session 1:

Basic Predictive Models

M. Kundegorski

13th December 2019

IAFIG-RMS - Bioimage Analysis With Python

Cambridge Bioinformatics Training Centre

Basic Models

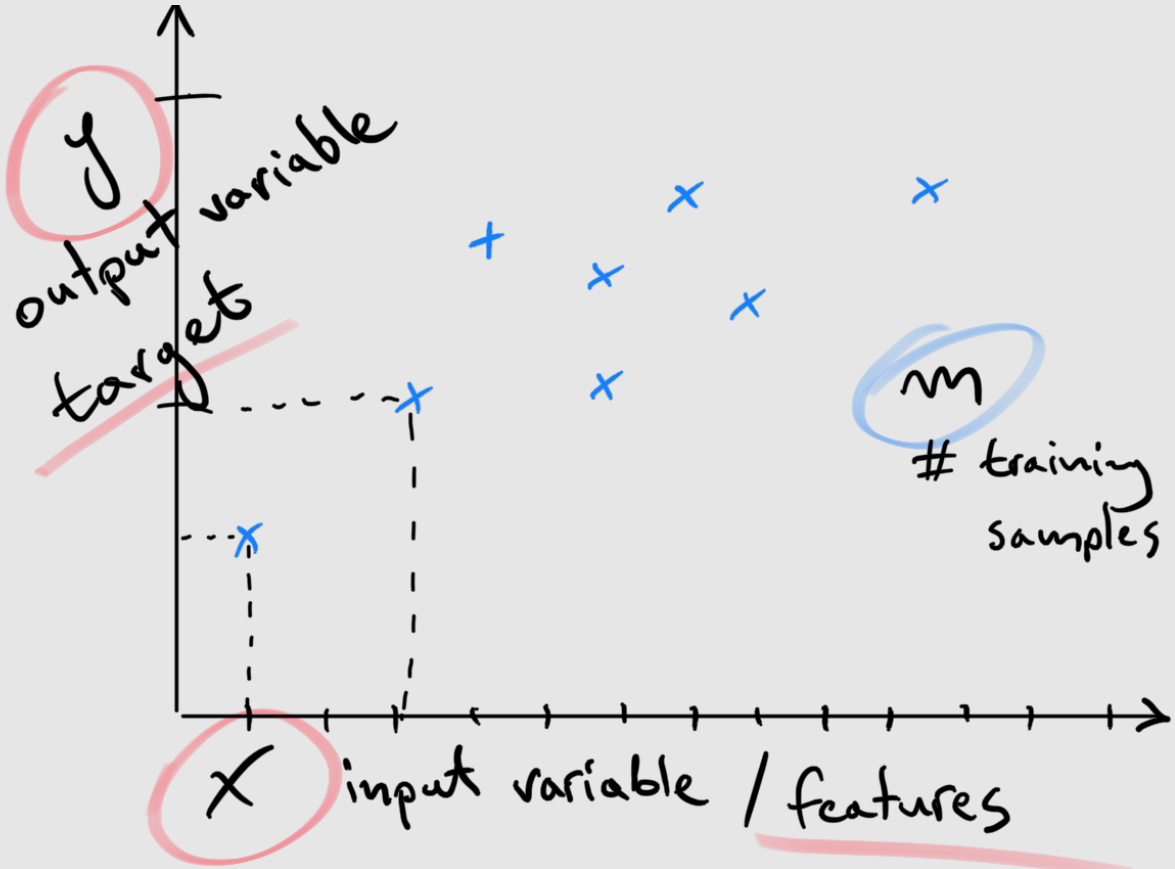
Statistics from a different point of view

In this lecture

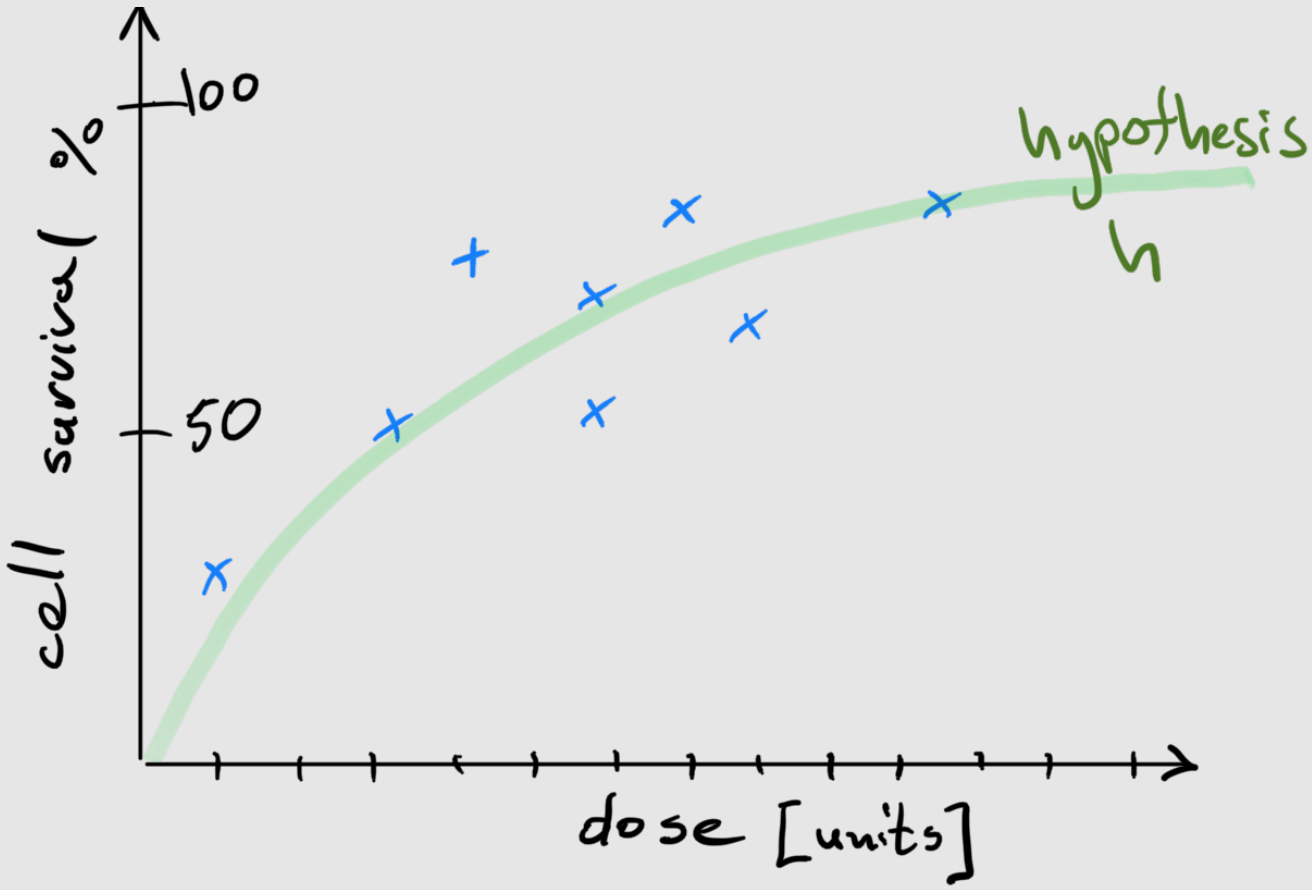
- Linear regression
- Multivariate regression
- Generalised linear regression
- Model fitting

Based on excellent online course by Andrew Ng:
https://www.coursera.org/learn/machine-learning/

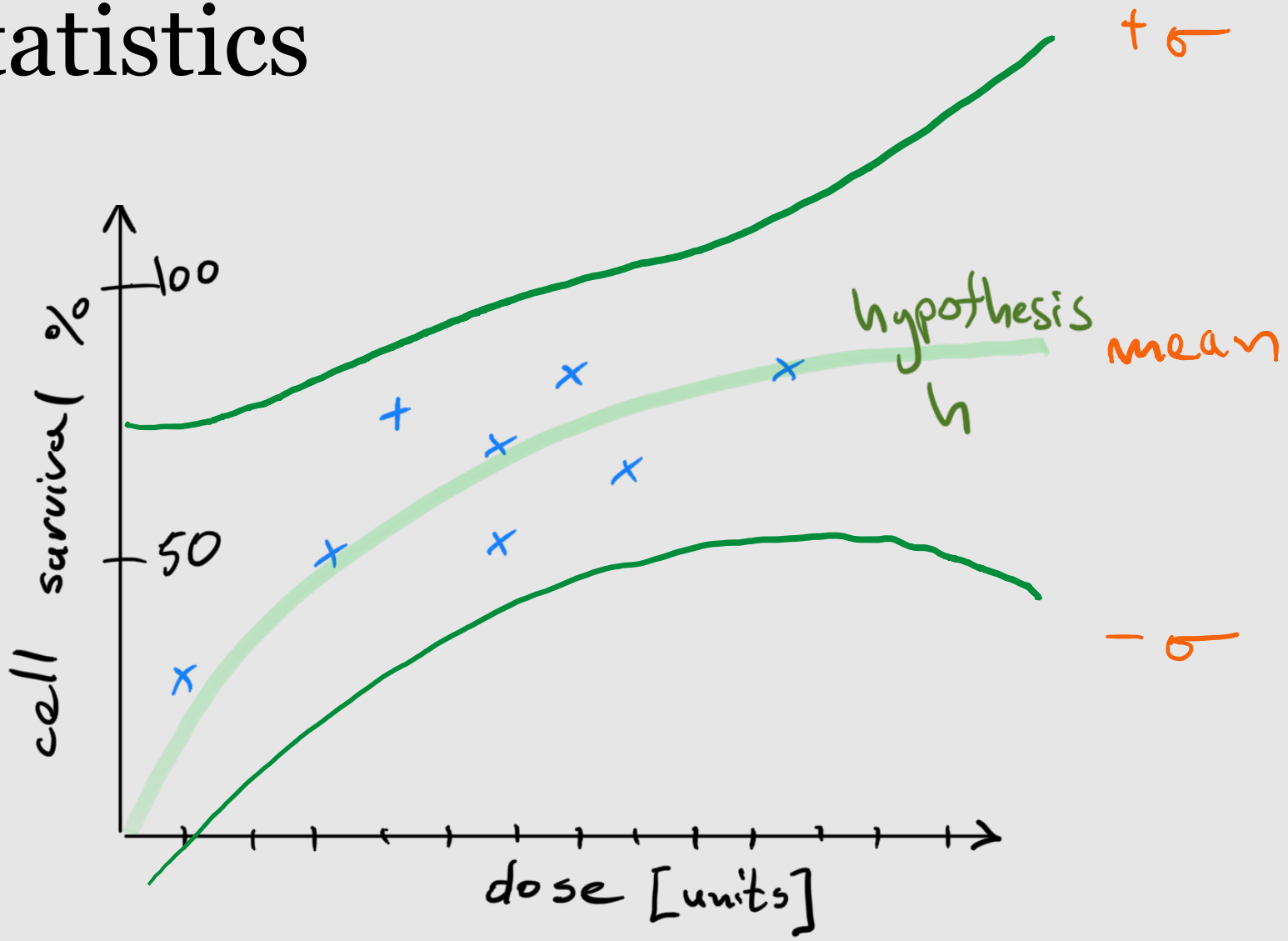
A problem


y
output variable
target
m
training samples
x
input variable / features

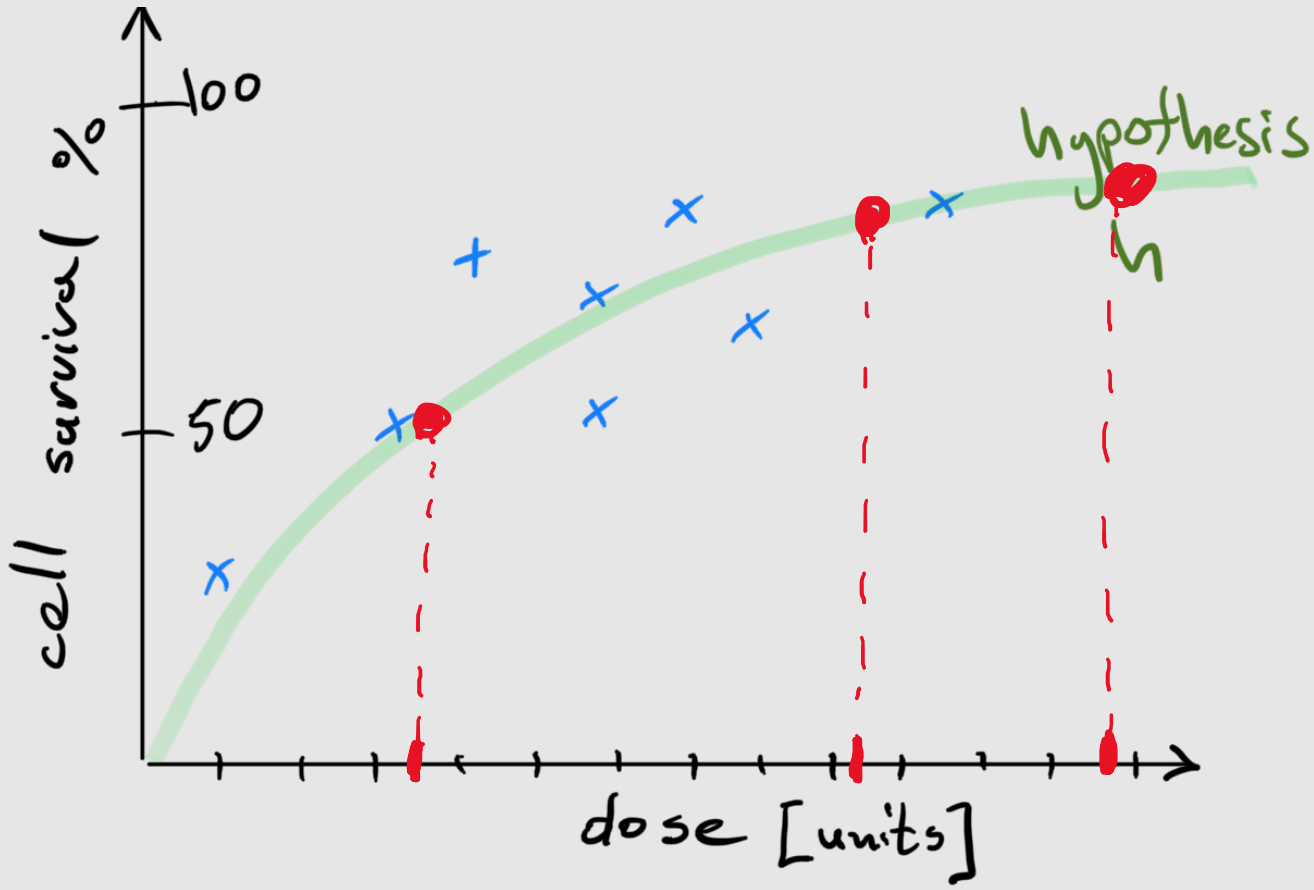
Model Representation



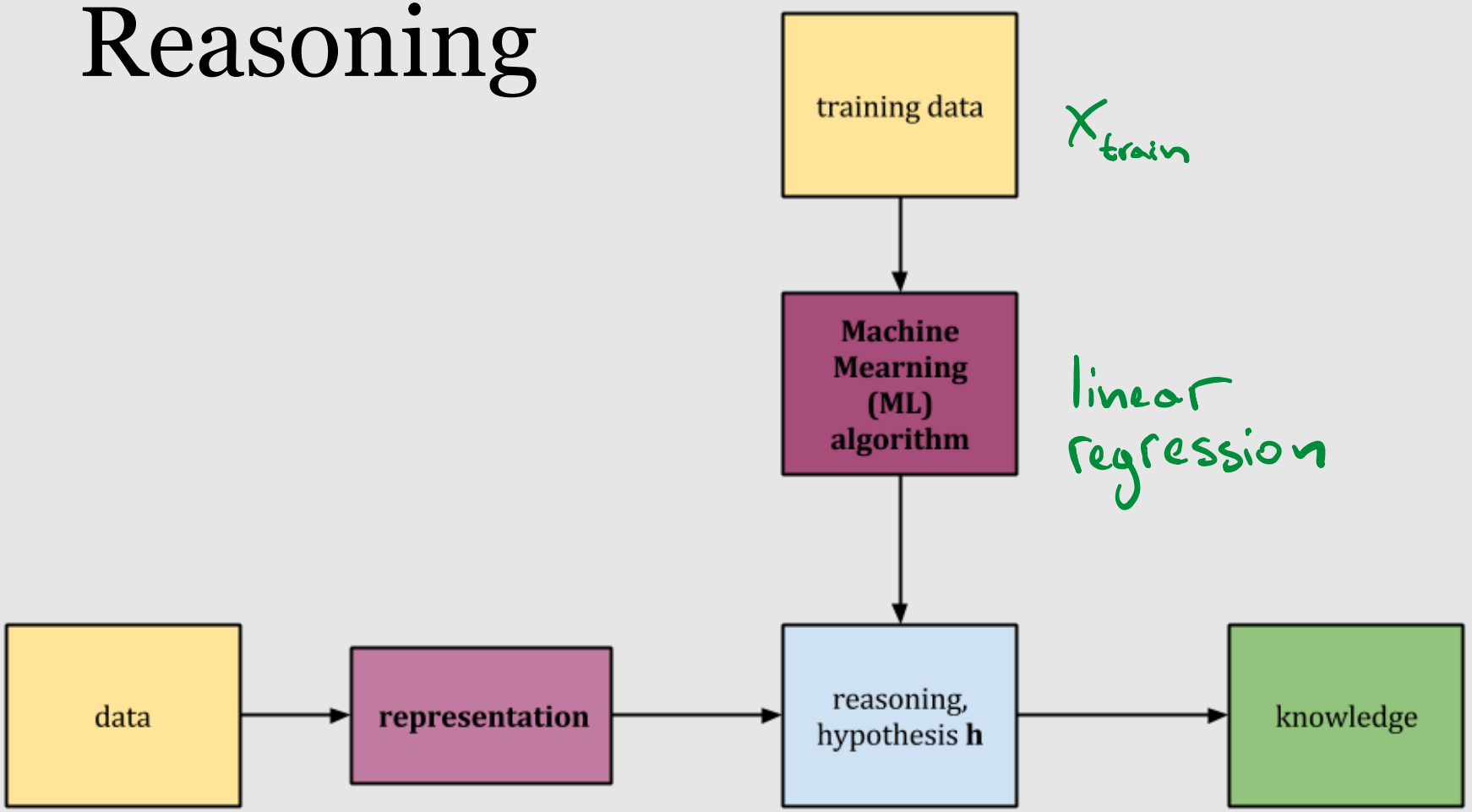
Statistics



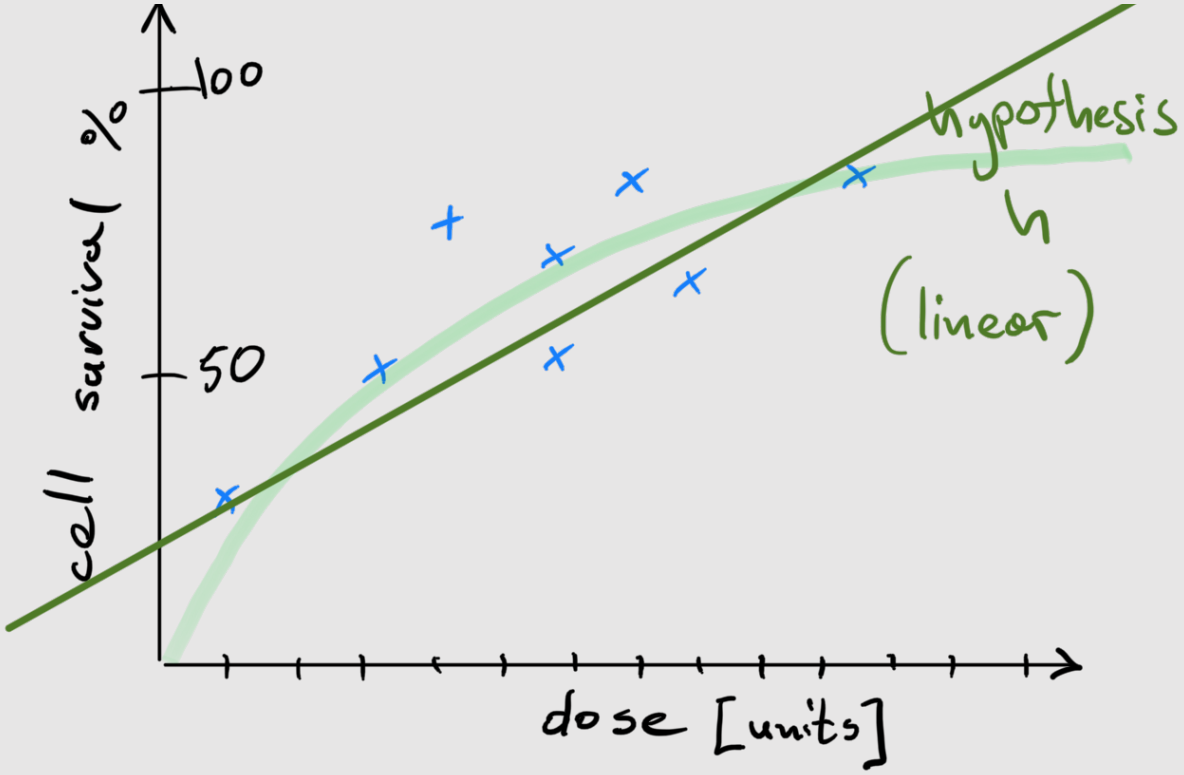
Machine learning



Reasoning


training data
X_{train}
Machine Mearning (ML) algorithm
linear regression
data
representation
reasoning, hypothesis h
knowledge
$X_? \rightarrow h_\theta(x) = \theta_0 + \theta_1 x \rightarrow y_{predicted}$

Linear regression

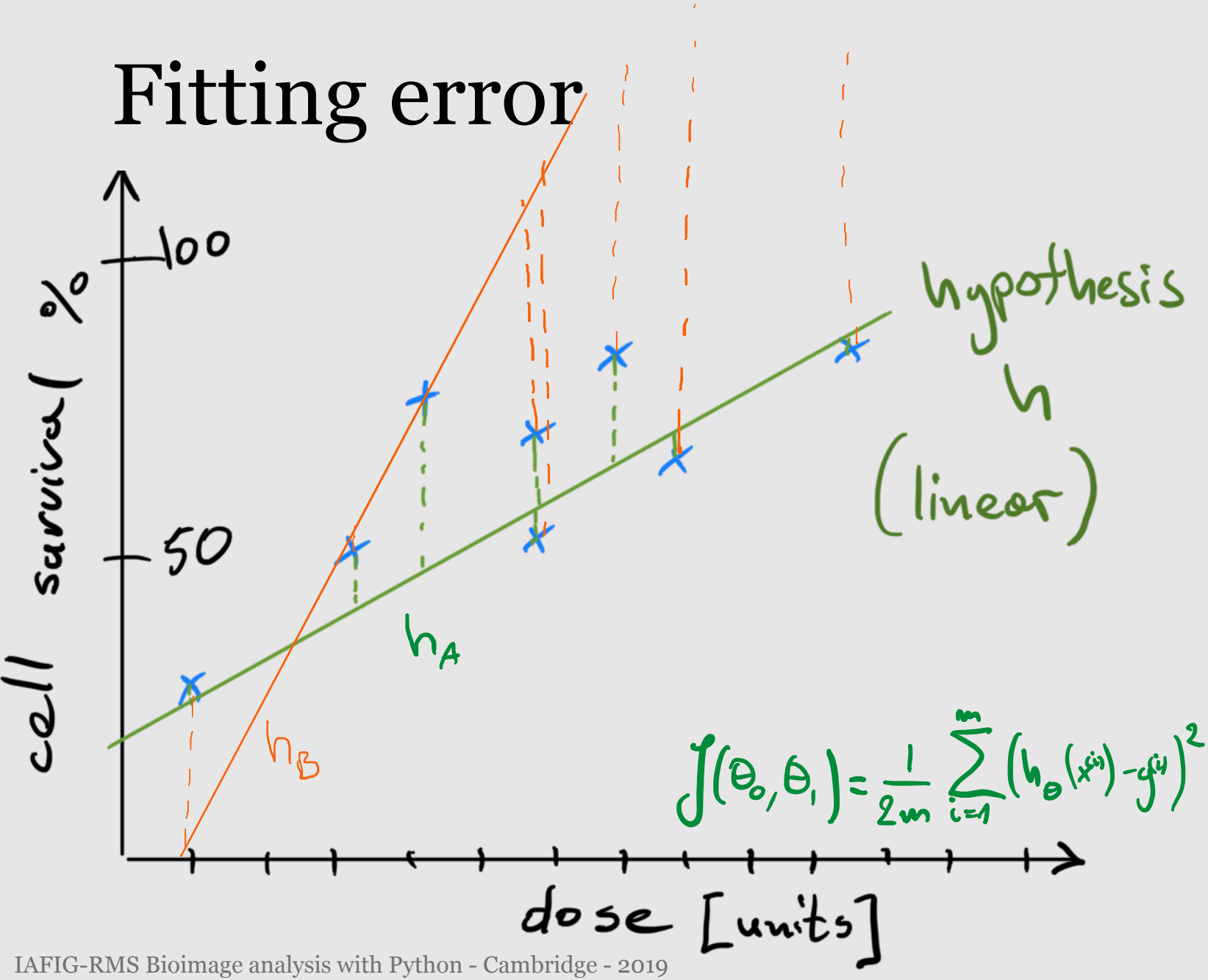


$$\frac{1}{2m}\sum_{i=0}^{m}\left(h_\theta(x_i)-y_i\right)^2$$

Squared error cost function.

We have m training examples (x_i, y_i)

Fitting error



Cost Function

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

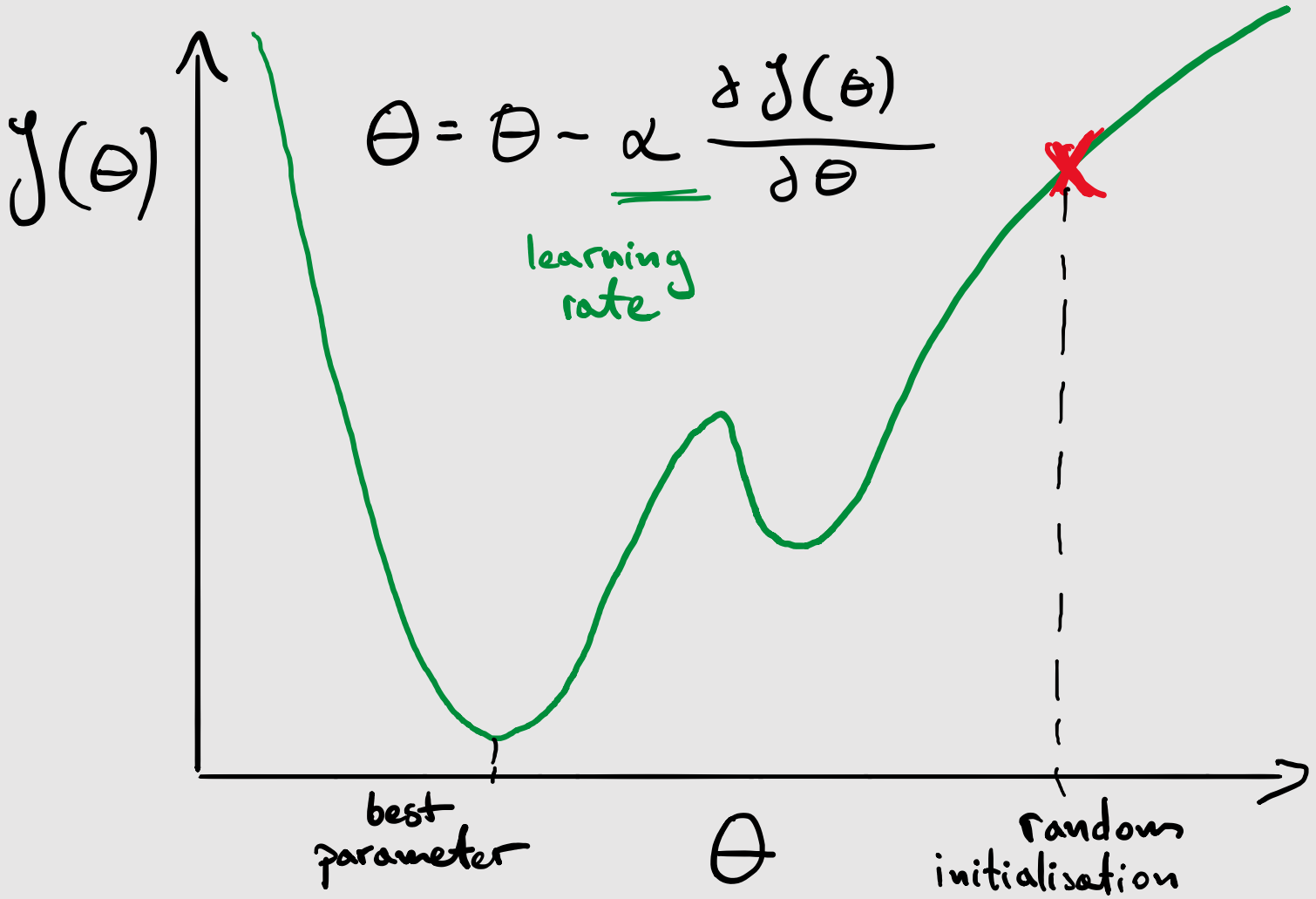
Parameters: θ_0, θ_1

Cost function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

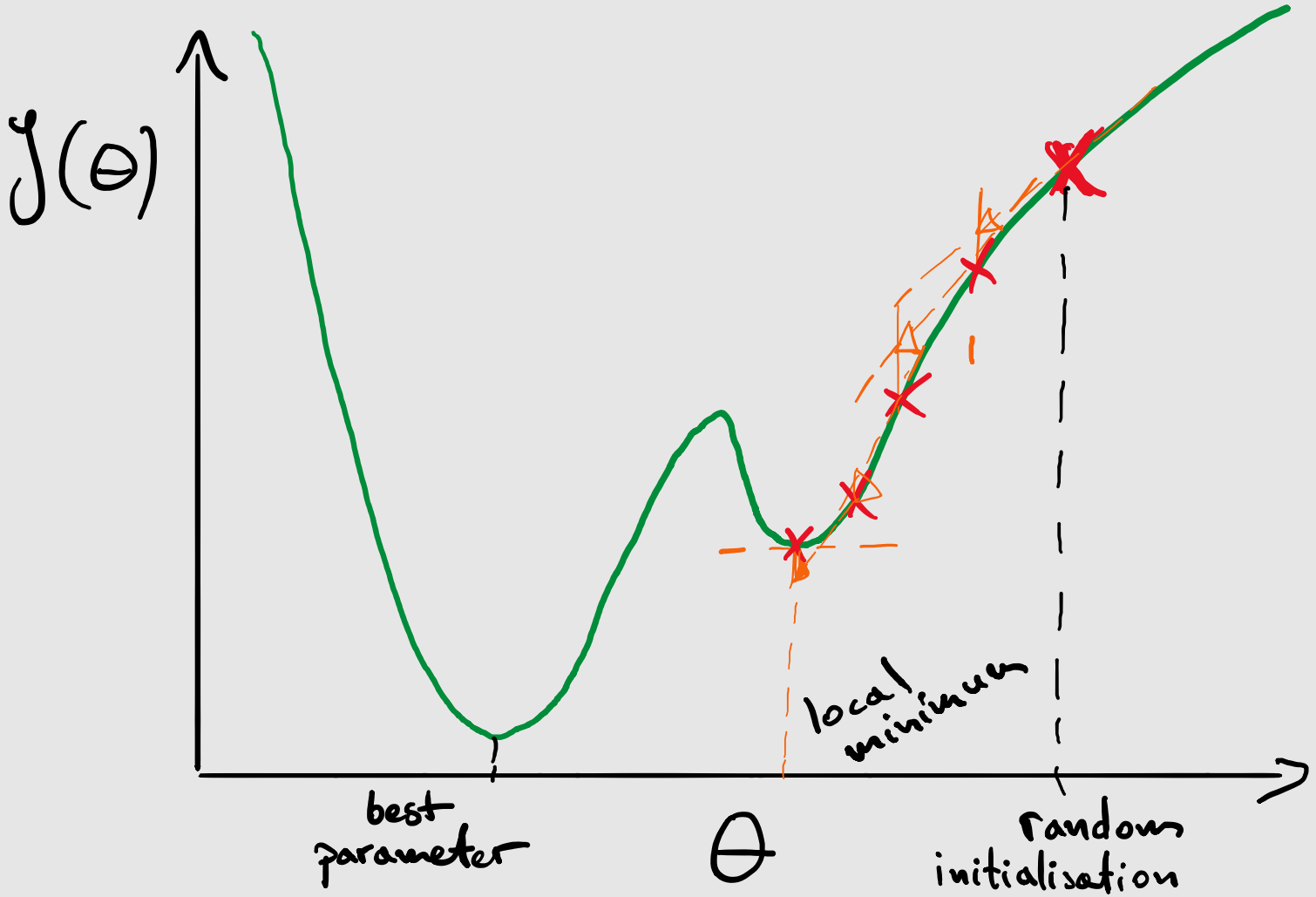
Goal: $\underset{\theta_0, \theta_1}{\operatorname{argmin}} J(\theta_0, \theta_1)$

aka *Loss function, Error function*

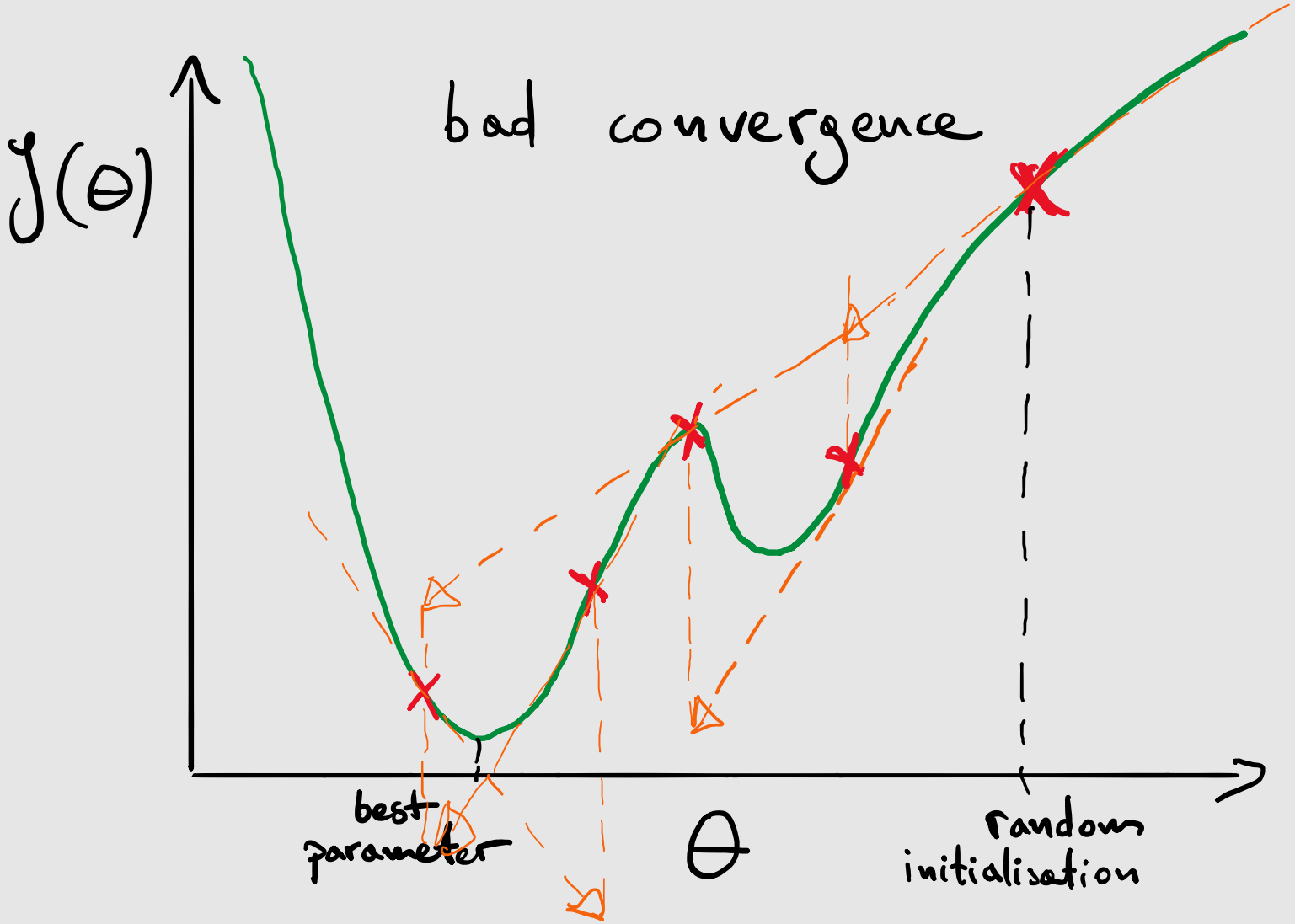
Gradient descent



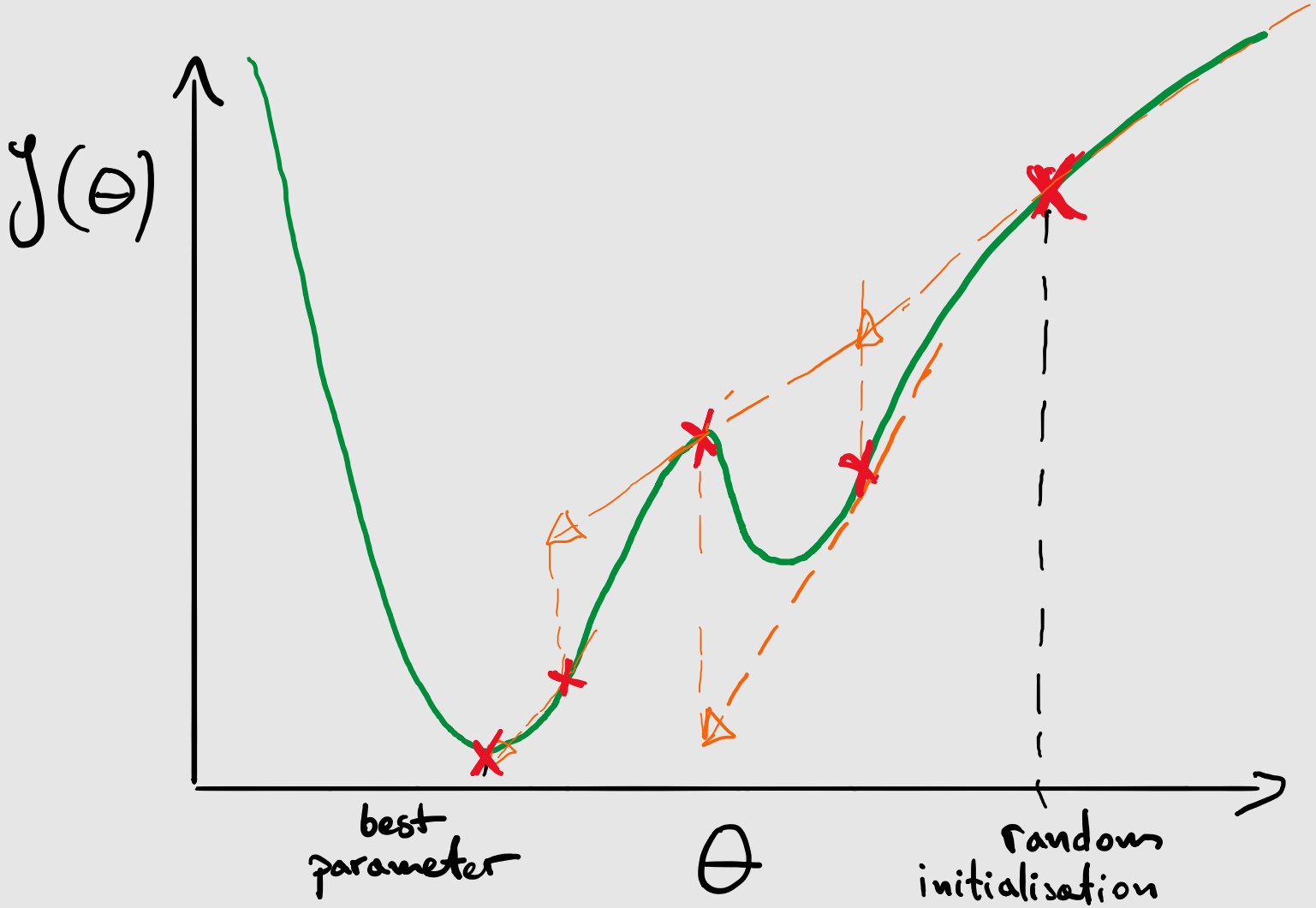
Gradient descent (small α)



Gradient descent (big α)



Gradient descent (adaptive α)



Gradient descent

- all parameters simultaneous
- batch: uses a "batch" of training examples

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \left(h_\theta(x) - y \right) x_j$$

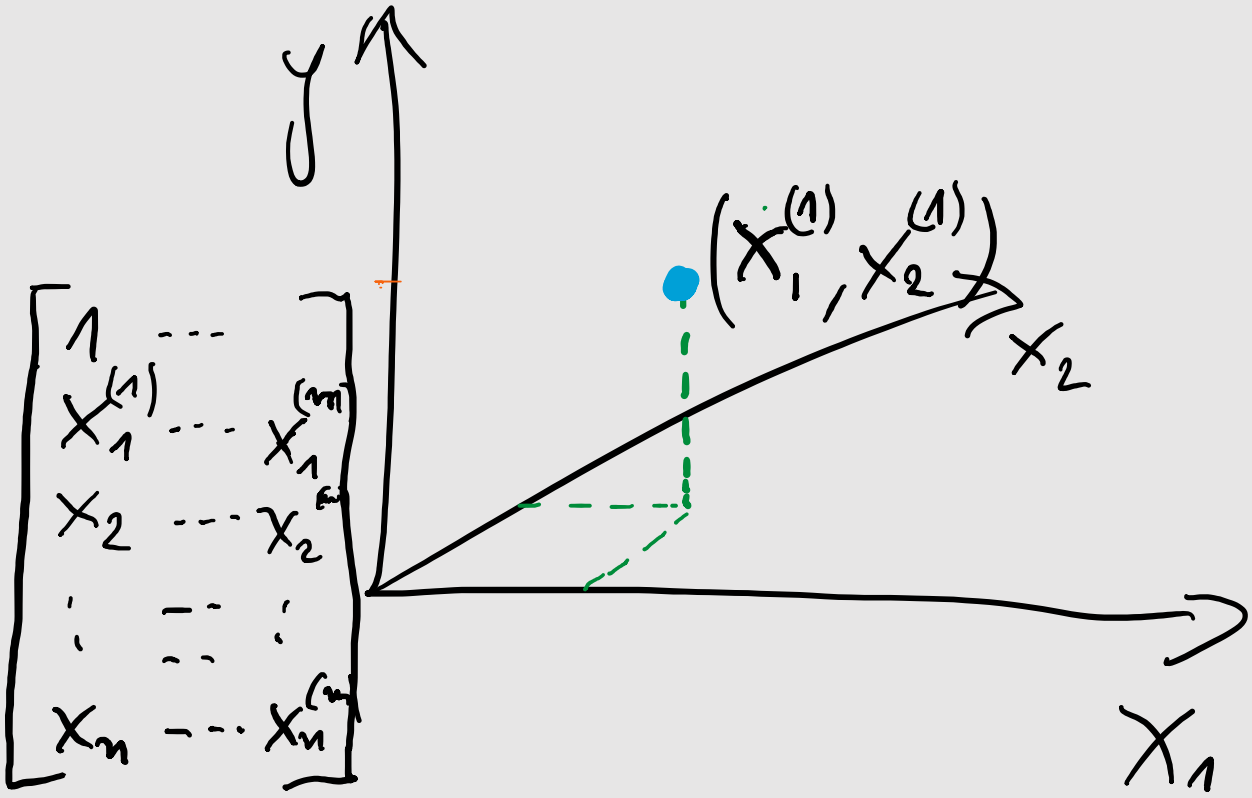
Multiple features

$x_0 = 1$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

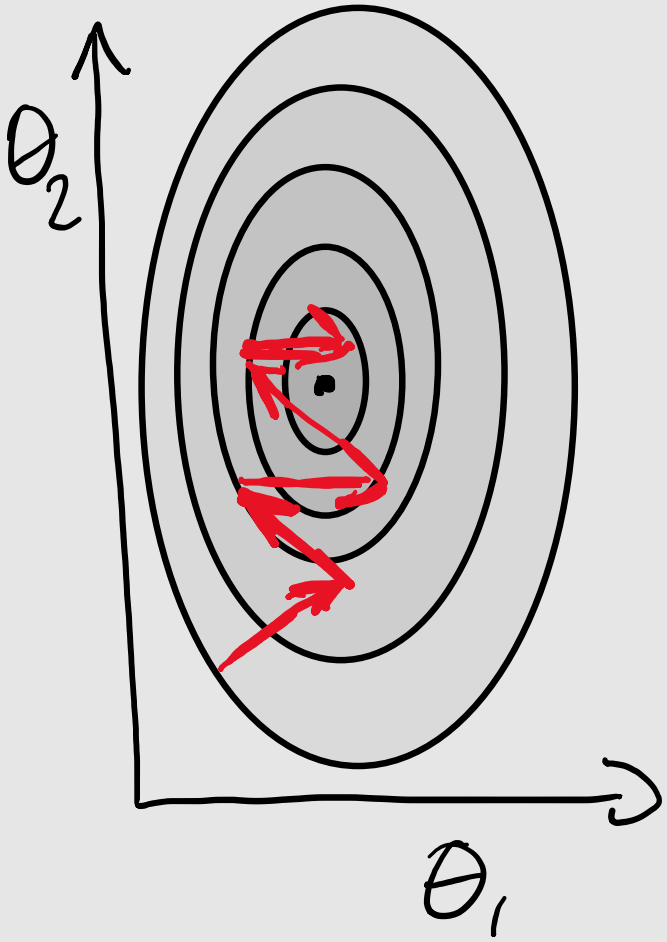
$$h_\theta(x) = \theta^T x$$

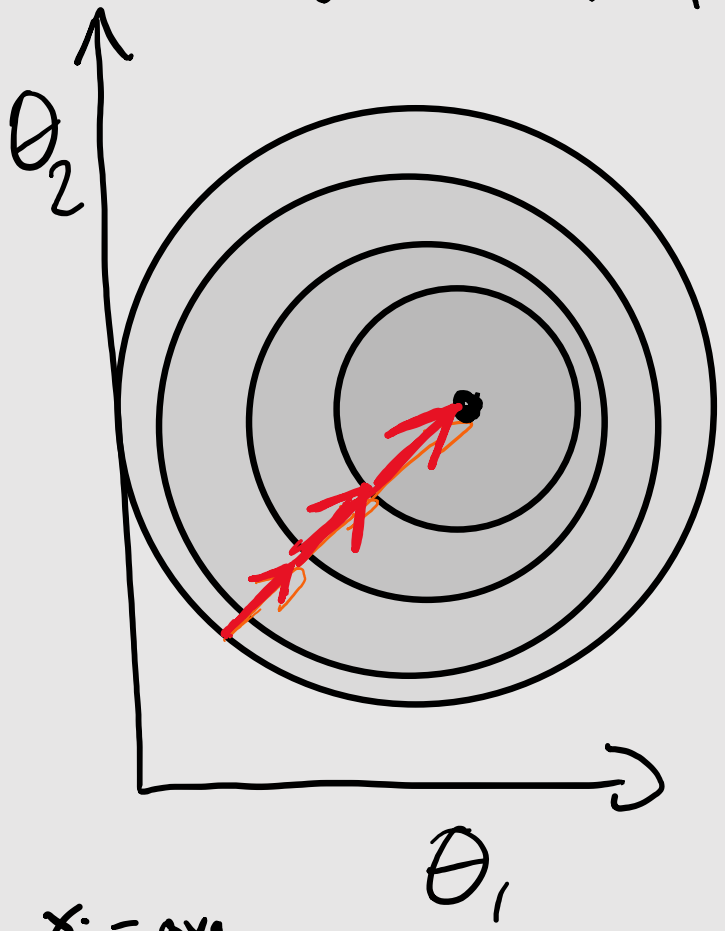
$$h_\theta(x) = [\theta_0 \theta_1 \cdots \theta_n] \begin{bmatrix} 1 & \cdots \\ x_1^{(1)} & \cdots & x_1^{(m)} \\ x_2 & \cdots & x_2^{(m)} \\ \vdots & \cdots & \vdots \\ x_n & \cdots & x_n^{(m)} \end{bmatrix}$$



Multiple features

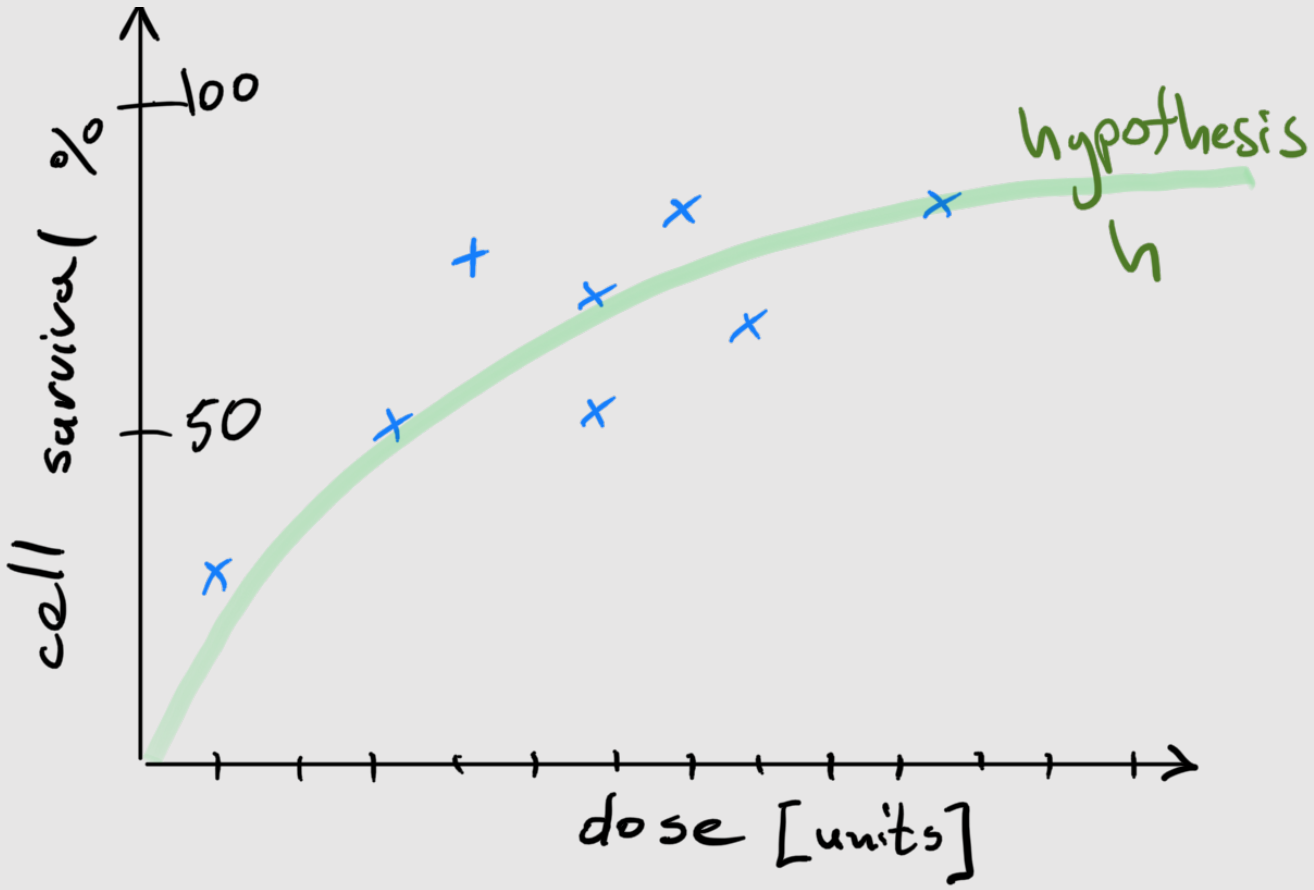
α - learning rate doesn't like un-scaled features





$$x_{scaled} = \frac{x_i - avg}{max - min}$$

Non-linear regression



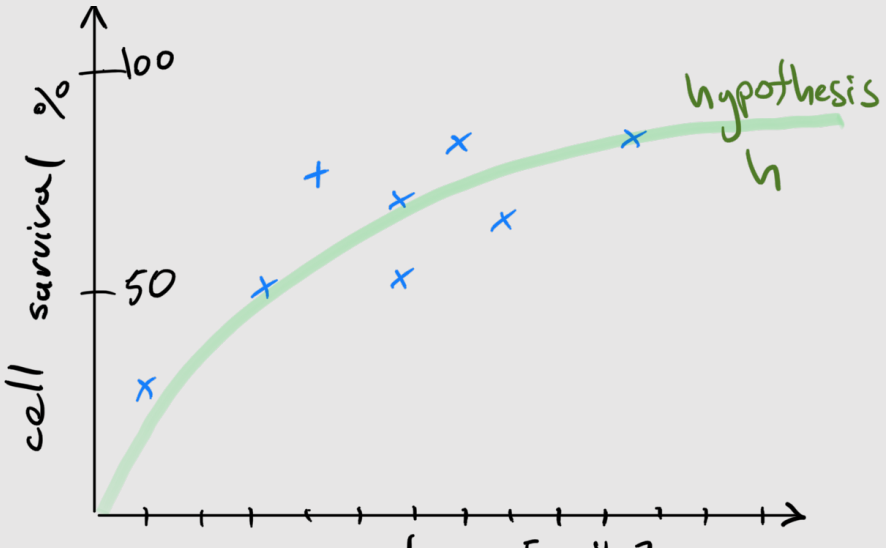
Multiple non-linear features

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_n x^n$ } polynomial

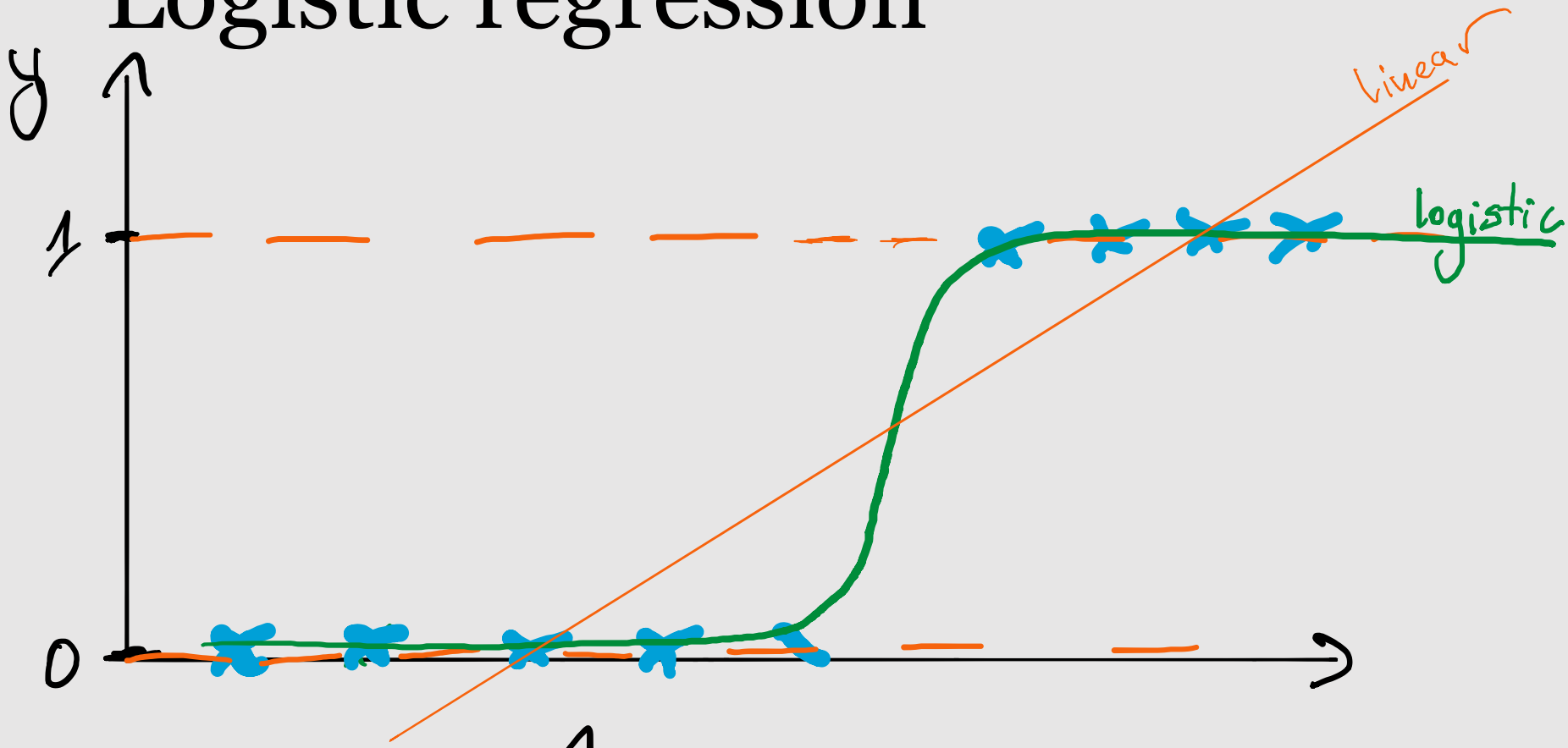
$\Downarrow$

$h_\theta(x) = \theta_0 + \theta_1 \sqrt{x} + \theta_2 \log(x) + \ldots$ } non-linear (generalised)

$J(\theta)$ → stays the same.



Logistic regression



$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$J(\theta) = ?$$

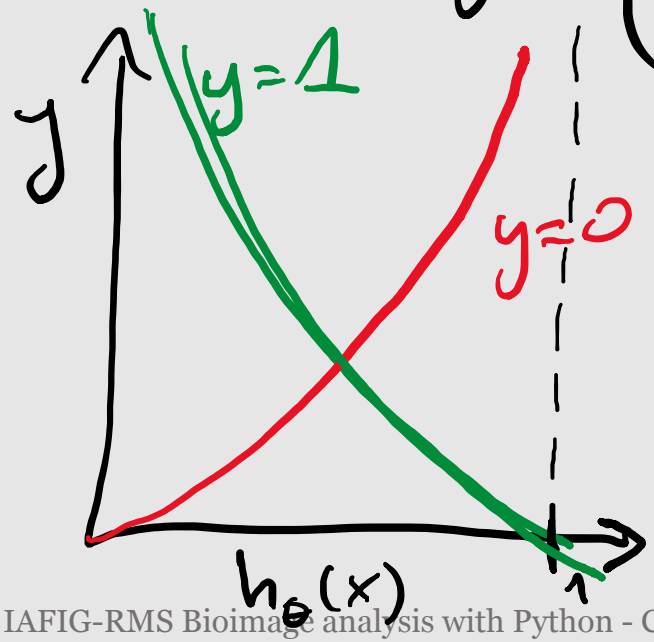
Logistic regression

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

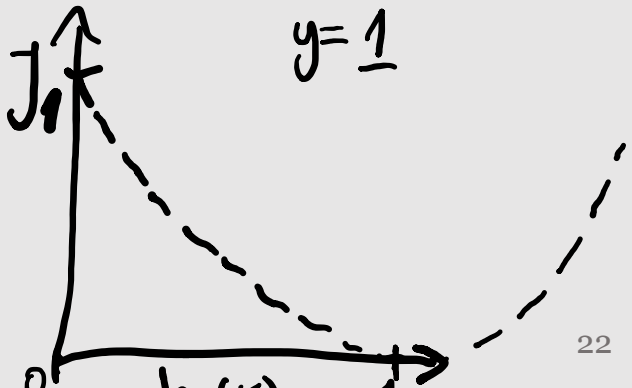
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^i), y)$$

normal sum of squares makes J non-convex

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & : y = 1 \\ -\log(1 - h_\theta(x)) & : y = 0 \end{cases}$$



Squared error



Scikit-learn library

scikit-learn

Machine Learning in Python

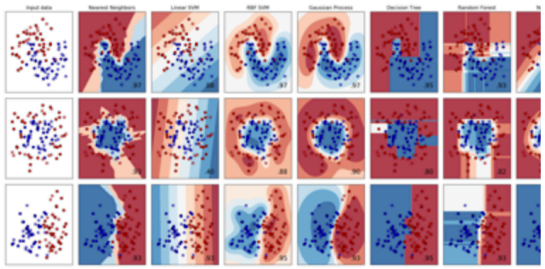
Getting Started | What's New in 0.22 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

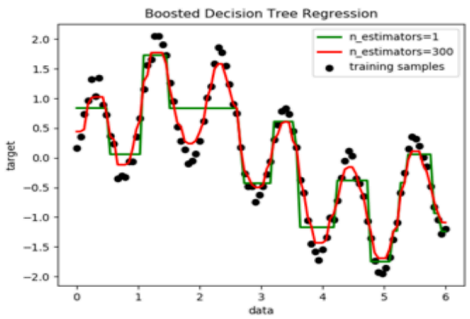


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

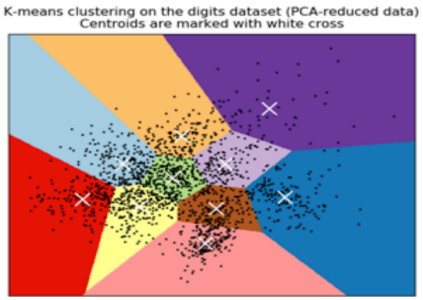


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...



Examples