbol{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$



Counter-example: XOR function

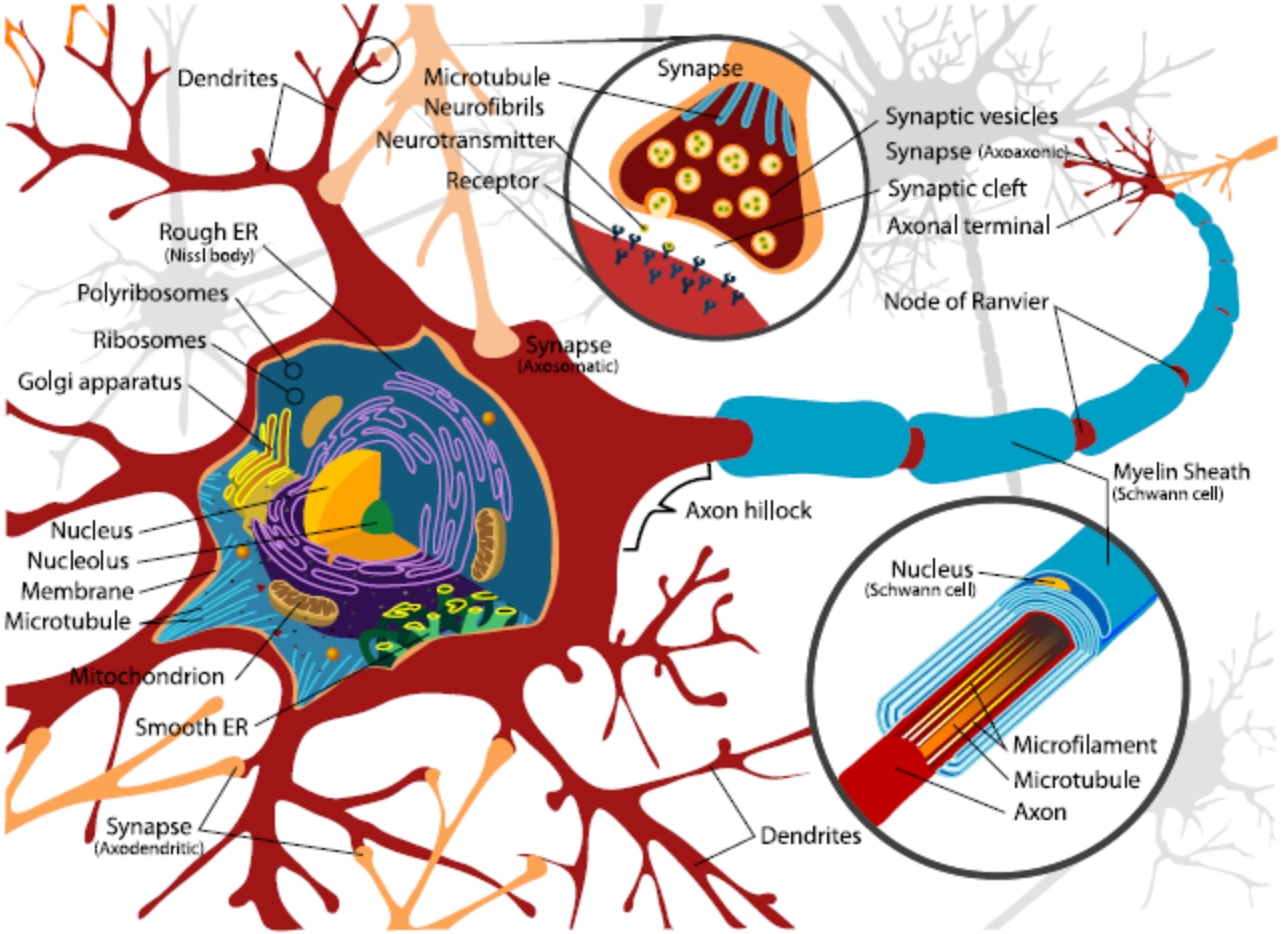
Cannot separate points with a line in two dim.





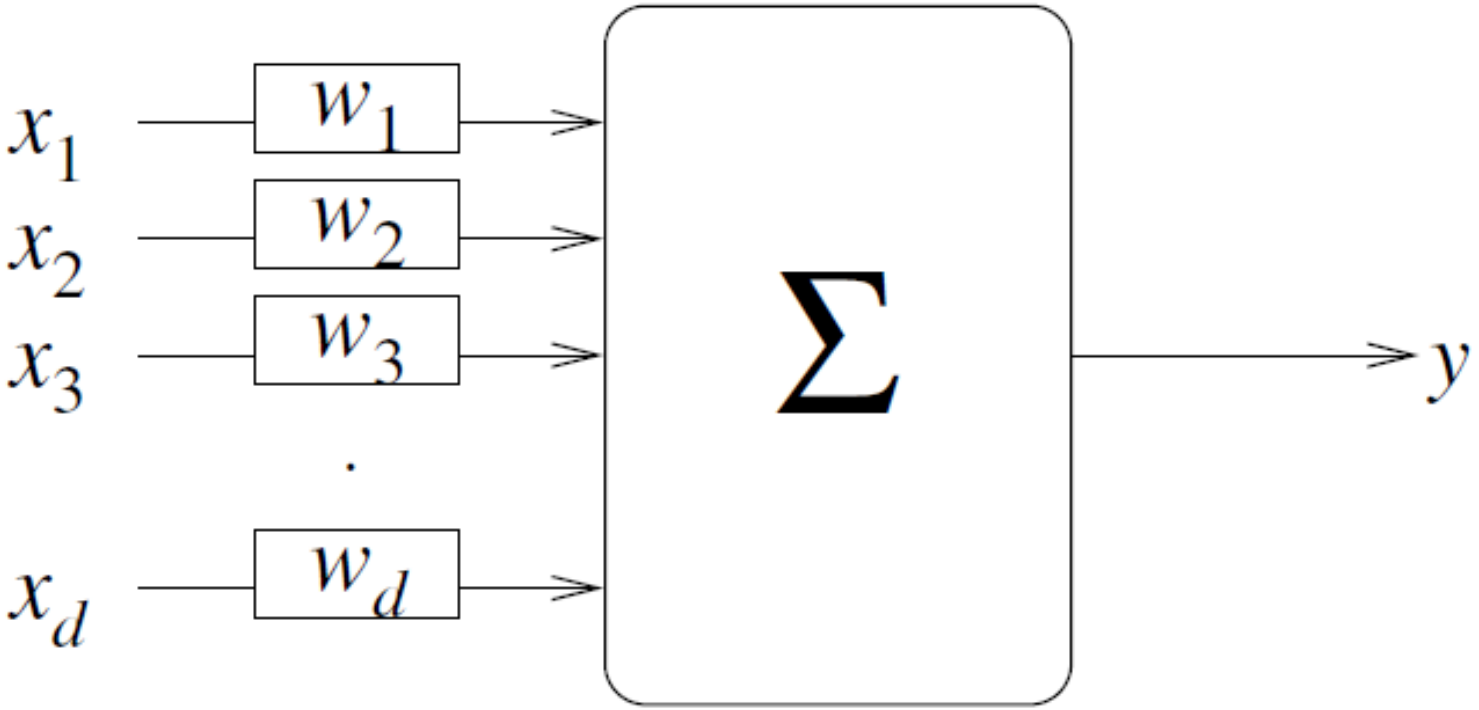
But points *can* be separated by a plane after mapping them into 3D

Biological motivations



Neurons and synapses in the human brain

Abstract model: the perceptron



- Output is a weighted sum of the inputs passed through a final threshold function.

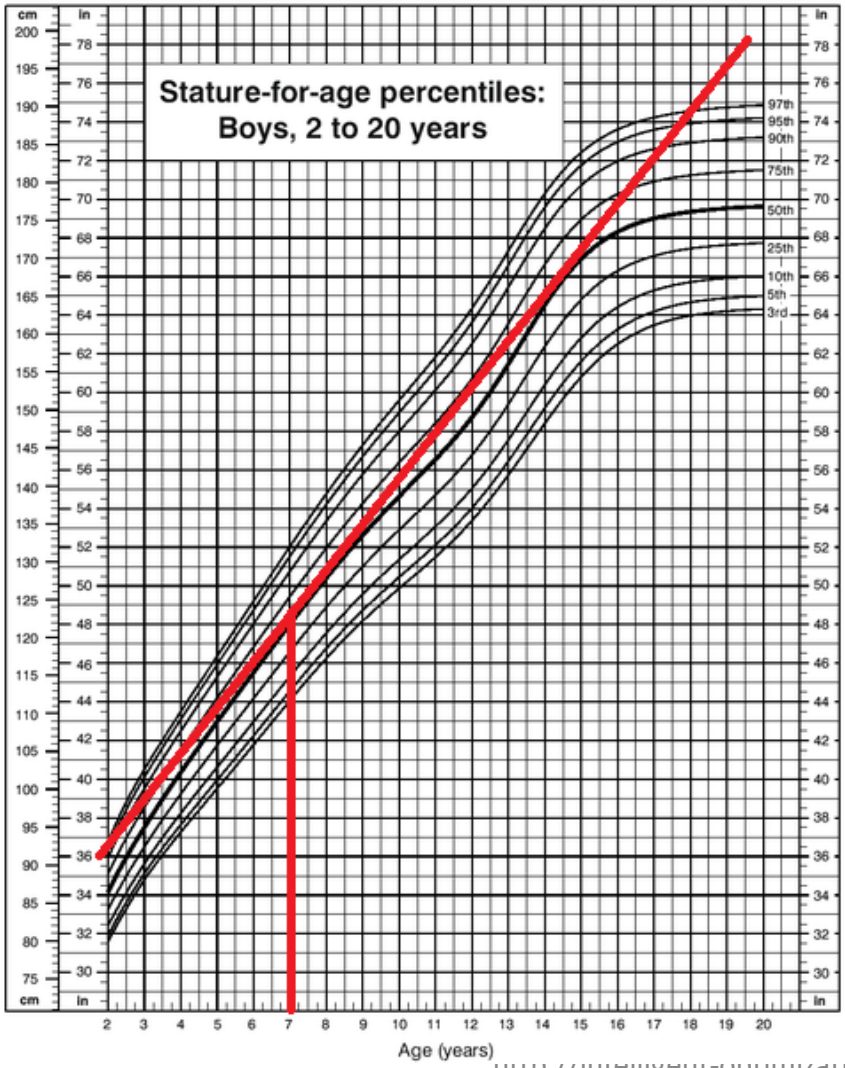
Why are linear models successful?

- smoothness underlying many physical phenomena
- every smooth (differentiable) function can be approximated around an operating point x_c with its Taylor series approximation

$$f(x) = f(x_c) + \nabla f(x_c) \cdot (x - x_c) + O(\|x - x_c\|^2).$$

Linear part

Why are linear models popular and successful?



Example:

The stature-for-age curve can be approximated well by a tangent line (red) from 2 to about 15 years.

Minimizing the sum of squared errors

- If zero measurement errors and a perfect linear model:

set of linear equations $\mathbf{w}^T \mathbf{x}_i = y_i$ one for each example

- in real-world cases, reaching zero for the ModelError is impossible, and *the number of data points can be much larger than the number of parameters* d.

- furthermore, the goal of learning is *generalization*

(and requiring zero error can cause "overtraining")

Minimizing the sum of squared errors

- Error is quadratic in parameters **w**

$$\text{ModelError}(\boldsymbol{w}) = \sum_{i=1}^{\ell} (\boldsymbol{w}^T \cdot \boldsymbol{x}_i - y_i)^2.$$

- Find minimum:
  - take partial derivatives
  - equate them to 0

- Obtain linear equations, typically *more* equations than examples

Minimizing the sum of squared errors

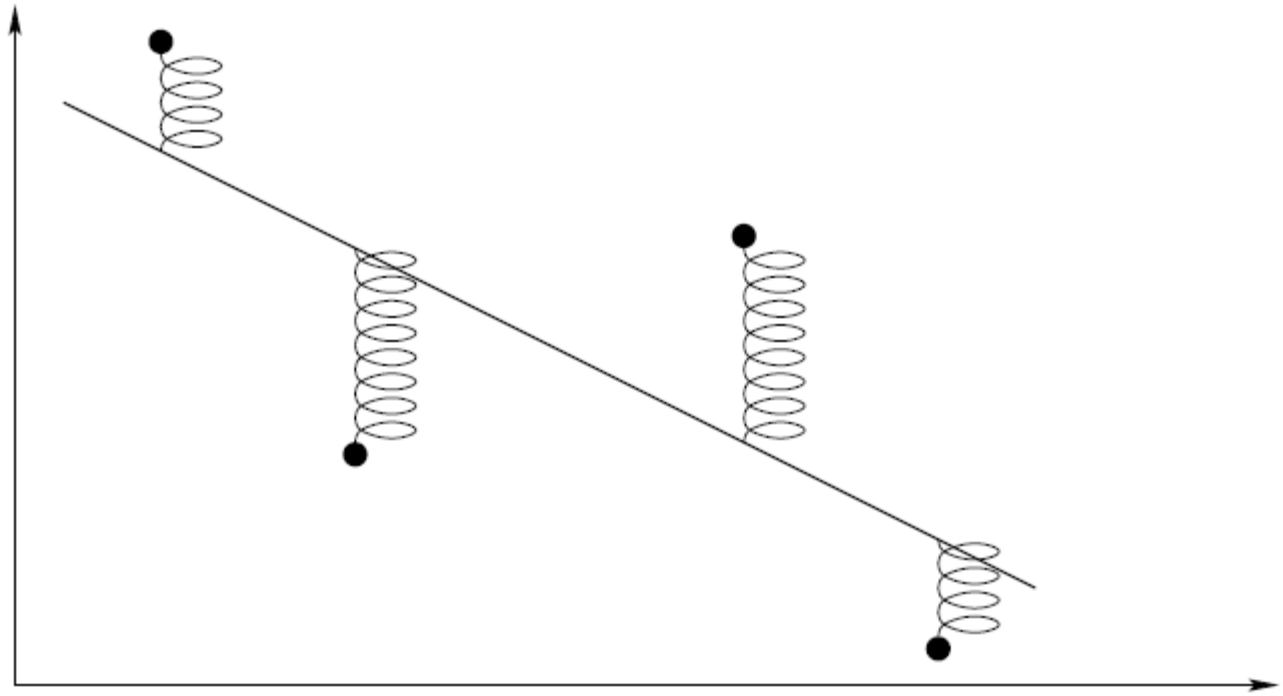
- From inverse to pseudo-inverse, solution is:

$$\boldsymbol{w}^* = (X^T X)^{-1} X^T \boldsymbol{y};$$

where y = (y_1; …; y_L) and **X** is the matrix whose rows are the $\mathbf{x}_i$ vectors.

- least-square and pseudo-inverse are among the most popular tools
- alternative is gradient descent

An analogy in Physics



Spring analogy for least squares fits.

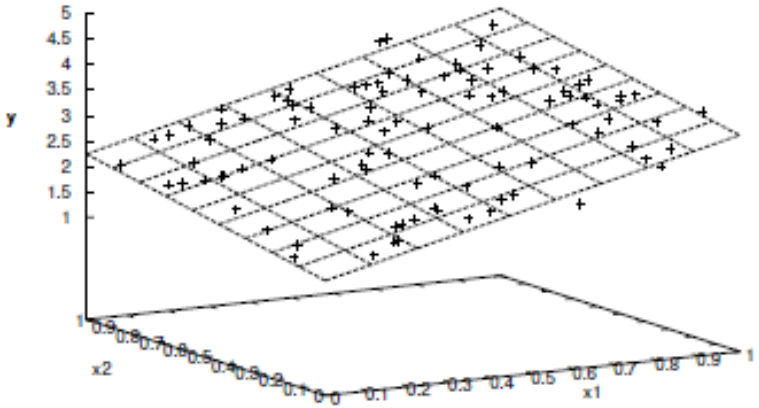
Springs connect a rigid bar to the experimental points.

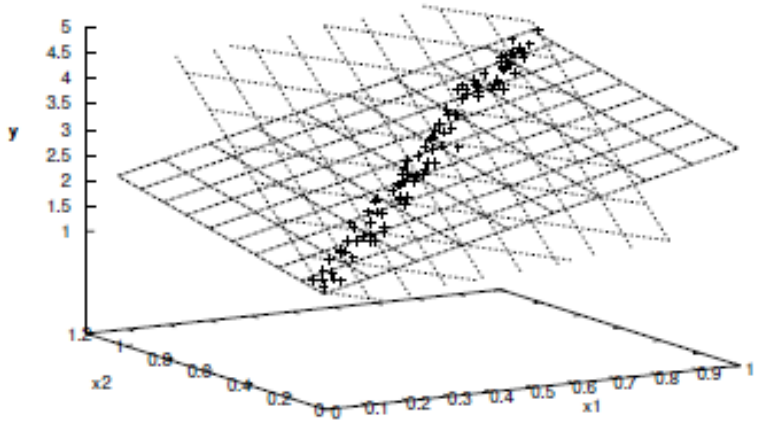
The best fit is the line that **minimizes the overall potential energy of the system** (proportional to thesum of the squares of the spring length).

Numerical instabilities

- Each number is assigned a fixed number of bits, no way to represent 3.14159265...
- **Mistakes will propagate** during mathematical operations
- **Stability** here means that **small perturbations** of the sample points lead to **small changes** in the results

Numerical instabilities





A well-spread training set (left) provides a stable numerical model, whereas a bad choice of sample points (right) may result in **wildly changing planes**, including very steep ones

Ridge regression to cure instability

Regularization term

$$\text{error}(\boldsymbol{w};\lambda) = \sum_{i=1}^{\ell} (\boldsymbol{w}^T \cdot \boldsymbol{x}_i - y_i)^2 + \lambda \boldsymbol{w}^T \cdot \boldsymbol{w}.$$

The minimization with respect to $\boldsymbol{w}$ leads to the following:

$$\boldsymbol{w}^* = (\lambda I + X^T X)^{-1} X^T \boldsymbol{y}.$$

The insertion of a small diagonal term makes the inversion more robust.

Gist

- Traditional linear models for regression (linear approximation of a set of input/output pairs) identify the best possible linear fit of experimental data by **minimizing a sum the squared errors** between the values predicted by the linear model and the training examples.
- **Minimization can be "one shot" by generalizing matrix inversion in linear algebra**, or iteratively, by gradually modifying the model parameters to
- lower the error. The **pseudo-inverse** method is possibly the most used technique for fitting experimental data.

Gist (2)

- In classification, linear models aim at **separating examples with lines, planes and hyper-planes**. To identify a separating plane one can require a mapping of the inputs to two distinct output values (like +1 and -1) and use regression.
- Real numbers in a computer do not exist and their approximation by limited-size binary numbers is a possible cause of mistakes and **instability** (small perturbations of the sample points leading to large changes in the results).
- Some machine learning methods are loosely related to the way in which biological brains work.