BioComp Summer School 2017

Introduction to Machine Learning

Chloé-Agathe Azencott
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Inserm
Institut national
de la santé et de la recherche médicale

institutCurie

PSL
RESEARCH UNIVERSITY PARIS

MINES
ParisTech

Keywords

machine learning

deep learning

likelihood

kernels

ensemble learning

gradient descent

generalization

model selection

overfitting

cross-validation

regularization

Overview

- What kind of **problems** can machine learning solve?
- Some popular **supervised** ML **algorithms:**
  - **Linear** models
  - **Support vector machines**
  - **Random forests**
  - **(Deep) neural networks.**
- How do we **select** a machine learning algorithm?
- What is **overfitting**, and how can we avoid it?

What is (Machine) Learning?

Why Learn?

- **Learning**:

  Modifying a behavior based on experience

  (F. Benureau)

- **Machine learning**: Programming computers to
  - **model data**
  - by means of **optimizing** an **objective function**
  - using **example** data.

Why Learn?

- There is no need to "learn" to calculate payroll.
- Learning is used when
  - Human expertise does not exist (bioinformatics);
  - Humans are unable to explain their expertise (speech recognition, computer vision);
  - Solutions change in time (routing computer networks);
  - Solutions need adapting to new cases (user biometrics).

What about AI?

Artificial Intelligence

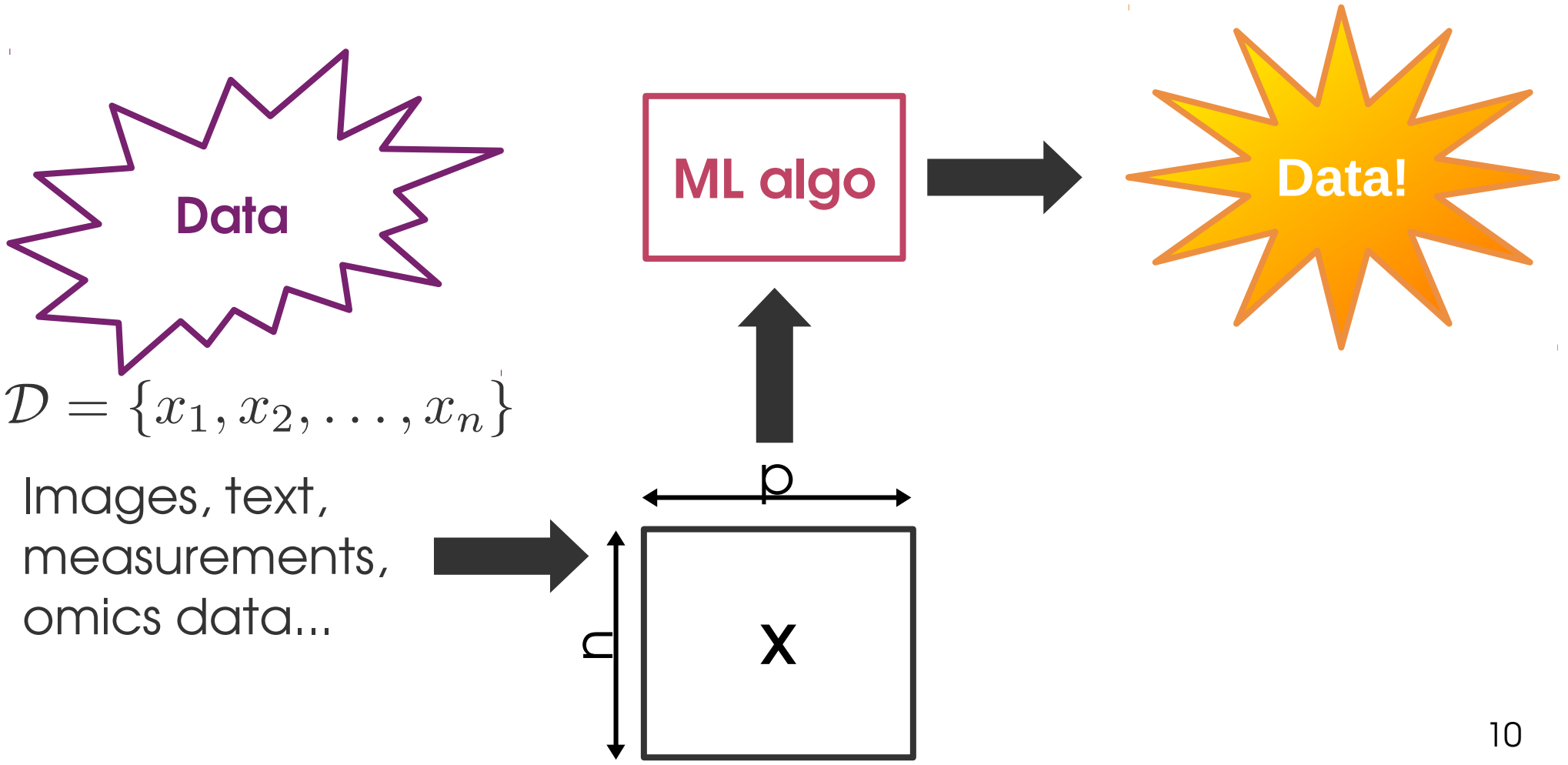
ML is a subfield of **Artificial Intelligence**

- A system that lives in a **changing environment** must have the ability to **learn** in order to **adapt.**
- ML algorithms are building blocks that make computers behave more intelligently by **generalizing** rather than merely storing and retrieving data (like a database system would do).

Zoo of ML Problems

Unsupervised learning

Learn a **new representation** of the data



Clustering

Group **similar** data points together

$\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$

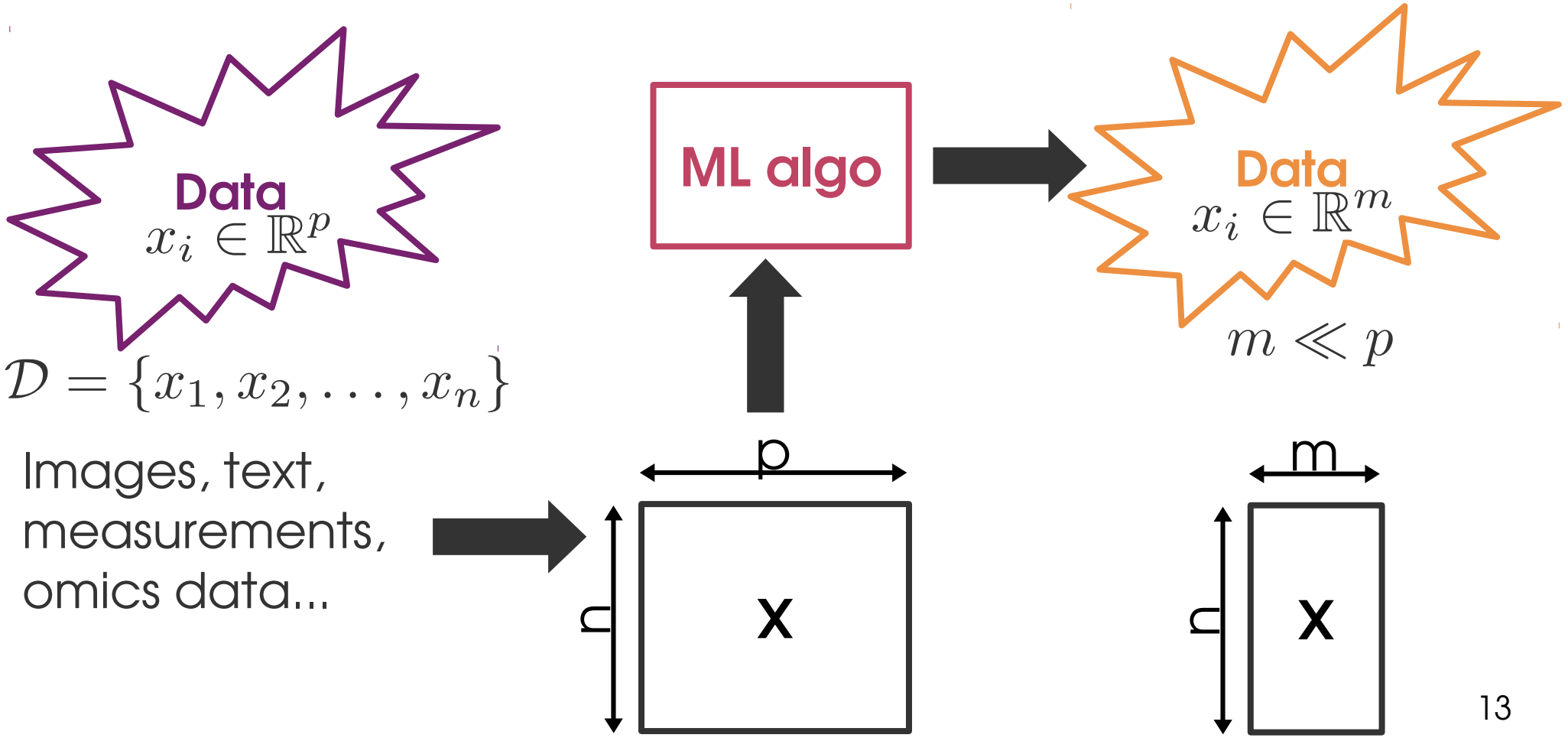
- Understand **general characteristics** of the data;
- **Infer some properties** of an object based on how it relates to other objects.

Clustering: applications

- **Customer relationship management:** Customer segmentation
- **Image compression:** Color quantization
- **Document clustering:** Group documents by topics (bag-of-words)
- **Bioinformatics:** Learning motifs.

Dimensionality reduction

Find a **lower-dimensional** representation



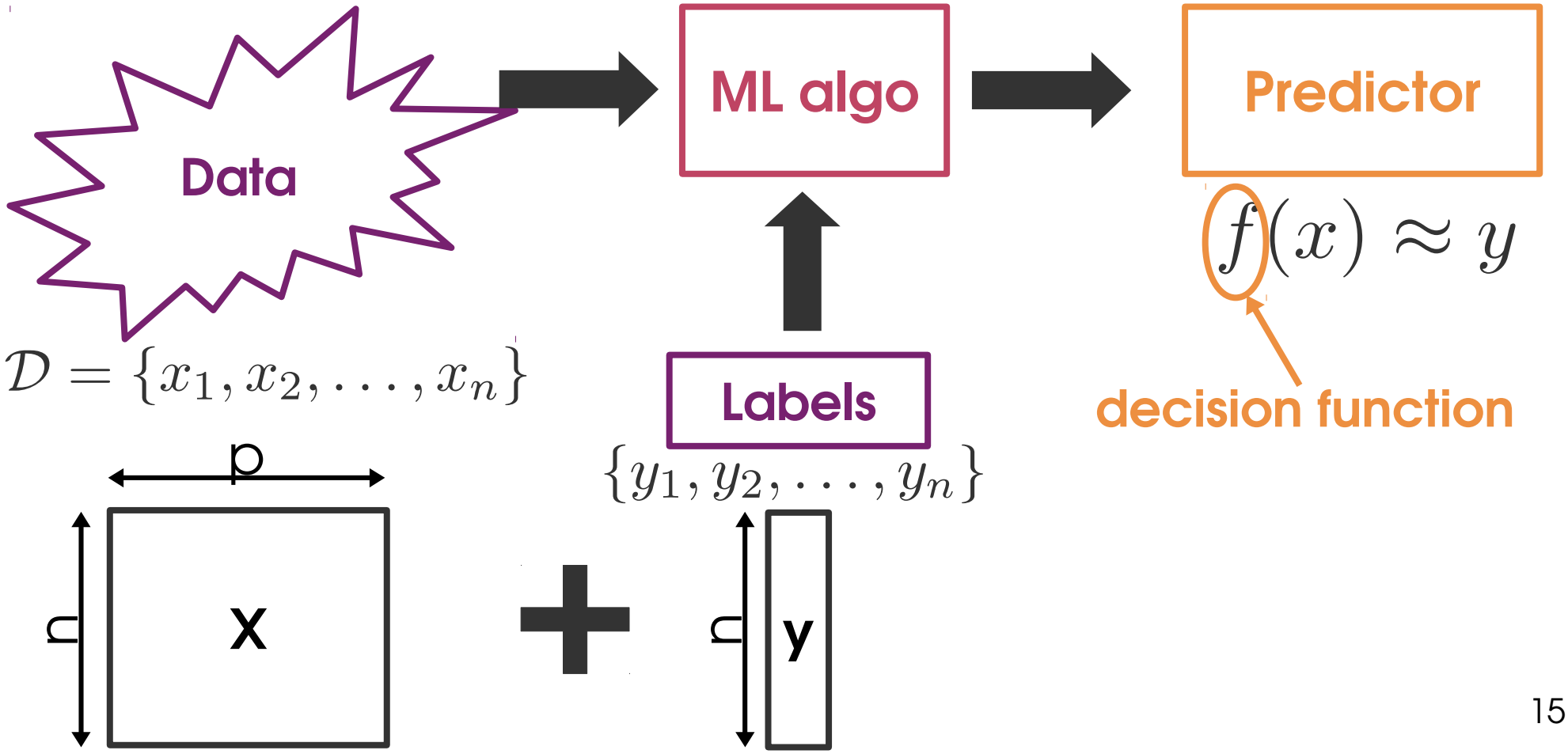
Dimensionality reduction

Find a **lower-dimensional** representation

- Reduce storage **space** & computational **time**
- Remove **redundances**
- **Visualization** (in 2 or 3 dimensions) and **interpretability**.

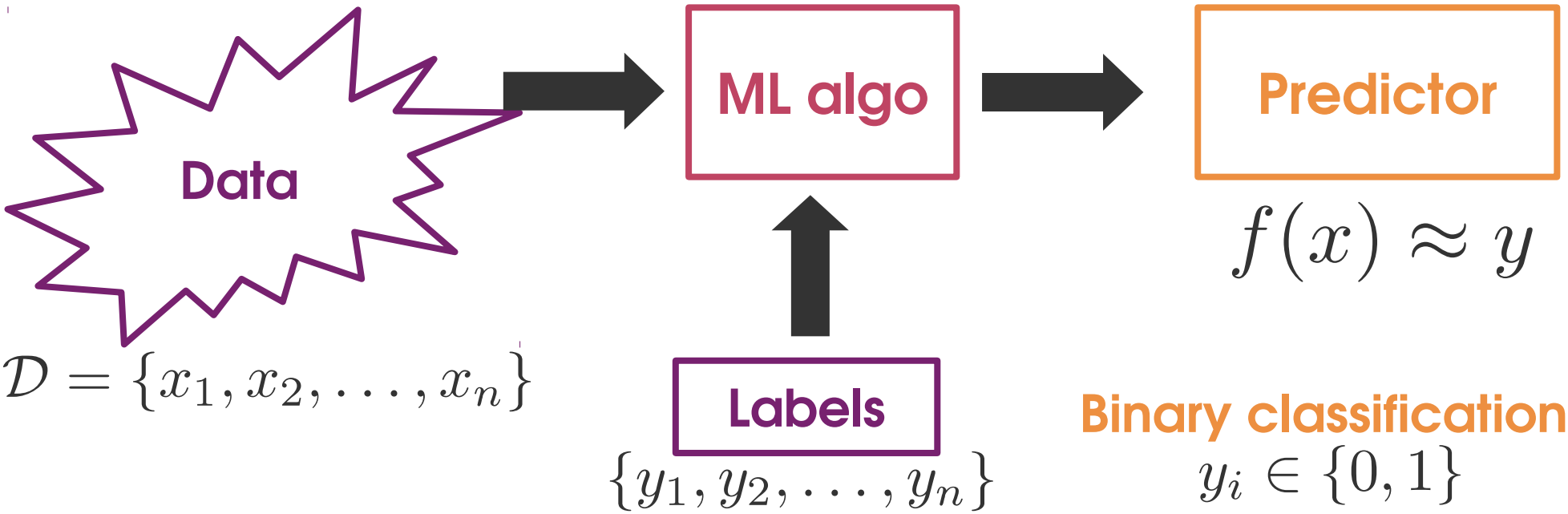
Supervised learning

Make **predictions**



Classification

Make **discrete** predictions



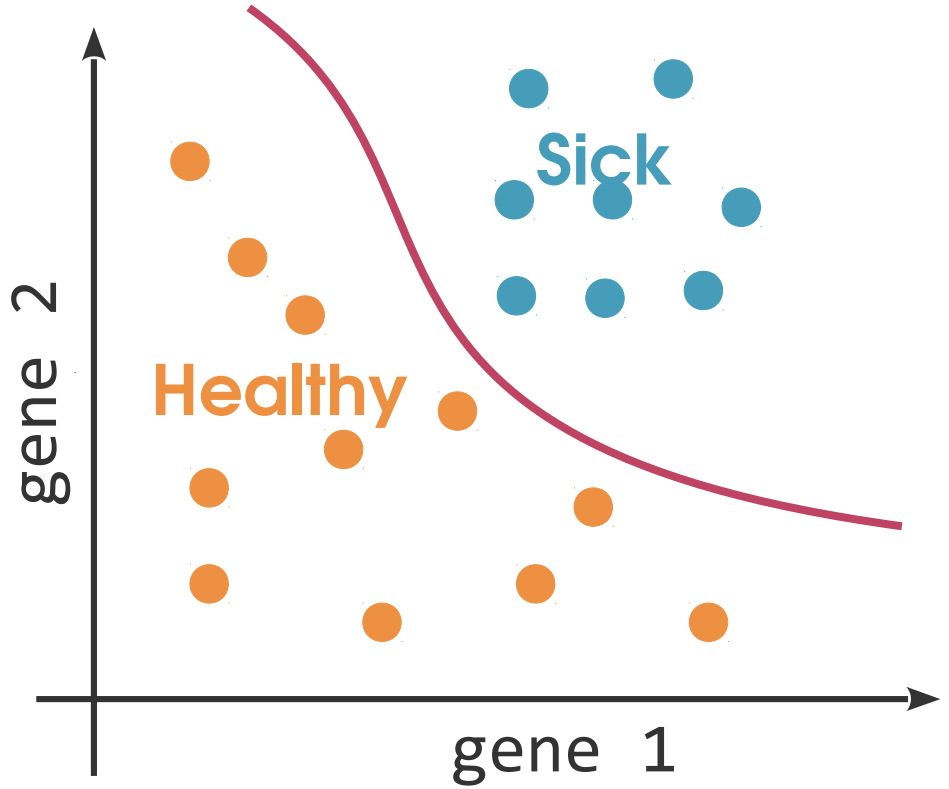
Binary classification

$y_i \in \{0, 1\}$

Multi-class classification

$y_i \in \{0, 1, \ldots, k\}$

Classification

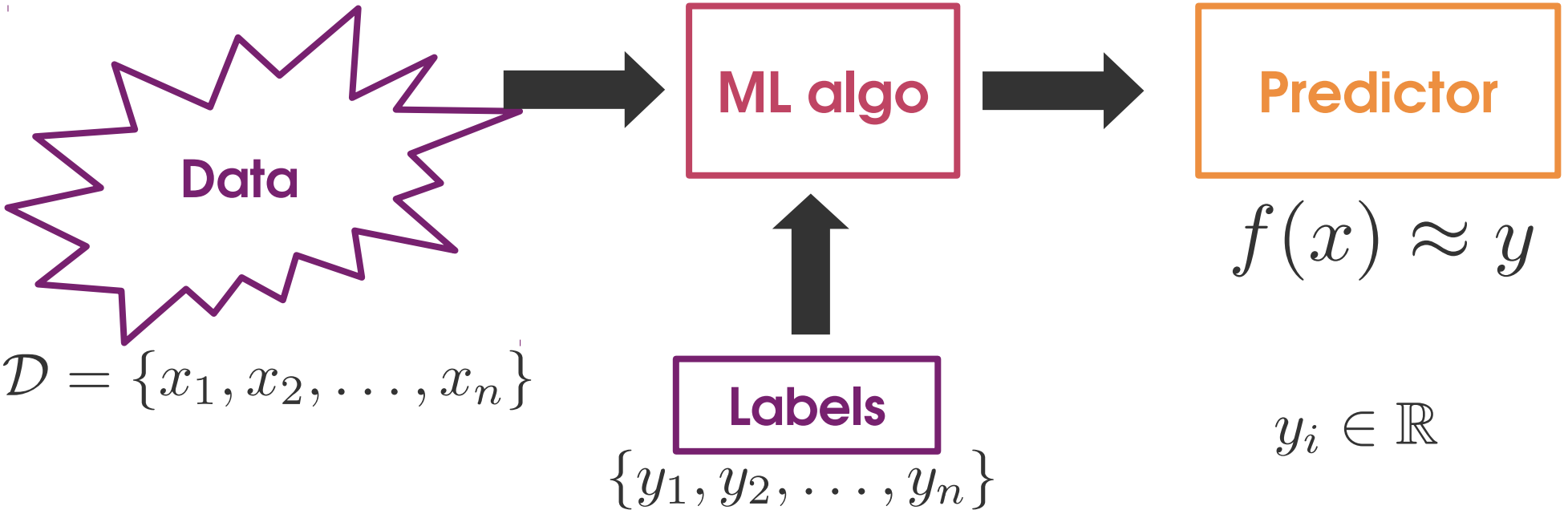


Classification: Applications

- **Face recognition:** independently of pose, lighting, occlusion (glasses, beard), make-up, hair style.
- **Character recognition:** independently of different handwriting styles.
- **Speech recognition:** account for temporal dependency.
- **Medical diagnosis:** from symptoms to illnesses.
- **Precision medicine:** from clinical & genetic features to diagnosis, prognosis, response to treatment.
- **Biometrics:** recognition/authentication using physical or behavioral characteristics: Face, iris, signature...

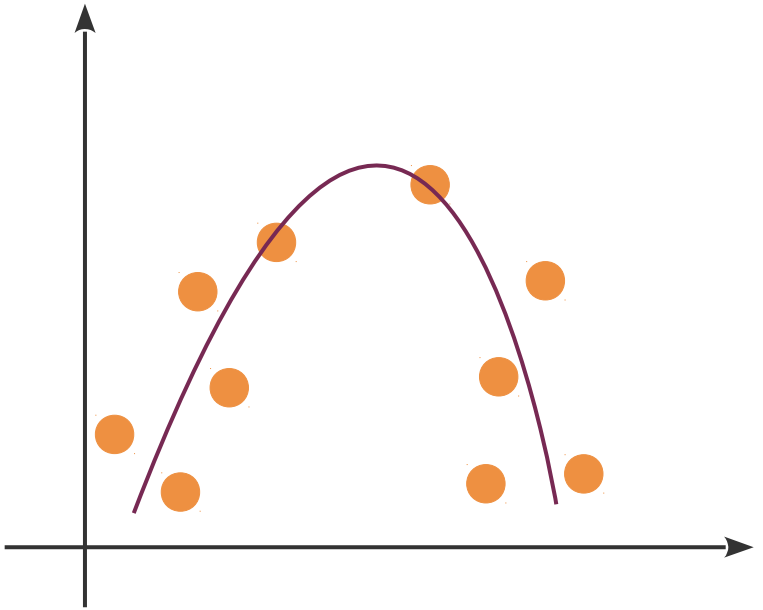
Regression

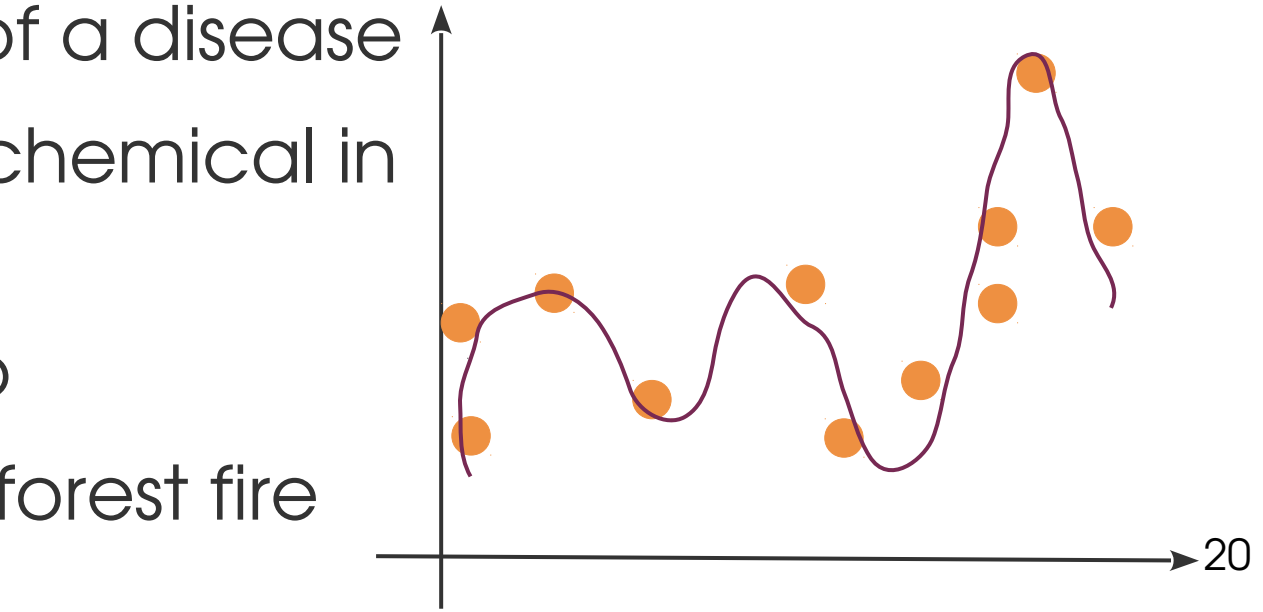
Make **continunous** predictions



Regression: Applications

- Car navigation: angle of steering
- Kinematics of a robot arm
- Binding affinities between molecules
- Age of onset of a disease
- Solubility of a chemical in water
- Yield of a crop
- Direction of a forest fire





Parametric models

- Decision function has a **set form**
- **Model complexity** ≈ number of parameters

$$f(x) = \alpha_1 x_1 + \alpha_2 (x_1 x_2)^{\beta} + \alpha_3 \log(x_3)$$

Non-parametric models

- Decision function can have **"arbitrary" form**
- **Model complexity** grows with the number of samples.

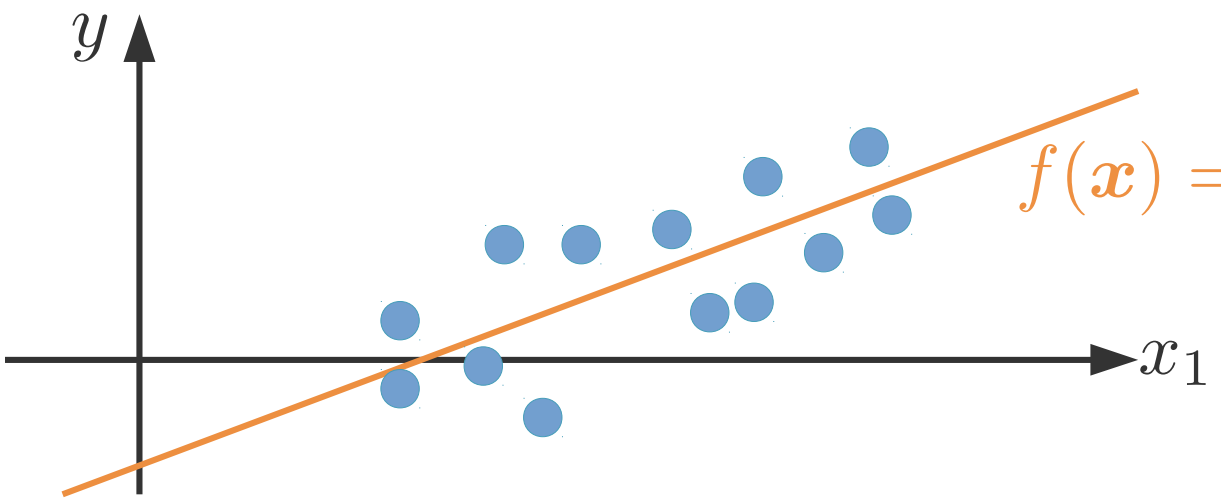
$$f(x) = \frac{1}{K} \sum_{i : x_i \in \mathcal{N}_K(x)} y_i$$

Linear models

Linear regression

$\boldsymbol{x} \in \mathbb{R}^p, y \in \mathbb{R}$

$\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$

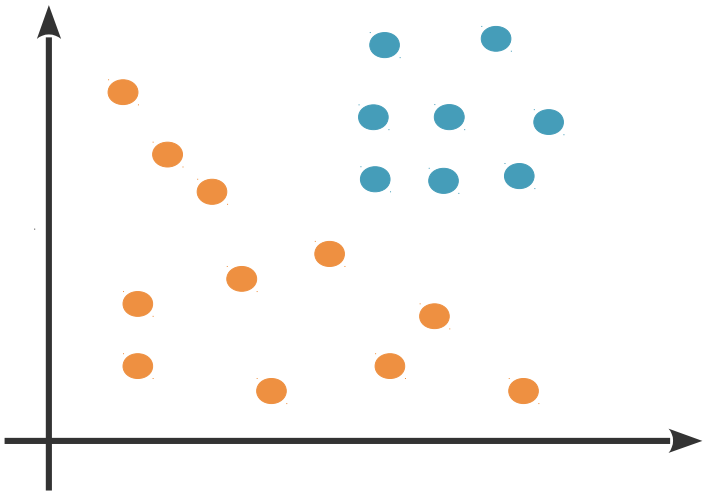


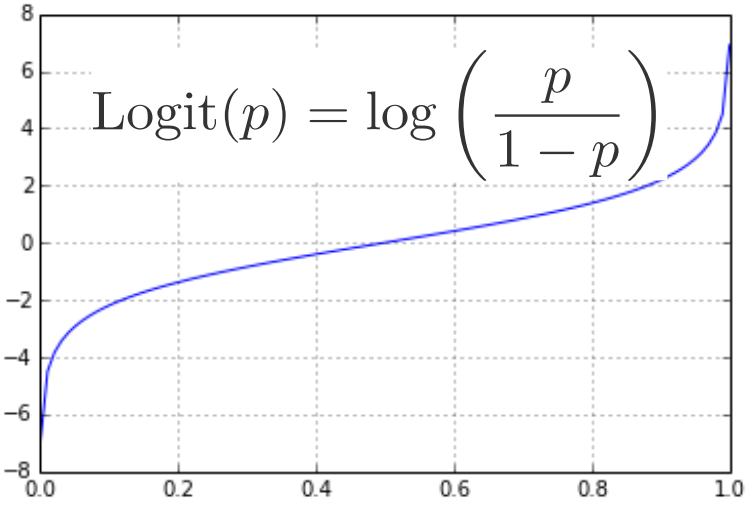
$$f(\boldsymbol{x}) = \sum_{j=1}^{p} \beta_j x_j + \beta_0$$

- **Least-squares fit:** $\boldsymbol{\beta} = \arg\min ||X\boldsymbol{\beta} - \boldsymbol{y}||_2^2$
- Equivalent to **maximizing the likelihood** $p(\mathcal{D}|\boldsymbol{\beta})$ under the assumption of Gaussian noise
- **Exact solution** $\hat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \boldsymbol{y}$ if X has full column rank

Classification: logistic regression

Linear function → probability





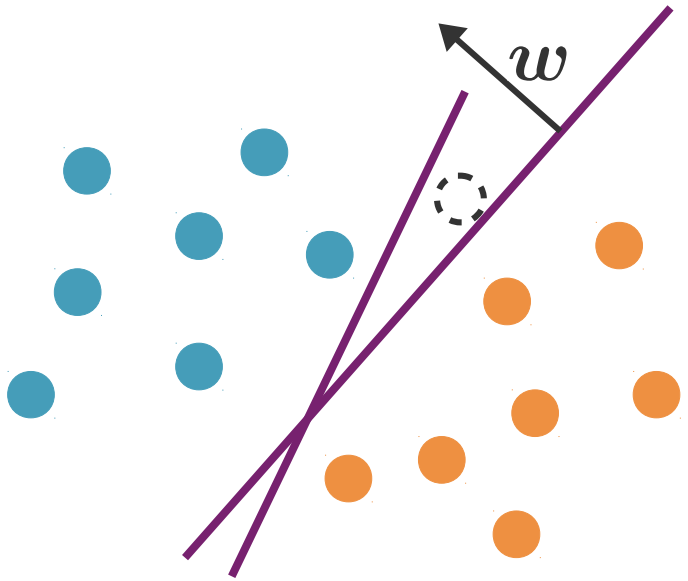
$$\text{logit}(p(y=1|\boldsymbol{x})) = \sum_{j=1}^{p} \beta_j x_j + \beta_0$$

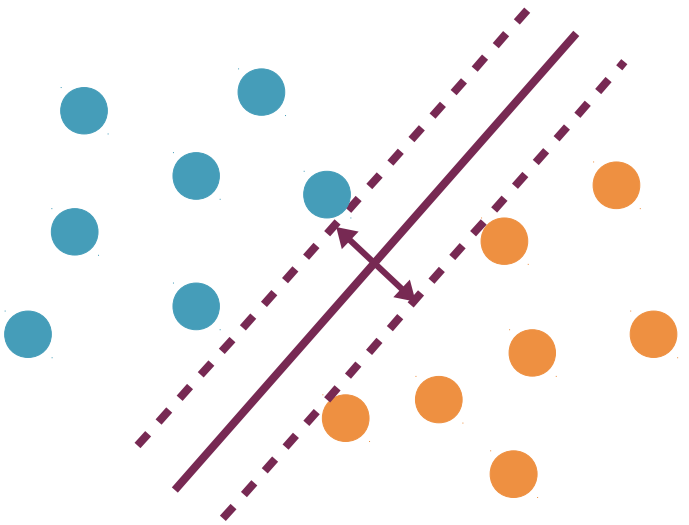
- Solve by **maximizing the likelihood**
- No analytical solution
- Use **gradient descent.**

Support Vector Machines

Large margin classifier

- Find the separating hyperplane with the **largest margin.**





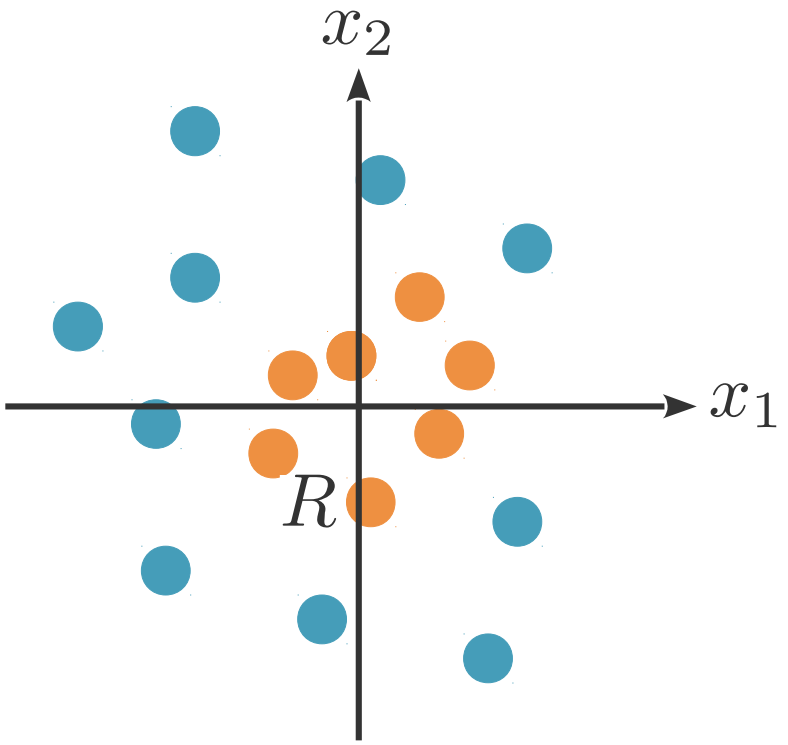
$$\arg\min_{\boldsymbol{w},b}\left(\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n}\max(0, 1 - y^i(\langle \boldsymbol{w}, x^i\rangle + b))\right)$$

f(x)

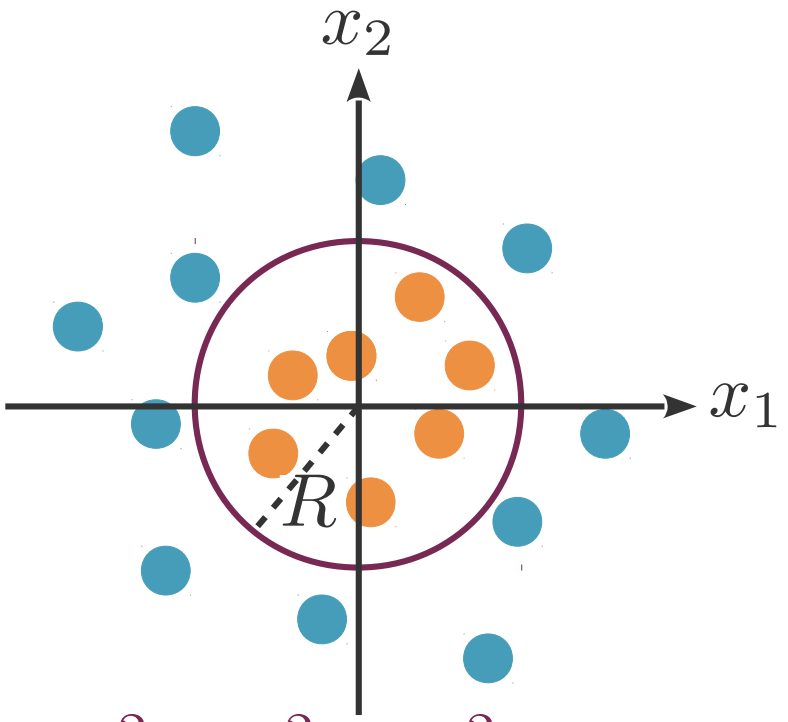
inverse of the margin

prediction error

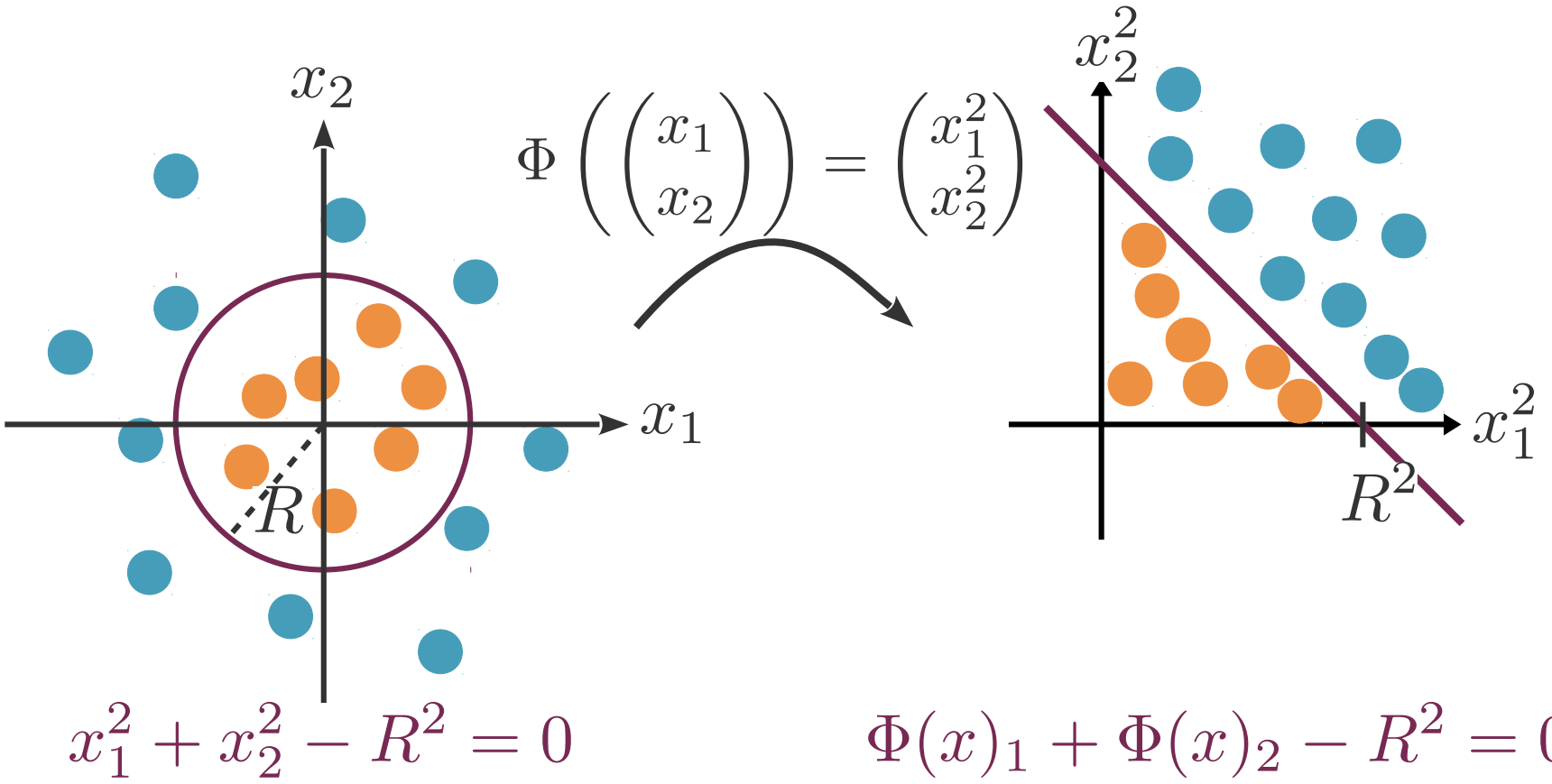
When lines are not enough



When lines are not enough



When lines are not enough



- **Non-linear mapping** to a **feature space**
- https://www.youtube.com/watch?v=3liCbRZPrZA

The kernel trick

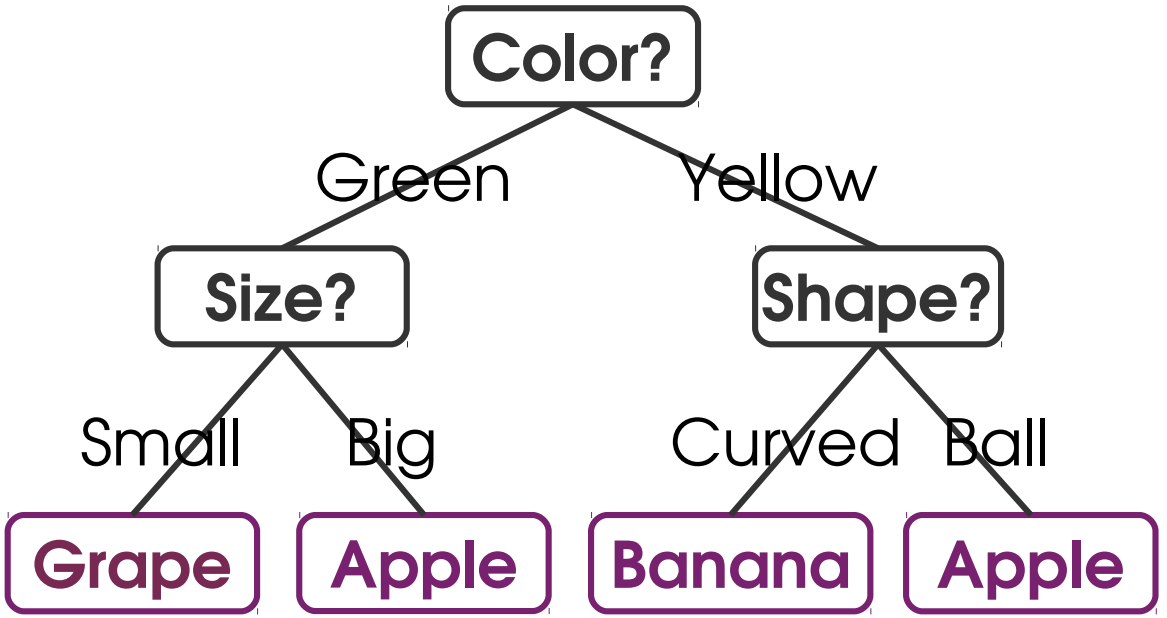
- The solution & SVM-solving algorithm can be expressed using **only** $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle$
- Never need to **explicitly compute** $\Phi(\boldsymbol{x})$
- k: **kernel**
  - must be **positive semi-definite**
  - can be interpreted as a **similarity function.**

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^* y^i k(\boldsymbol{x}^i, \boldsymbol{x}) + b^*$$

- **Support vectors:** training points for which $\alpha \neq 0$.

Random Forests

Decision trees



- Well suited to **categorical features**
- Naturally handle **multi-class classification**
- **Interpretable**
- **Perform poorly.**

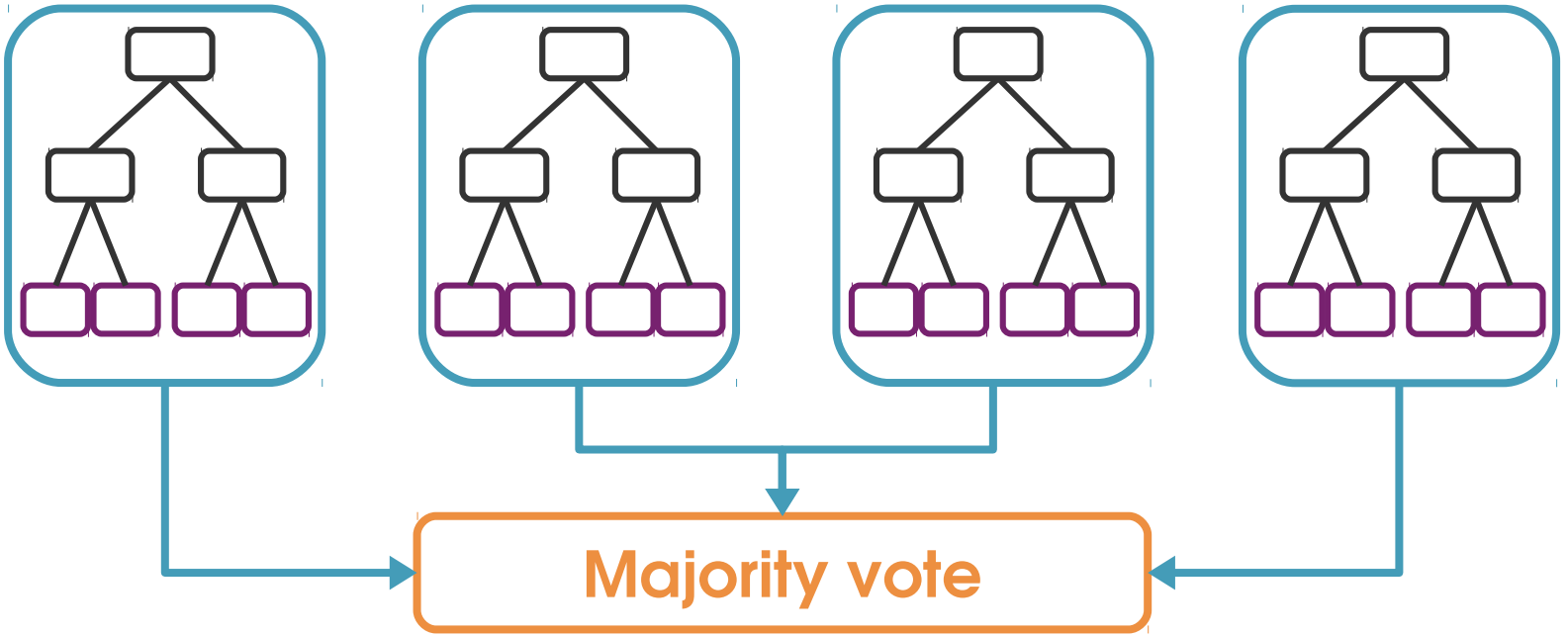
Ensemble learning

- Combining **weak learners** averages out their individual errors (**wisdom of crowds**)
- Final prediction:
  - Classification: **majority vote**
  - Regression: **average.**
- **Bagging:** weak learners are trained on **bootstraped samples** of the data (Breiman, 1996).

  bootstrap: sample n, with replacement.
- **Boosting:** weak learners are built iteratively, based on performance (Shapire, 1990).

Random forests

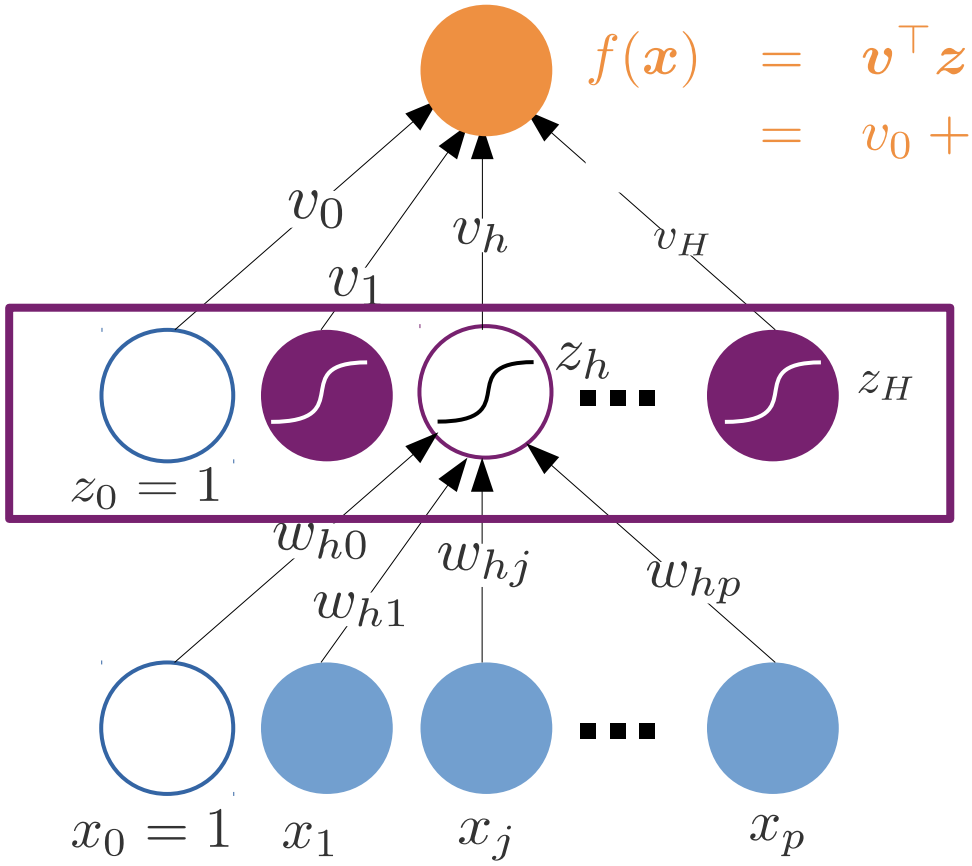
- Combine many decision trees



- Each tree is trained on a data set created using
  - A **bootstrap sample** (sample with replacement) of the **data**
  - A **random sample** of the **features.**
- Very **powerful** in practice.

(Deep) Neural Networks

(Deep) neural networks

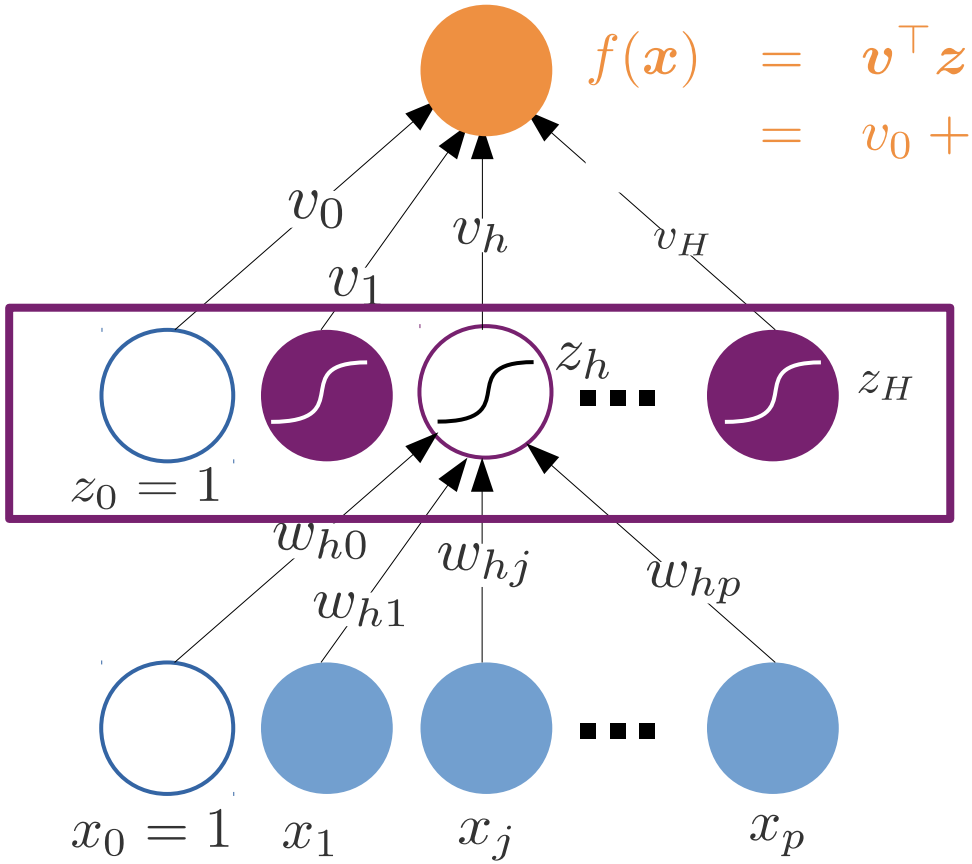


hidden layer

$$z_h = \frac{1}{1 + e^{-\boldsymbol{w}_h^\top \boldsymbol{x}}}$$

- Nothing more than a (possibly complicated) **parametric model**

(Deep) neural networks

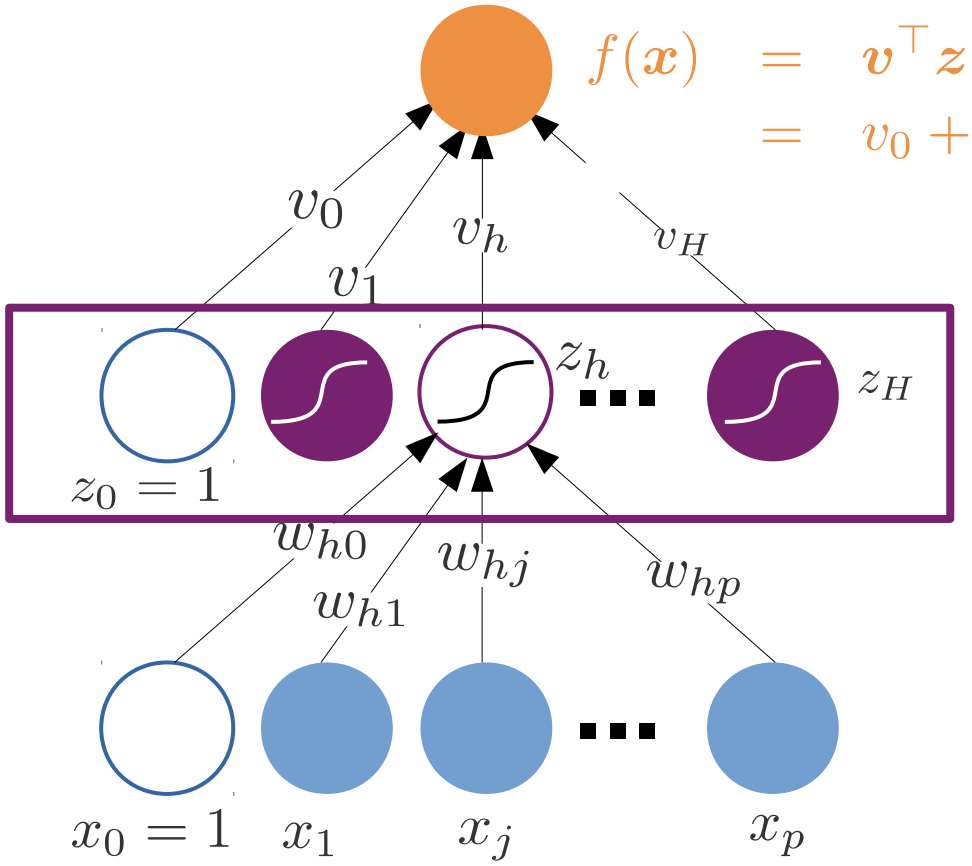


hidden layer

$$z_h = \frac{1}{1 + e^{-w_h^\top x}}$$

- **Fitting weights:**
  - **Non-convex** optimization problem
  - Solved with gradient descent
  - Can be difficult to tune.

(Deep) neural networks



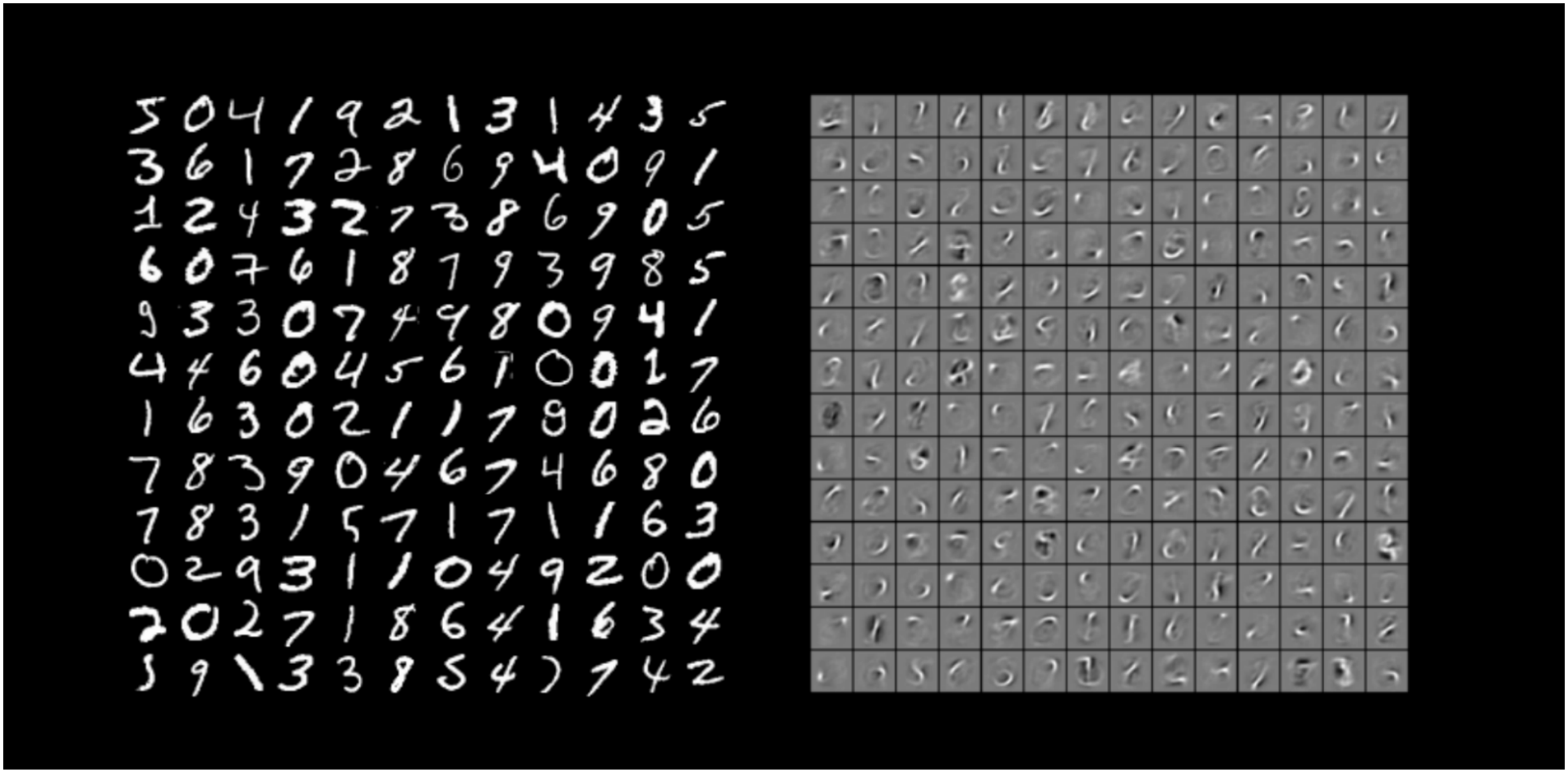
hidden layer

$$z_h = \frac{1}{1 + e^{-\boldsymbol{w}_h^\top \boldsymbol{x}}}$$

- Learn an **internal representation** of the data on which a **linear model** works well.
- Currently one of the **most powerful** supervised learning algorithms **for large training sets.**

(Deep) neural networks

Internal representation of the digits data



Yann Le Cun et al. (1990)

Generalization & overfitting

Generalization

- **Goal:** build models that make good predictions on **new data.**
- Models that work "too well" on the data we learn on tend to model noise as well as the underlying phenomenon: **overfitting.**

Overfitting (Classification)

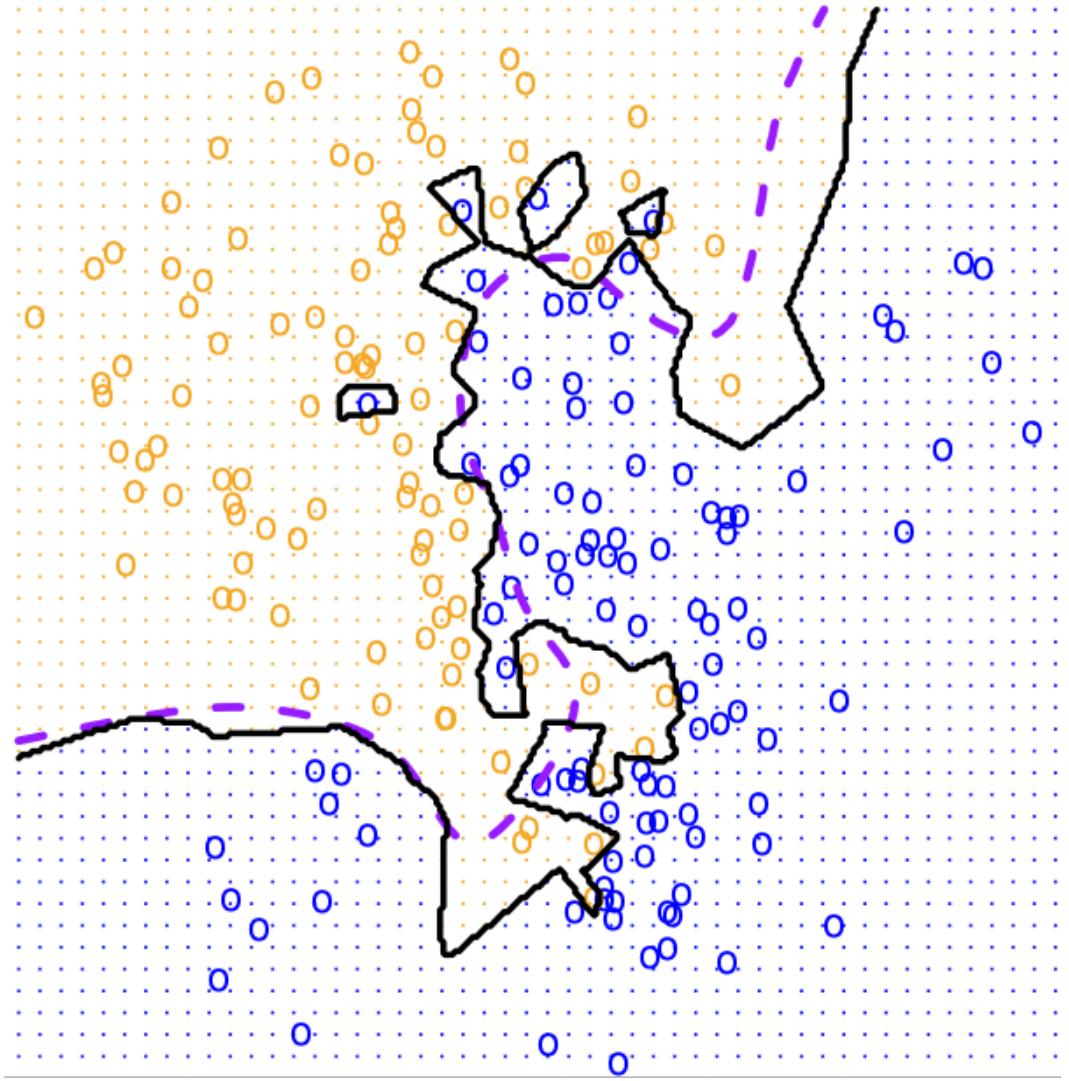


Figure from Hastie, Tibshirani & Friedman, *The Elements of Statistical Learning.*

Overfitting (Regression)

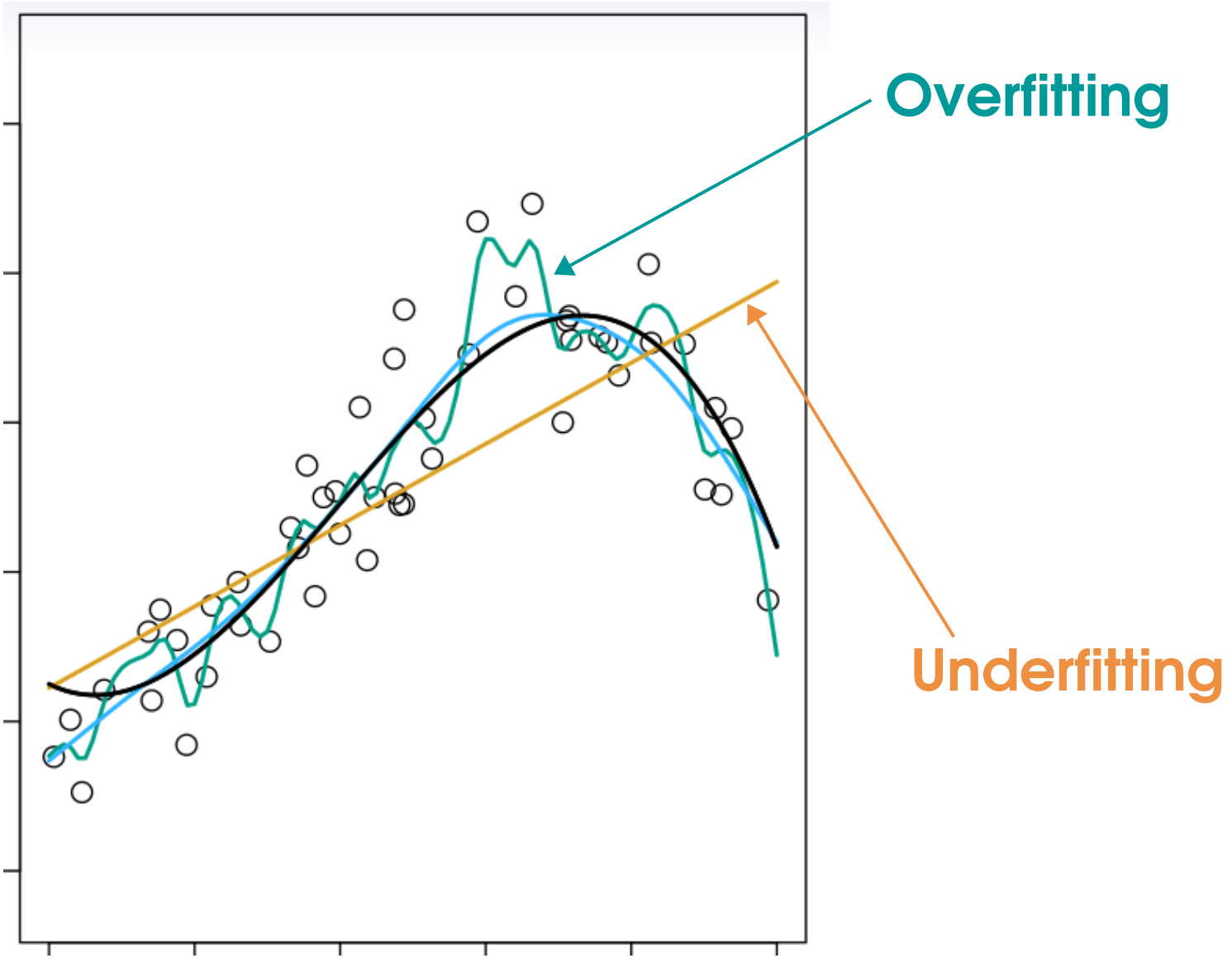
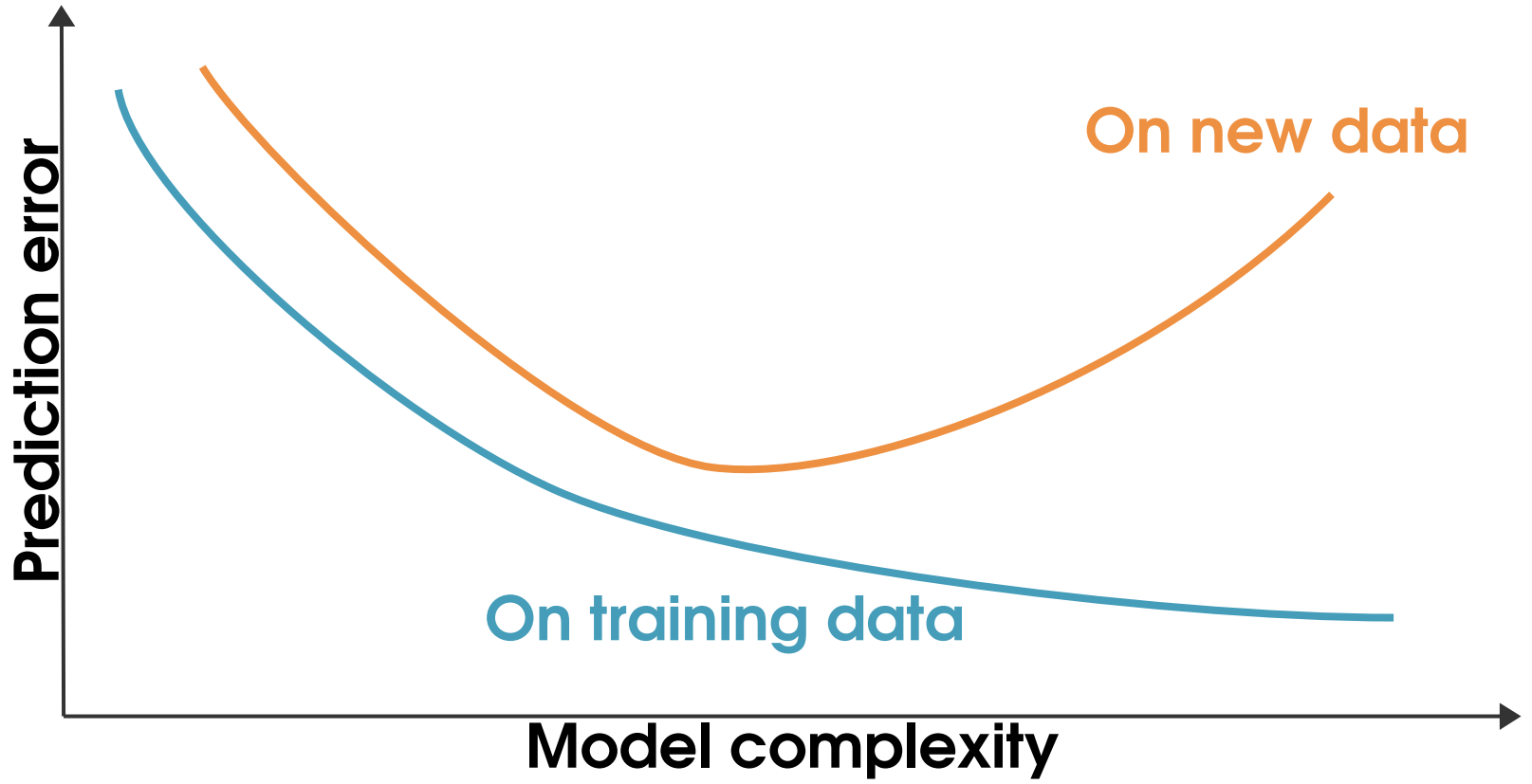


Figure from Hastie, Tibshirani & Friedman, *The Elements of Statistical Learning.*

Model complexity

Simple models are

- more **plausible** (Occam's Razor)
- easier to **train, use,** and **interpret.**



Regularization

- Prevent overfitting by designing an **objective function** that accounts not only for the **prediction error** but also for **model complexity.**

```
min (empirical_error + λ*model_complexity)
```

- Rember the SVM

$$\arg\min_{\boldsymbol{w},b} \left(\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{n} \max(0, 1 - y^i(\langle \boldsymbol{w}, x^i \rangle + b)) \right)$$

inverse of the margin

f(x)

prediction error

Ridge regression

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \boxed{||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2} + \lambda \boxed{||\boldsymbol{\beta}||_2^2}$$

prediction error

regularizer

hyperparameter

- **Unique solution, always exists**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top \boldsymbol{y}$$

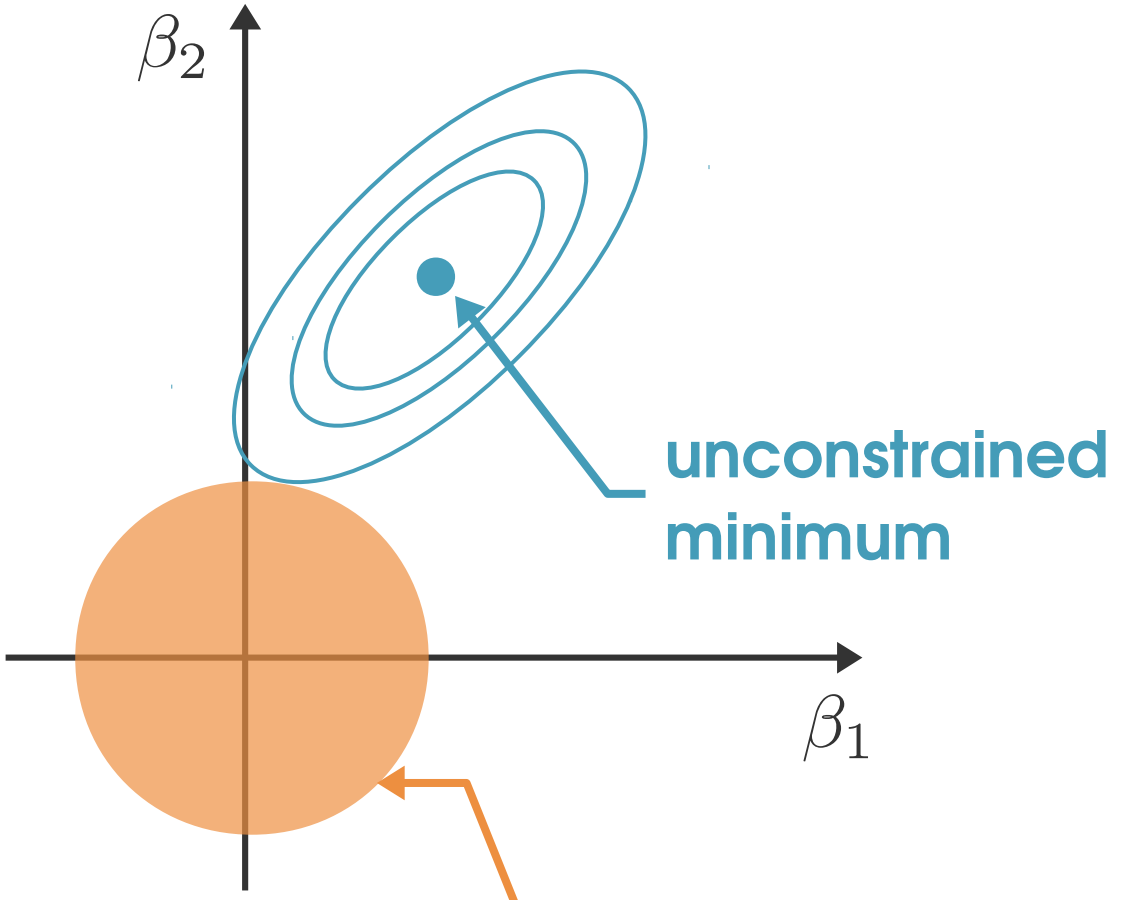
- **Grouped selection:**

  Correlated variables get similar weights.

- **Shrinkage**

  Coefficients shrink towards 0.

Geometry of ridge regression



Model selection & evaluation

Evaluation on held-out data

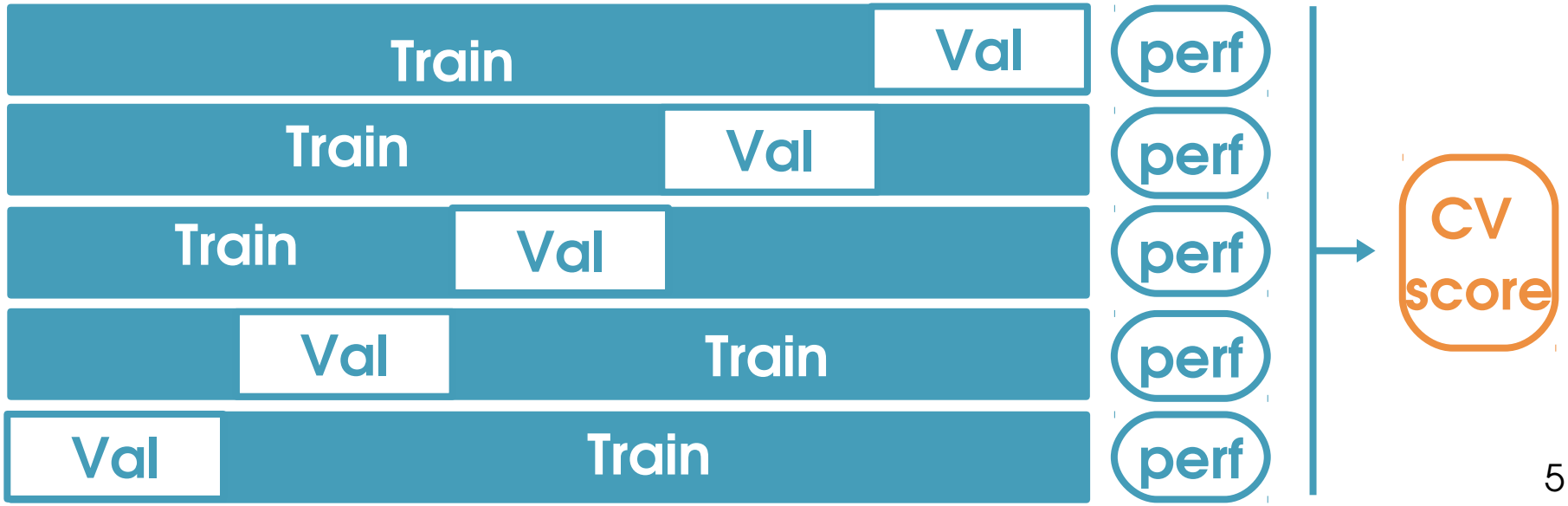
- If we evaluate the model on the data we've used to train it, we risk **over-estimating performance.**
- **Proper procedure:**
  - Separate the data in **train/test** sets

Evaluation on held-out data

- **Proper procedure:**
  - Separate the data in **train/test** sets

Evaluation on held-out data

- **Proper procedure:**
  - Separate the data in **train/test** sets
  - Use a **cross-validation** on the train set to find the best algorithm + hyperparameter(s)



Evaluation on held-out data

- **Proper procedure:**
  - Separate the data in **train/test** sets
  - Use a **cross-validation** on the train set to find the best algorithm + hyperparameter(s)
  - Train this best algorithm + hyperparameter(s) on the entire train set
  - The performance on the test set estimates **generalization performance.**

ML Toolboxes

- **Python: scikit-learn**

  `http://scikit-learn.org`

  Get started in python: `http://scipy-lectures.github.io/`

- **R: Machine Learning Task View**

  `http://cran.r-project.org/web/views/MachineLearning.html`

- **Matlab™: Machine Learning with MATLAB**

  `http://mathworks.com/machine-learning/index.html`

  - Statistics and Machine Learning Toolbox
  - Neural Network Toolbox

Summary

Machine learning =

data + model + objective function

- Catalog:
  - Supervised vs unsupervised
  - Parametric vs non-parametric
  - Linear models, SVMs, random forests, neural networks.
- **Key concerns:**
  - avoid overfitting
  - Measure generalization performance.