Course 2

Introduction to Machine Learning

Christophe Eloy

Outline

- Practical information
- What is machine learning?
- Different types of learning
- Examples
- Linear regression with one variable

Practical information

- Alternance of classes and tutorials
- Tutorials with Python/Jupyter + standard libraries (numpy, pandas, etc.)
- Assessment with homework + project
- Contact me: christophe.eloy@univ-amu.fr

Useful resources

- MOOC Machine Learning, Andrew Ng (Stanford)
- Review paper “A high-bias, low-variance introduction to Machine Learning for physicists”, Pankaj Mehta et al. with Jupyter notebooks
- Lectures on Learning from Data, Yaser Abu-Mostafa (Caltech) on iTunesU and YouTube

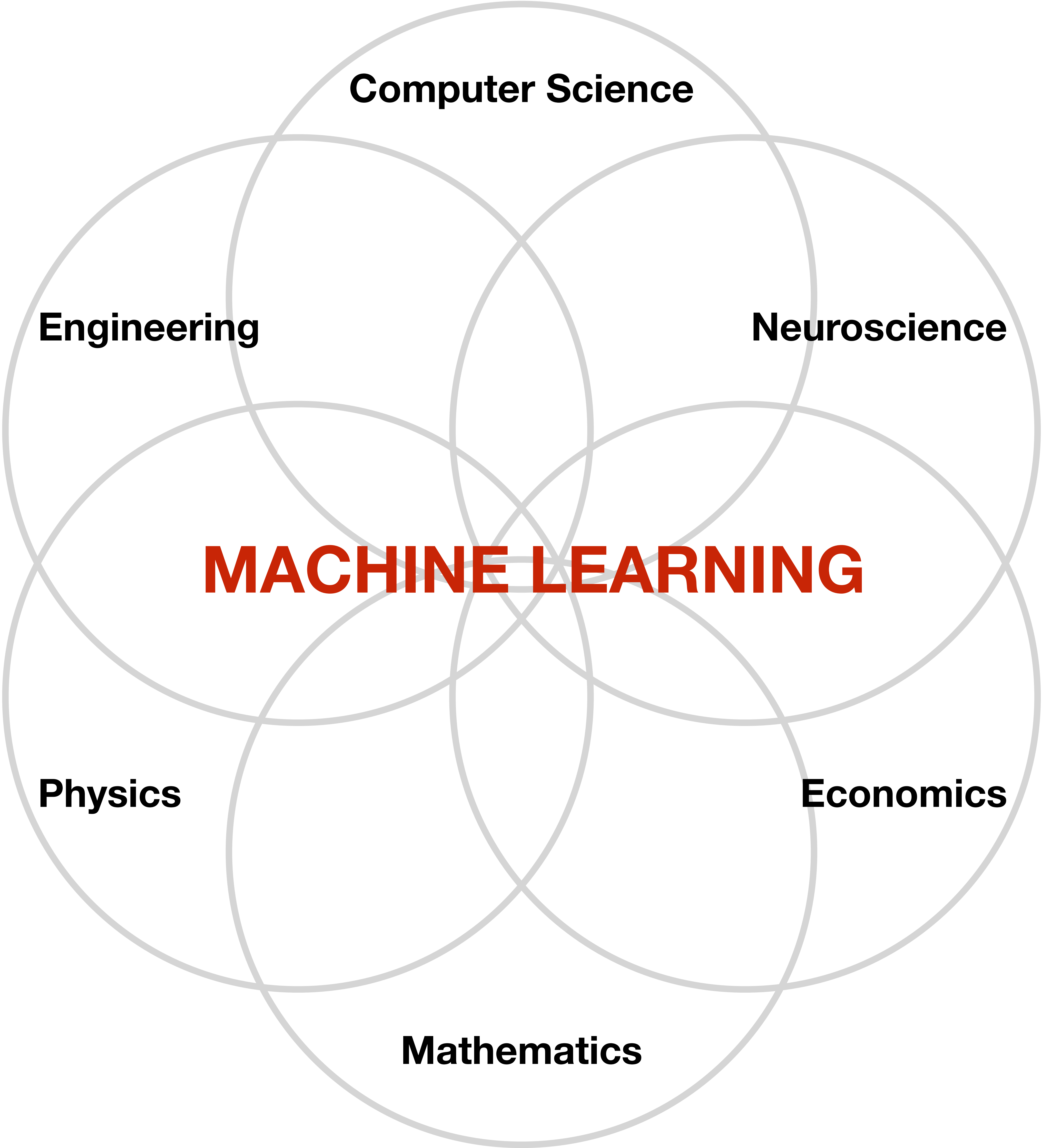
What is Machine Learning?

- Two definitions

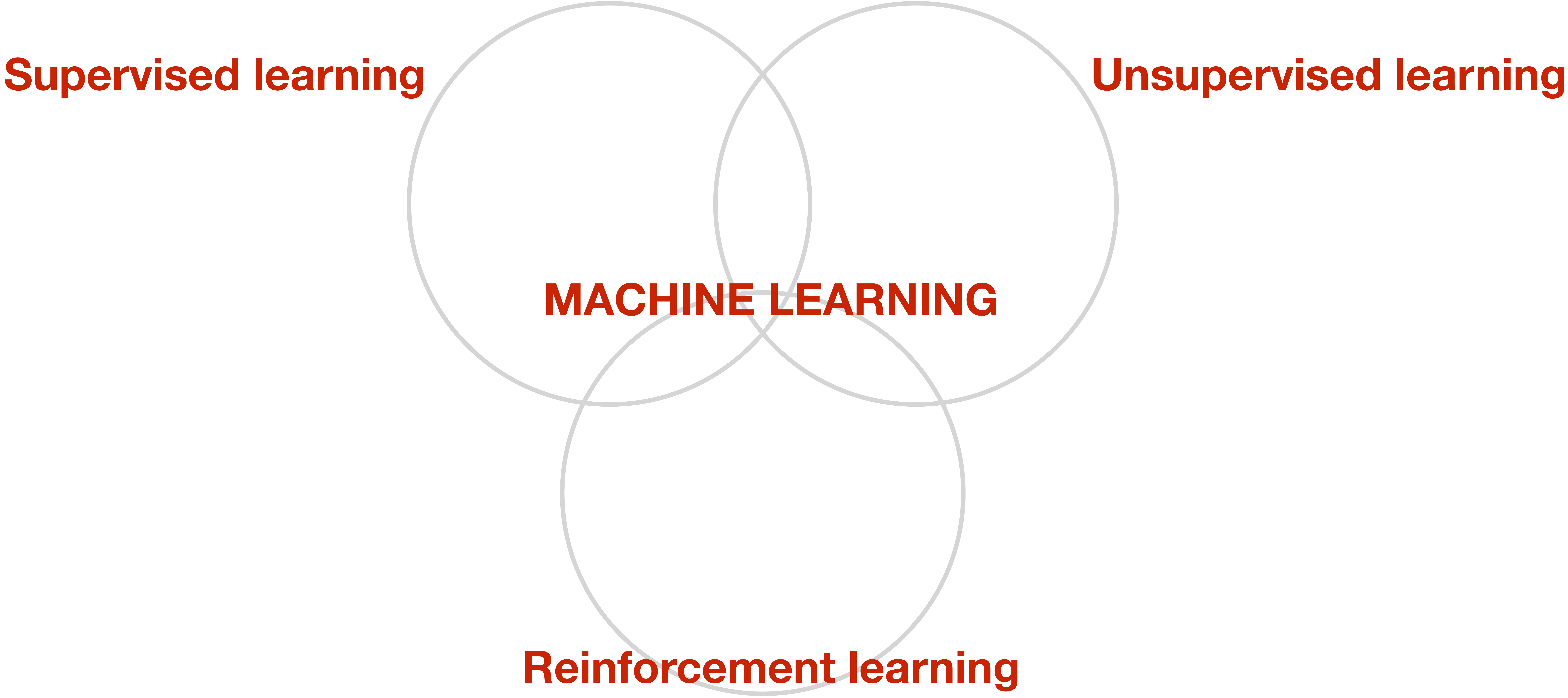
“The field of study that gives computers the ability to learn without being explicitly programmed.”
—Arthur Samuel

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”
—Tom Mitchell

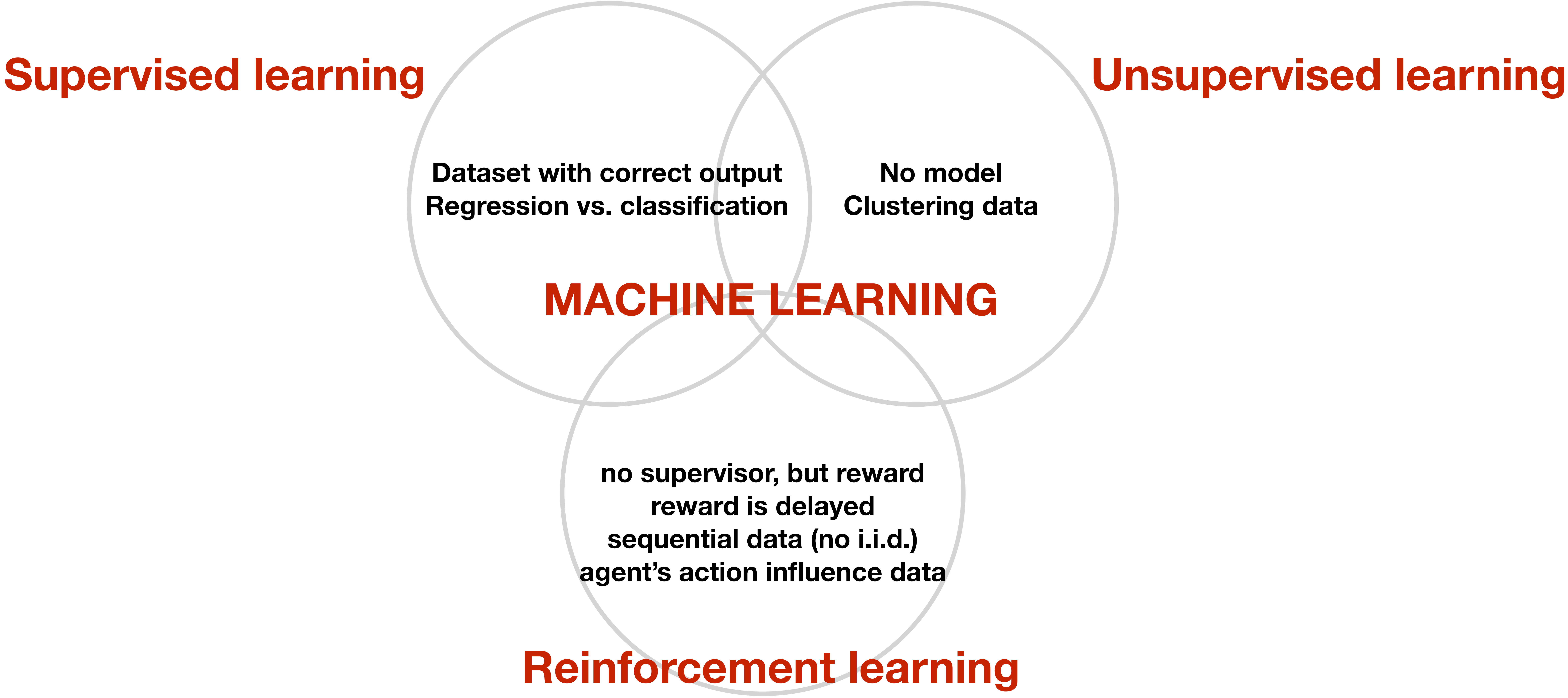
At the crossroad of many fields



Branches of Machine Learning



Branches of Machine Learning



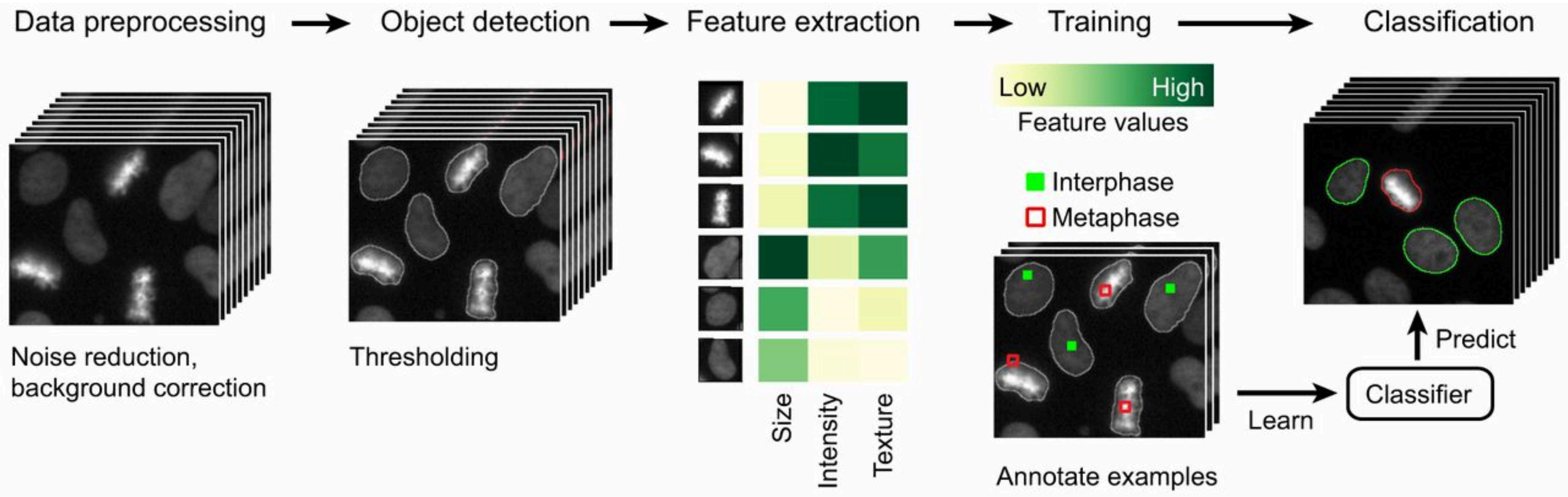
Examples

- Classify emails as spam or not
- Predict the success of a movie
- Diagnose from a list of symptoms
- Drive a car autonomously
- Translate a text
- Find groups in a social network
- Defeat the world champion of Go

Examples

• Classify emails as spam or not	Classification (supervised learning)
• Predict the success of a movie	Regression (supervised learning)
• Diagnose from a list of symptoms	Classification (supervised learning)
• Drive a car autonomously	Reinforcement learning
• Translate a text	Reinforcement learning
• Find groups in a social network	Unsupervised learning
• Defeat the world champion of Go	Reinforcement learning

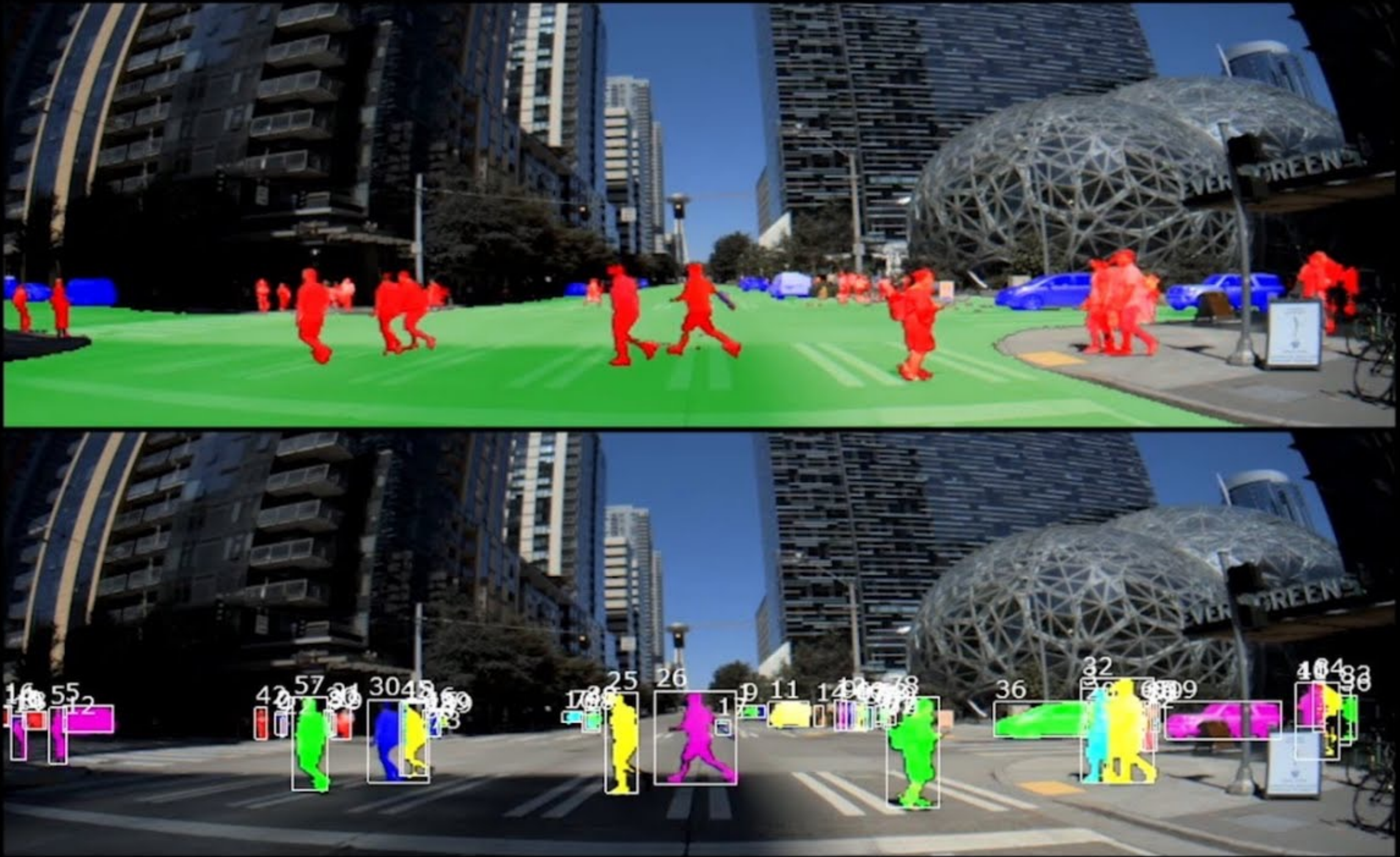
Recognize and classify phenotypes



Sommer & Gerlich, *J. Cell Sci*. (2013) DOI: 10.1242/jcs.123604

Learning to exploit vortices

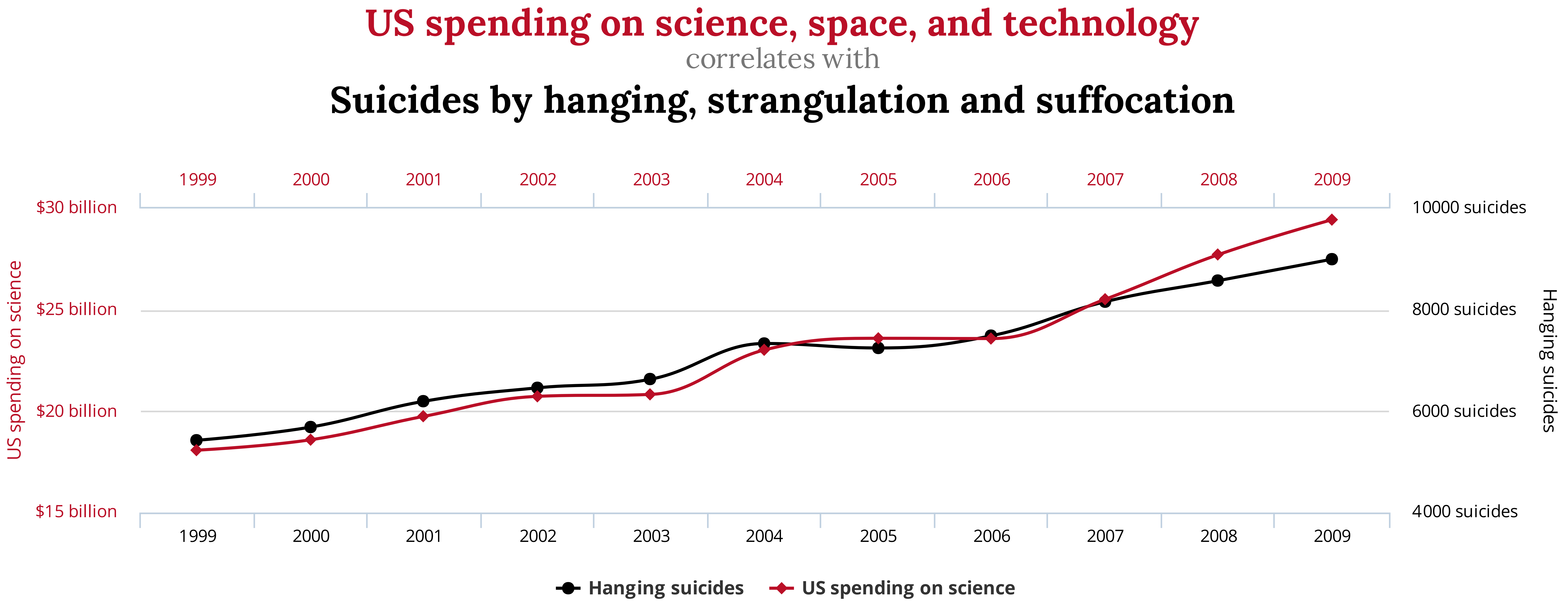
Verma, Novati & Koumoutsakos, *Proc. Nat. Acad. Sci.* (2018). DOI: 10.1073/pnas.1800923115



https://youtu.be/HS1wV9NMLr8

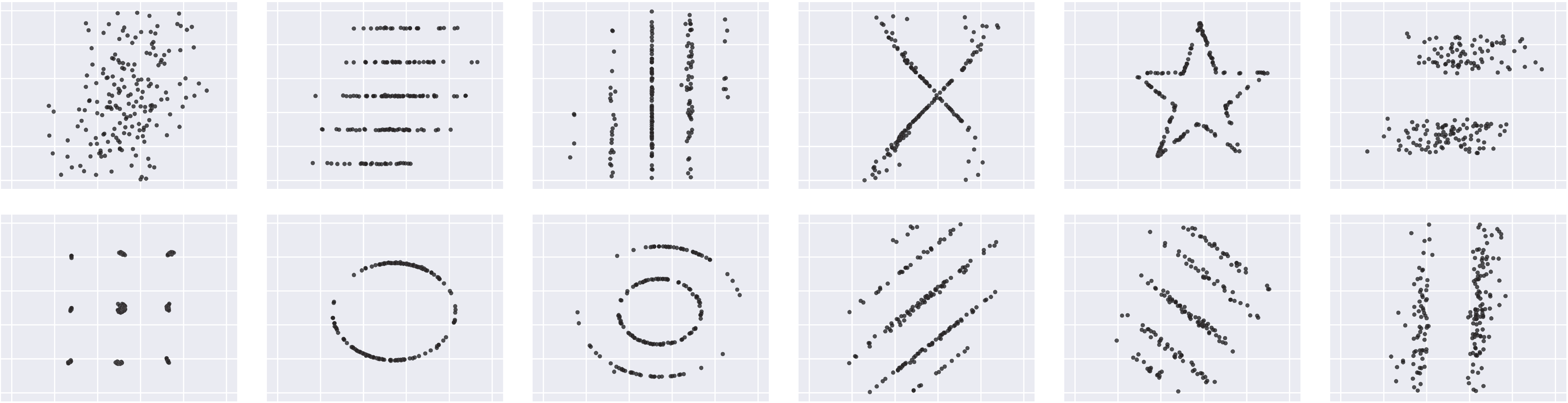
https://youtu.be/V1eYniJ0Rnk

Beware of spurious correlations!



tylervigen.com

Plot your data!



All data have same means, same std, same Pearson’s correlation coefficient r

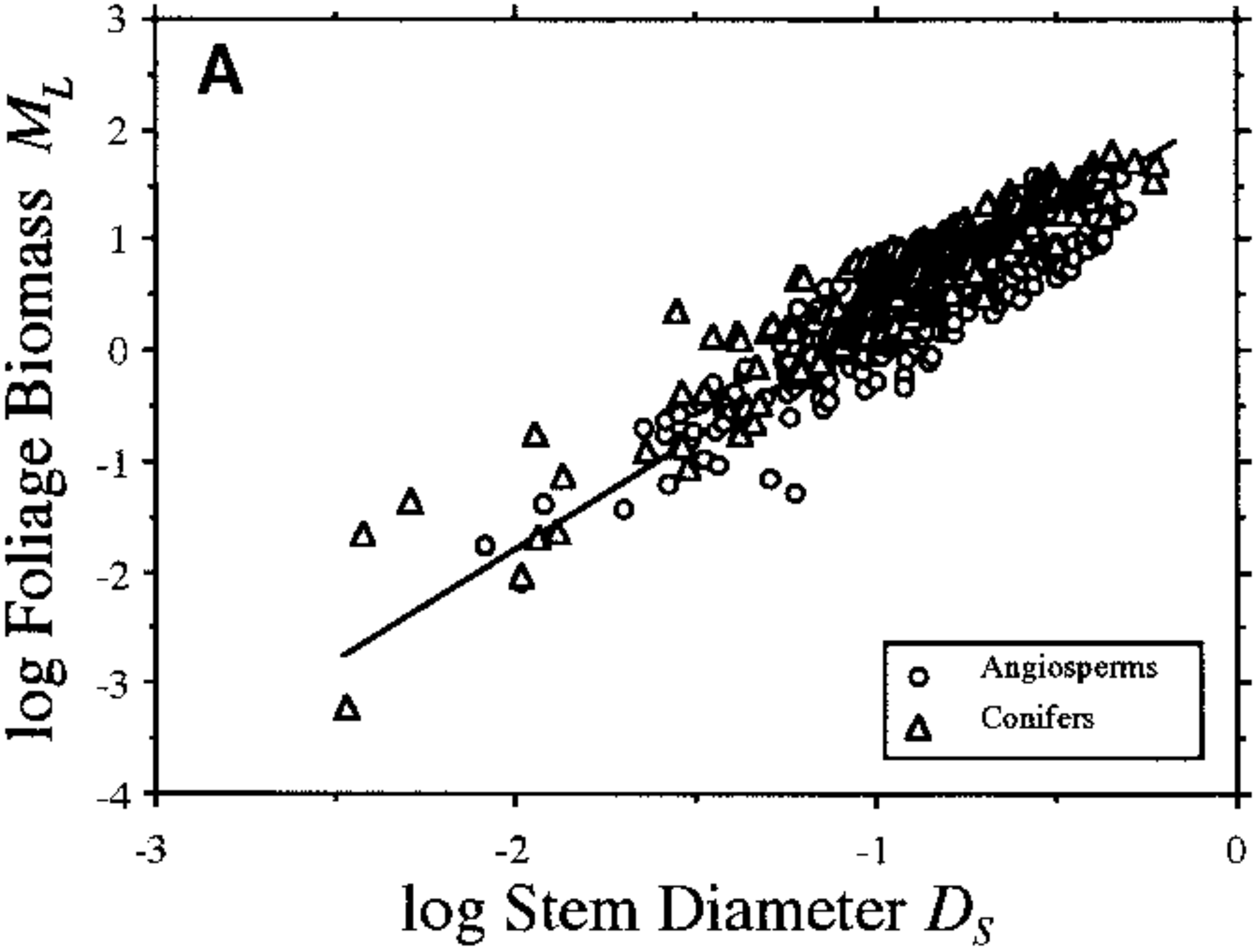
Matejka & Fitzmaurice (2017) Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems

When to use Machine Learning?

- There is a “pattern”
- Relationships are not tractable mathematically
- There is (a lot of) data

Univariate linear regression

Simple working example



Enquist & Niklas, *Science* (2002) DOI: 10.1126/science.1066360

Formalization of learning

- Input: $\mathbf{x}$ (trunk diameter)
- Output: $\mathbf{y}$ (leaf mass)
- Target function: $f : \mathscr{X} \rightarrow \mathscr{Y}$ (the relationship we are looking for)
- Data: $\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right) \cdots \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right)$ (to be split into training and testing data)
- Hypothesis: $h_\theta : \mathscr{X} \rightarrow \mathscr{Y}$ (the set of functions parametrized by $\boldsymbol{\theta}$)

Learning components

unknown target function

$f : \mathscr{X} \to \mathscr{Y}$

training set

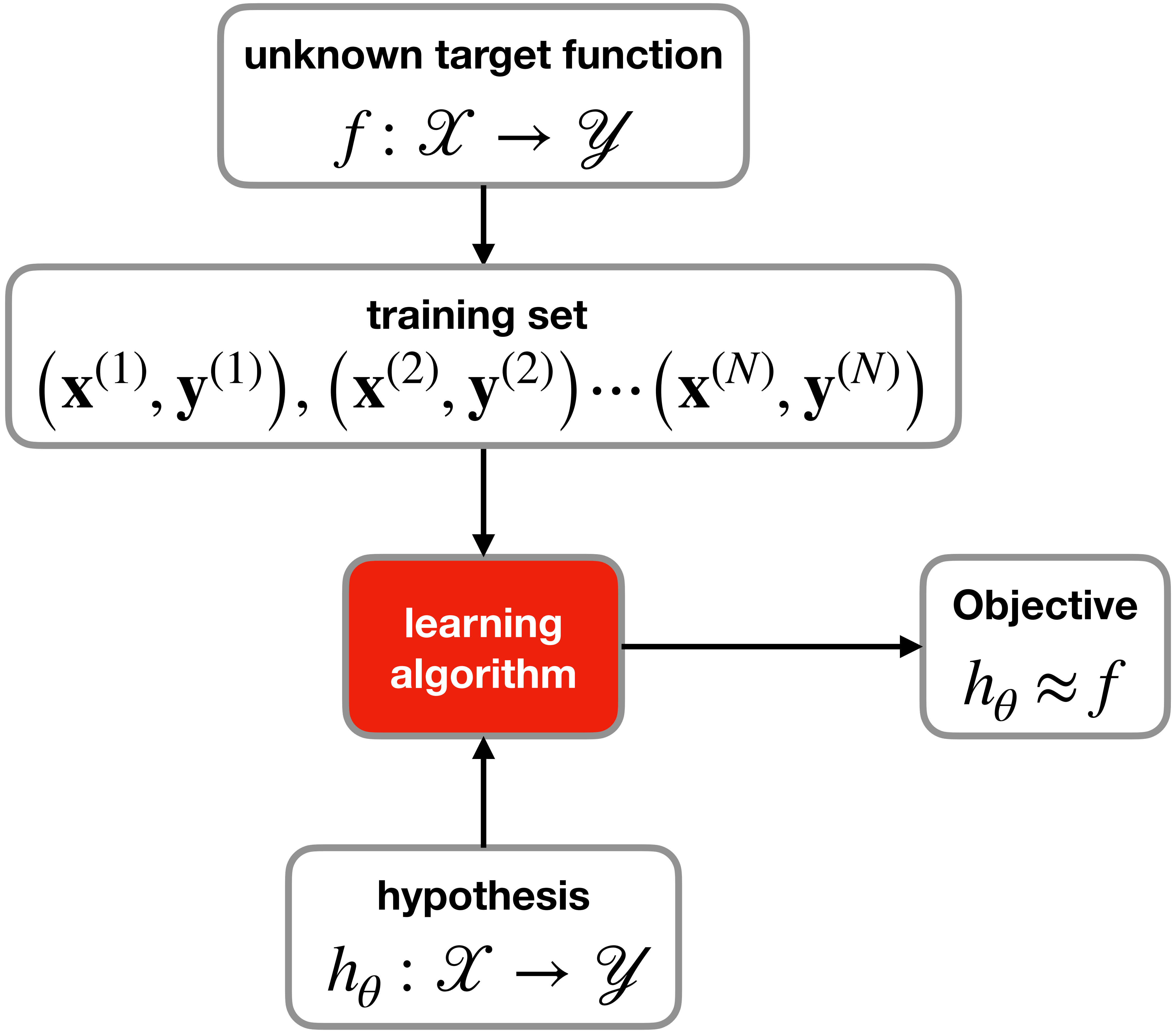
$$\left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \left(\mathbf{x}^{(2)}, \mathbf{y}^{(2)}\right) \cdots \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right)$$

Supervised learning

Objective

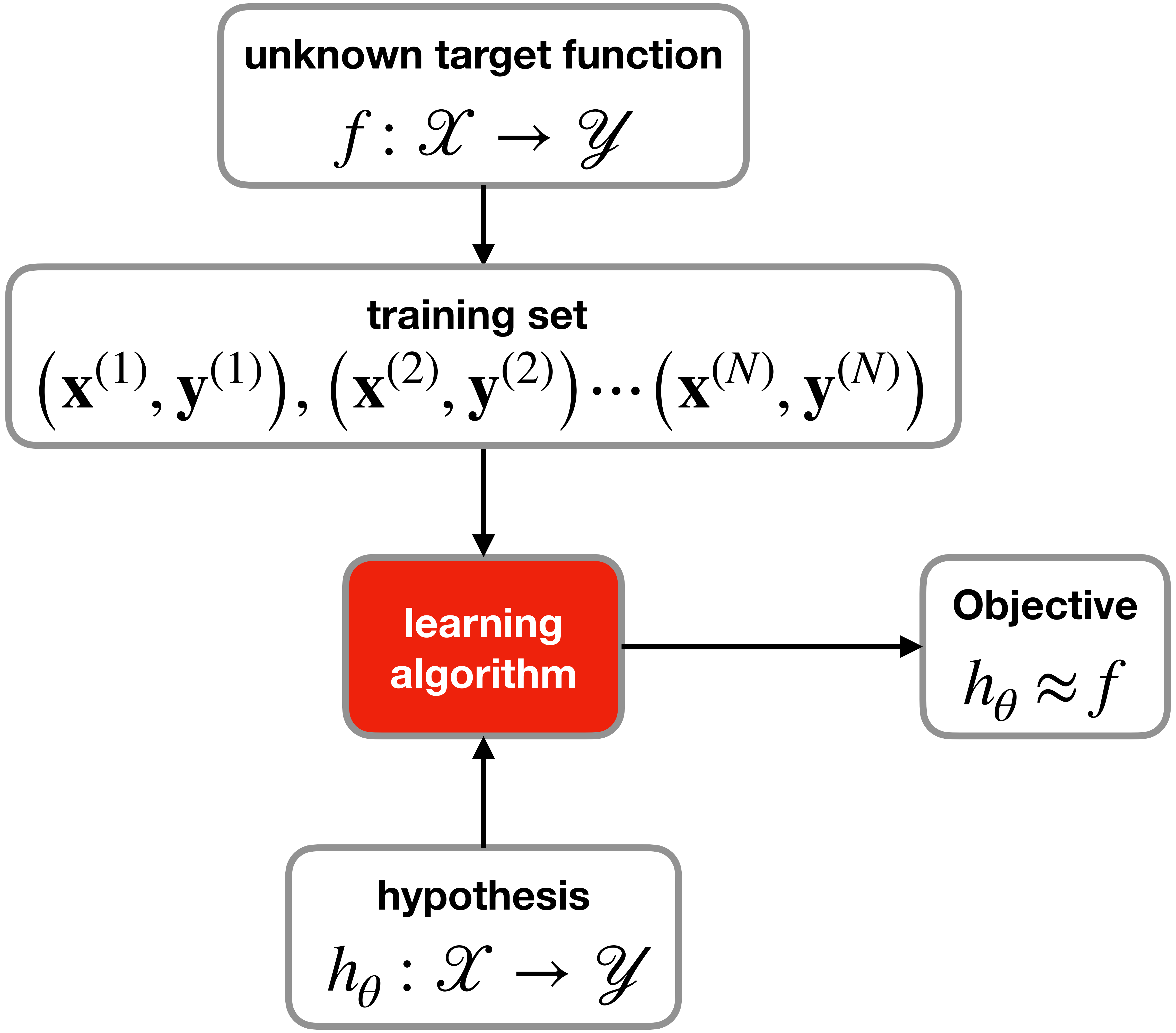
$h_\theta \approx f$

Learning components



Learning components

- Two components of the learning problem
  - Hypothesis: $h_\theta \in \mathscr{H}$
  - Learning algorithm
    - Iterative procedure
    - Cost function
- These two components form the **learning model**



Univariate linear regression

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
- Parameters: $\theta = [\theta_0, \theta_1]^T$
- Cost function (least squares): $J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$
- Goal: find $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Gradient descent algorithm

- Iterative procedure

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\theta := \theta - \alpha \nabla_\theta J(\theta)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

- Learning parameter: α

- Full batch vs. stochastic vs. mini-batch gradient descent