Foundations of Machine Learning
École Centrale Paris — Fall 2016

2. Supervised learning

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

Learning objectives

- **Formulate a supervised learning problem** formally;
- Explain some **basic elements of learning theory;**
- Understand the notion of **model complexity.**

Supervised classification

Learning a class from examples

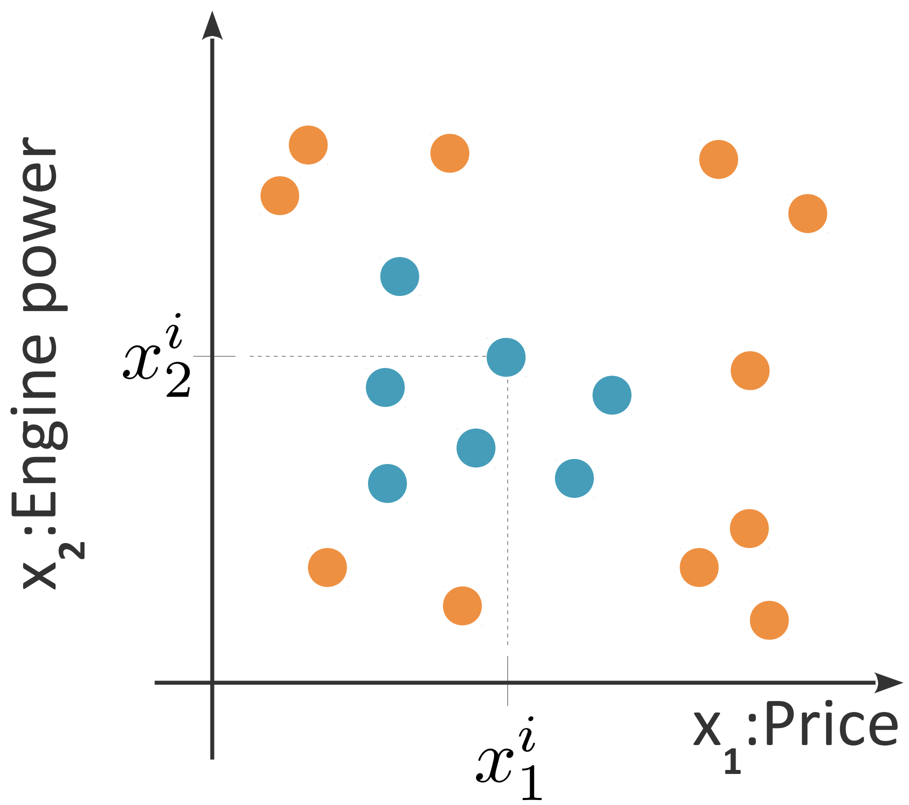
- Class C of a “family car”
  - **Prediction**: Is car x a family car?
  - **Knowledge extraction**: What do people expect from a family car?
- **Output**:

  Positive (+) and **negative** (–) examples
- **Input representation**:

  x1: price

  x2 : engine power

Training set X



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

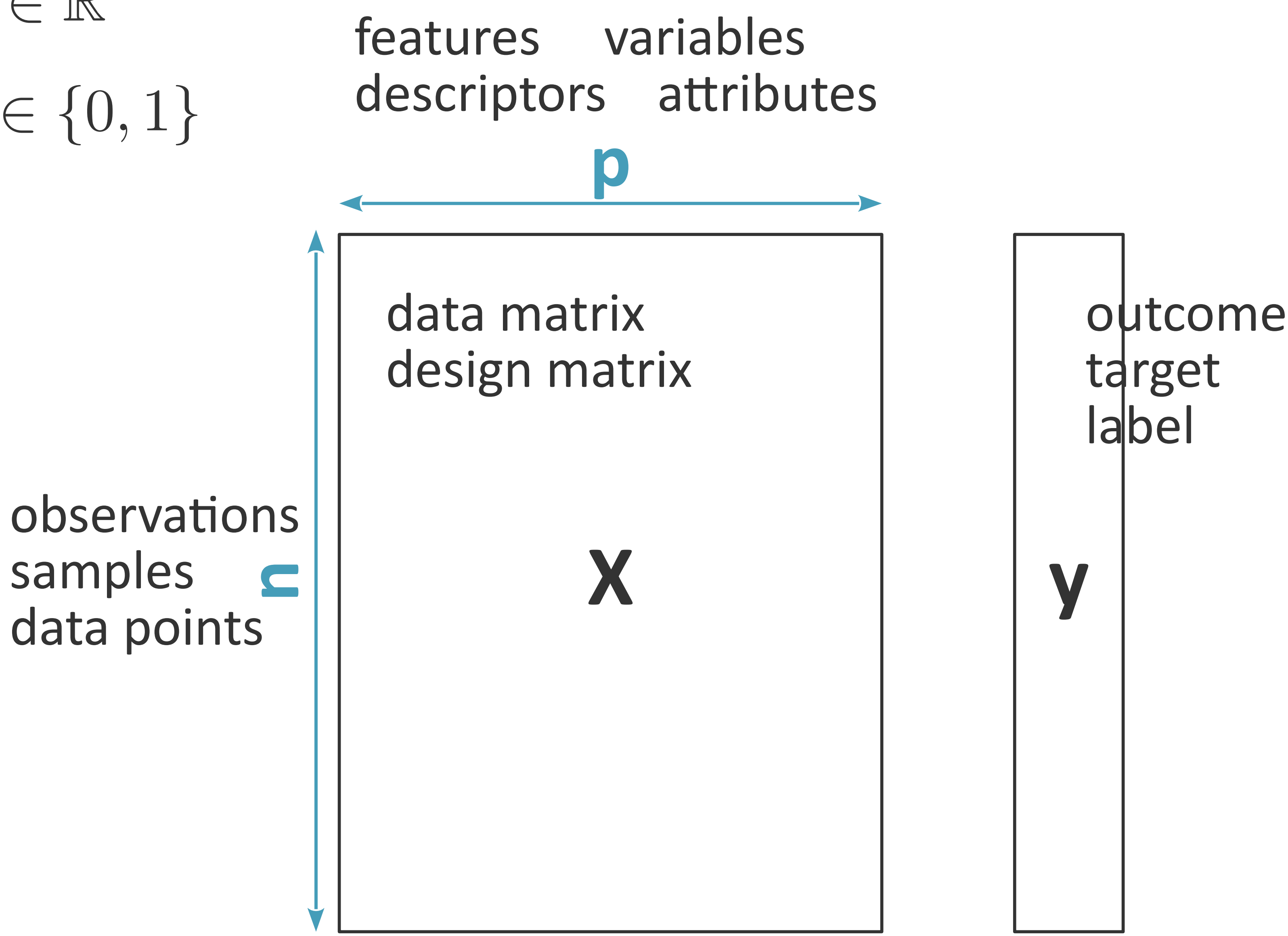
$$y^i = \begin{cases} 1 & \text{if } \boldsymbol{x}^i \in \mathcal{P} \quad + \\ 0 & \text{if } \boldsymbol{x}^i \in \mathcal{N} \quad - \end{cases}$$

$$\boldsymbol{x}^i = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}$$

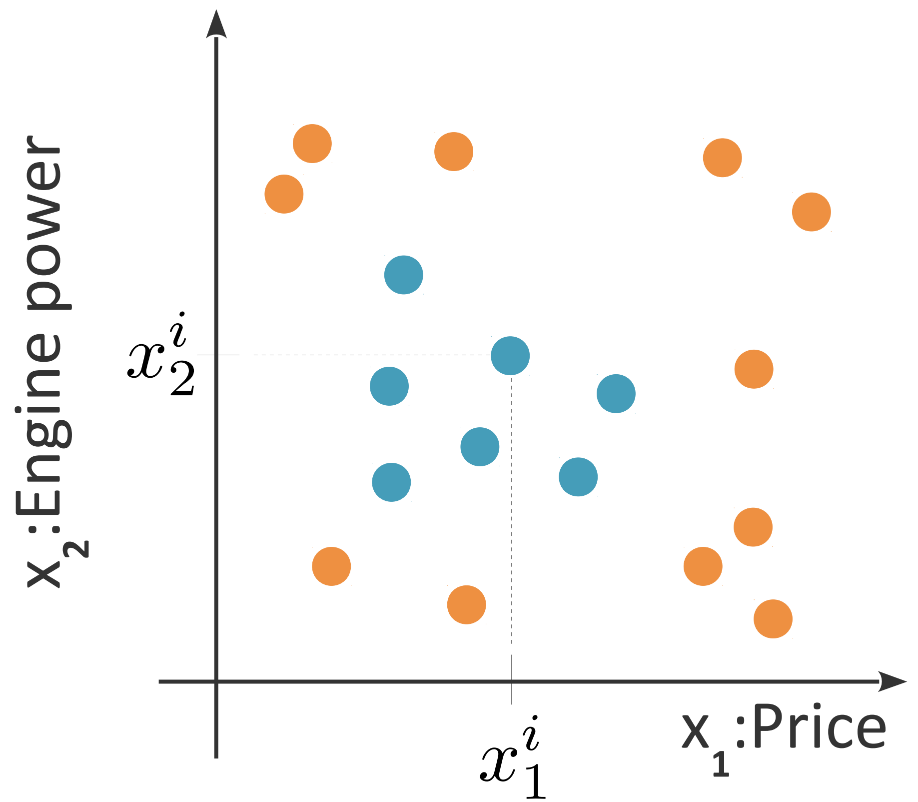
Classification setting

$$x_j^i \in \mathbb{R}$$

$$y^i \in \{0, 1\}$$

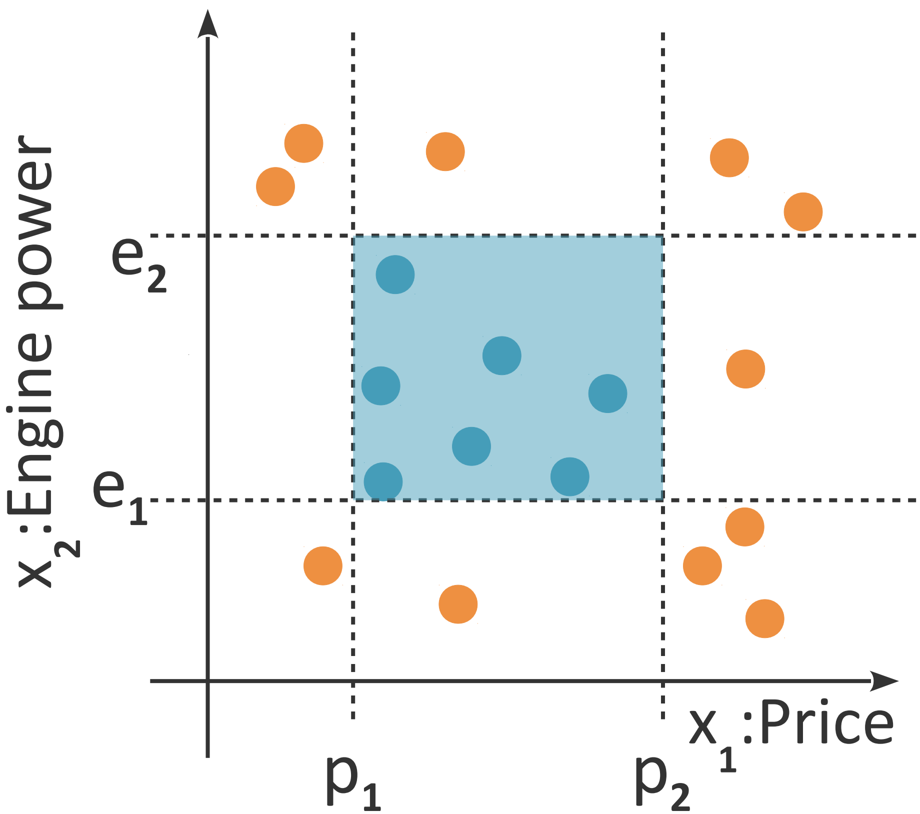


Version space



What shape do you think the discriminant should take?

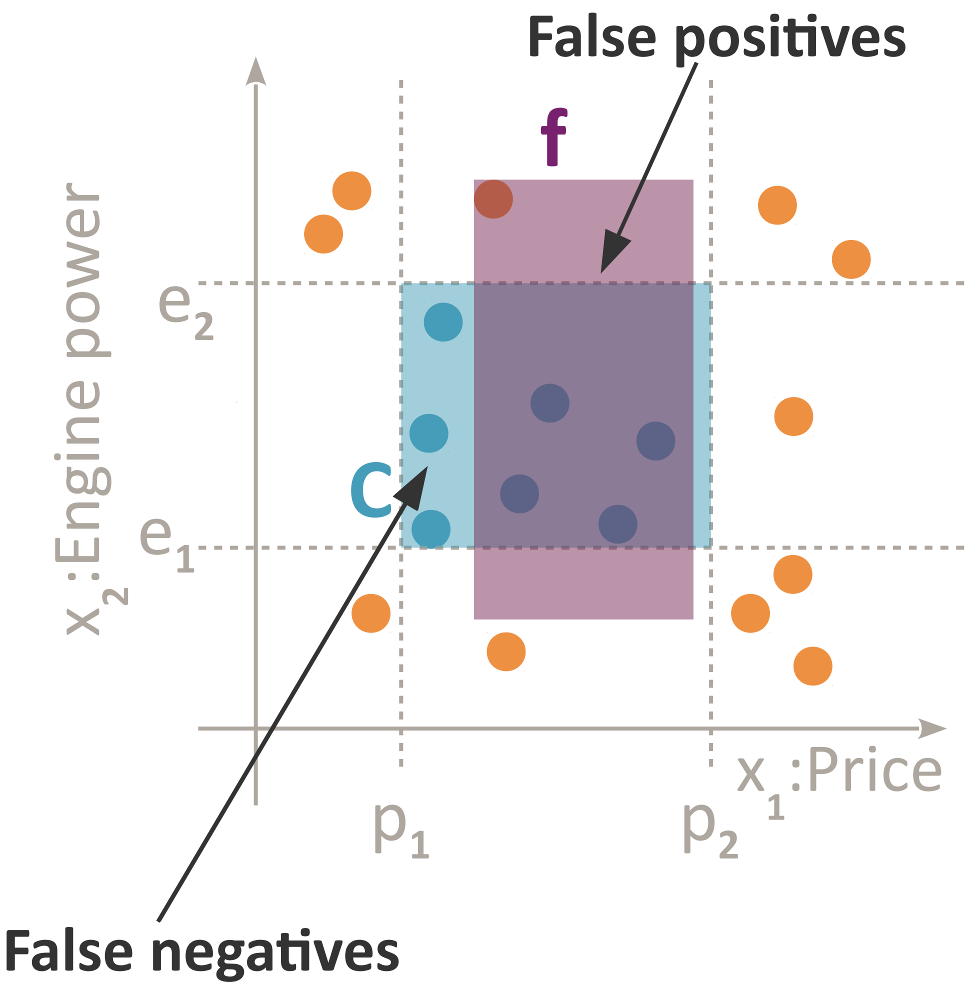
Class C



- **Belief about family cars:**
  - price between p_1 and p_2
  - engine power between e_1 and e_2
- **Hypothesis space** from which we believe C is drawn = set of rectangles

$$(p_1 \leq x_1 \leq p_2) \text{ AND } (e_1 \leq x_2 \leq e_2)$$

Hypothesis f



$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f \text{ says } \boldsymbol{x} \in \mathcal{P} \\ 0 & \text{if } f \text{ says } \boldsymbol{x} \in \mathcal{N} \end{cases}$$

Empirical error of f on the training set:

$$E(f|X) = \frac{1}{n}\sum_{i=1}^{n} 1_{f(\boldsymbol{x}^i) \neq y^i}$$

Choosing f in H

- **Generalization**

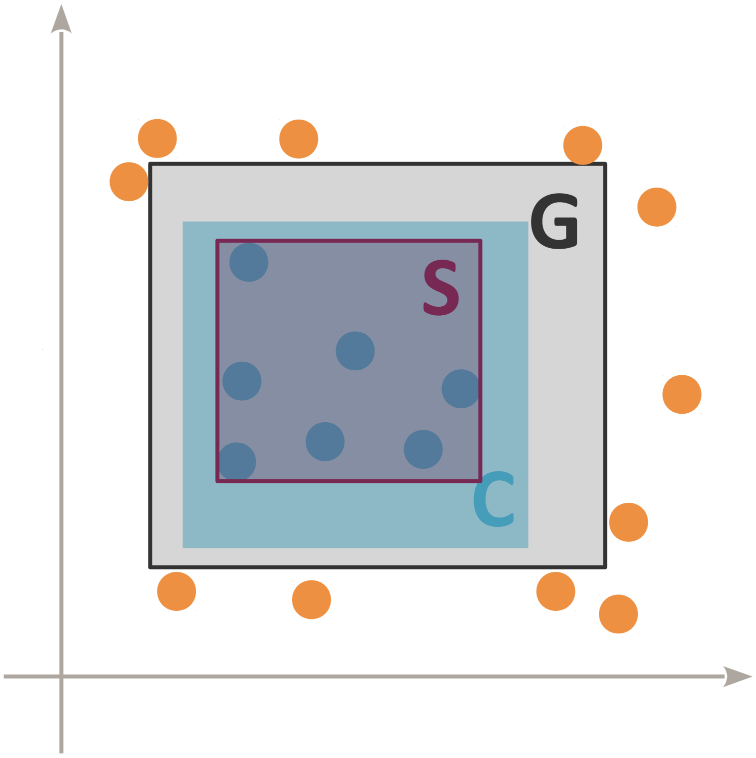
  We want f to work well on unseen data

- **Most specific hypothesis**

  S: Tight to the positive examples

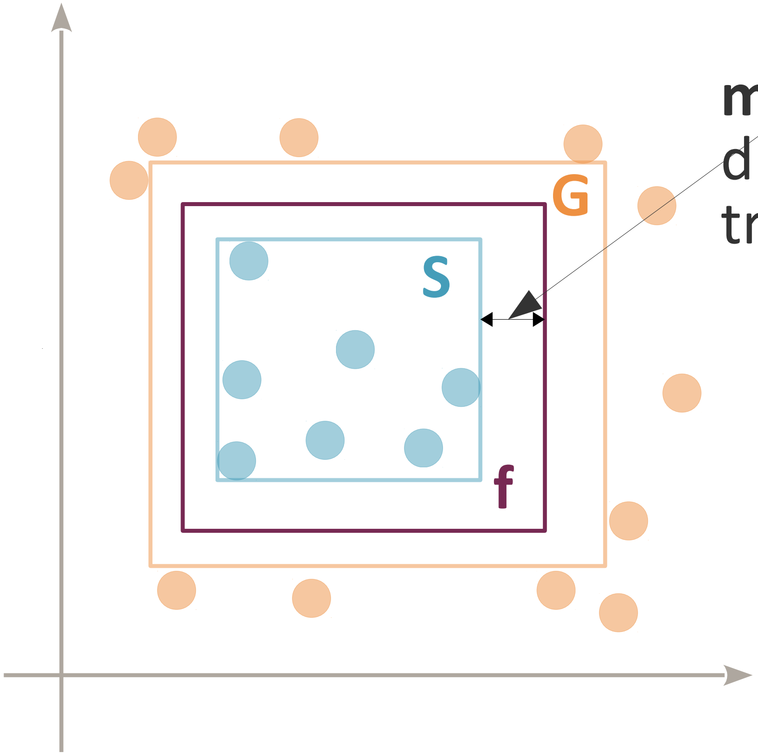
- **Most generic hypothesis**

  G: Tight to the negative examples



Where do you think we should put f?

- Any hypothesis between S and G is **consistent** with the training set (i.e. makes no mistake on X).
- **Version space:** set of consistent hypotheses [Mitchell, 1997]
- Choose f halfway between S and G = maximize the **margin**

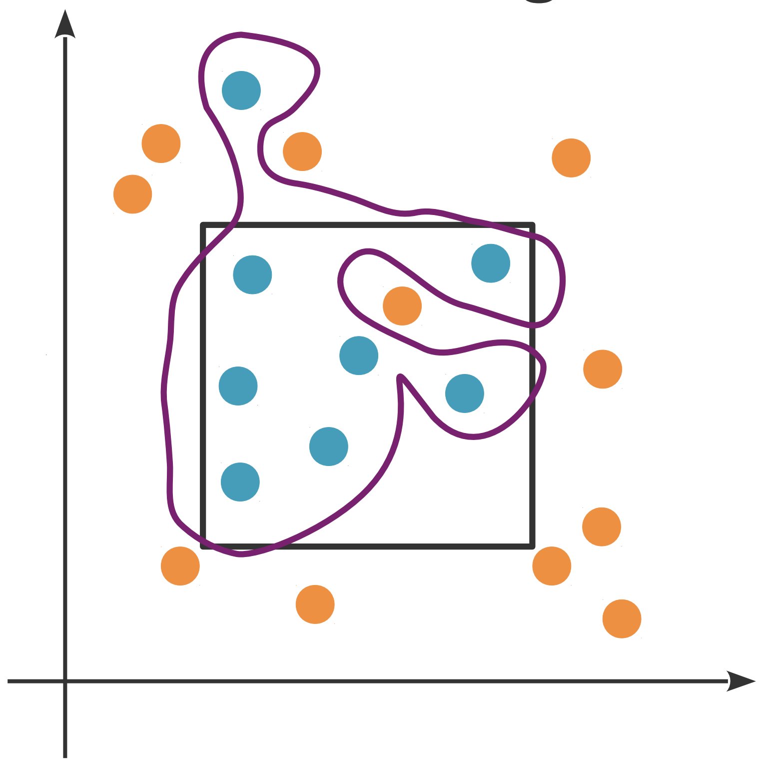


margin: distance between the discriminant and its nearest training point.

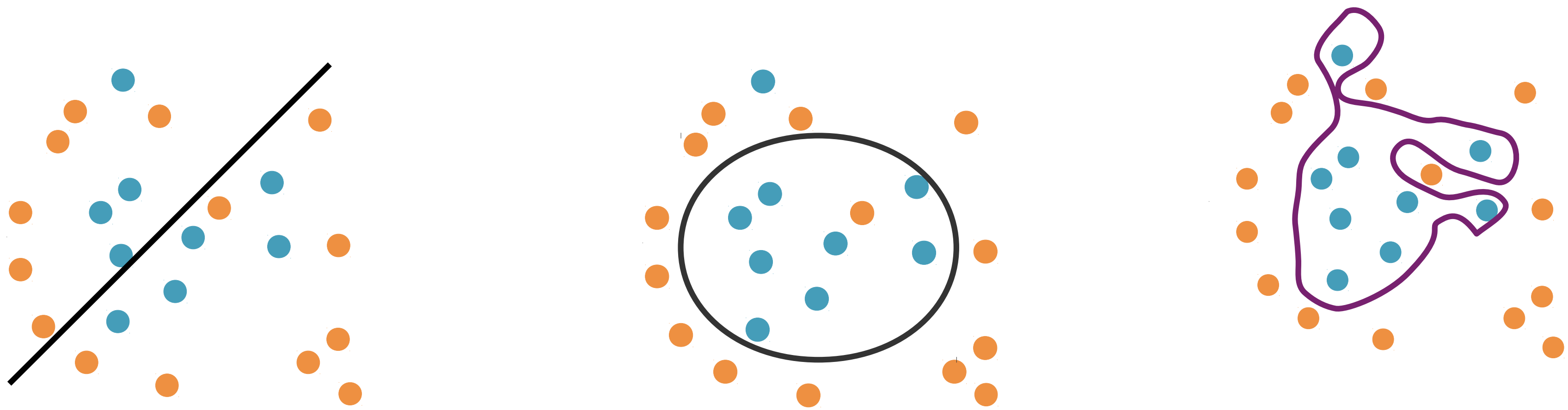
Model complexity

Noise in the data

- Imprecision in **recording the features**
- **Errors in labeling** the data points (**teacher noise**)
- **Missing features** (**hidden** or **latent**)
- Making no errors on the training set might not be possible.



Models of increasing complexity



Noise and model complexity

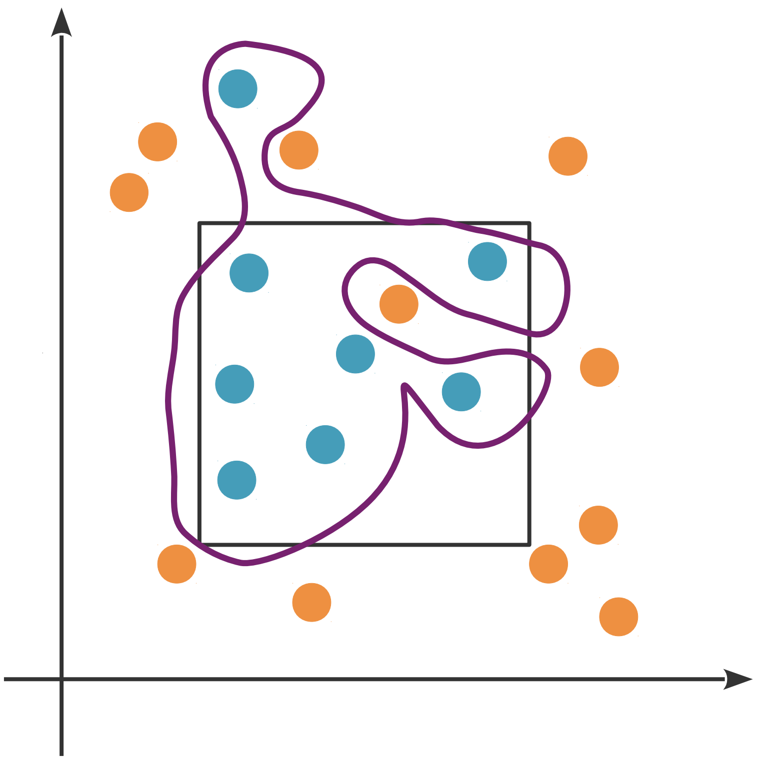
- **Use simple models!**
  - Easier to **use**

    lower computational complexity
  - Easier to **train**

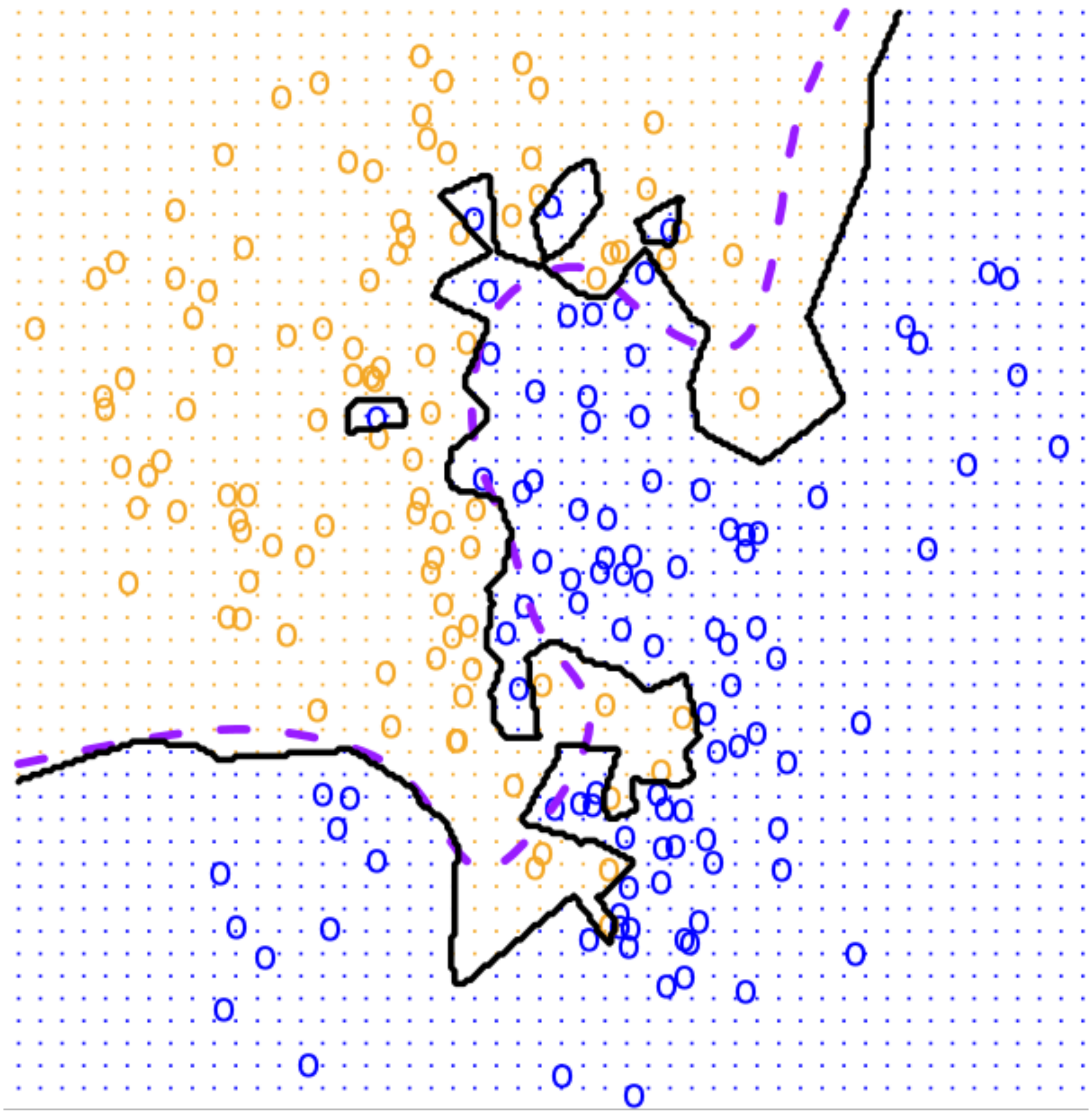
    lower space complexity
  - Easier to **explain**

    more interpretable
  - **Generalize better**

    Occam's razor: simpler explanations are more plausible.



Overfitting



- **What are the empirical errors of the black and purple classifiers?**
- **Which model seems more likely to be correct?**

Model selection & generalization

- **Generalization**:

  How well a model performs on new data
- **Overfitting**:

  f more complex than C
- **Underfitting**:

  f less complex than C.

Bias-variance tradeoff

- **Bias:** difference between the expected value of the estimator and the true value being estimated.

$$\mathrm{Bias}(\hat{y}) = \mathbb{E}(\hat{y} - c(\boldsymbol{x}))$$

  - A simpler model has a higher bias.
  - **High bias can cause underfitting.**
- **Variance:** deviation from the expected value of the estimates.

$$\mathrm{Var}(\hat{y}) = \mathbb{E}((\hat{y} - \mathbb{E}(\hat{y}))^2)$$

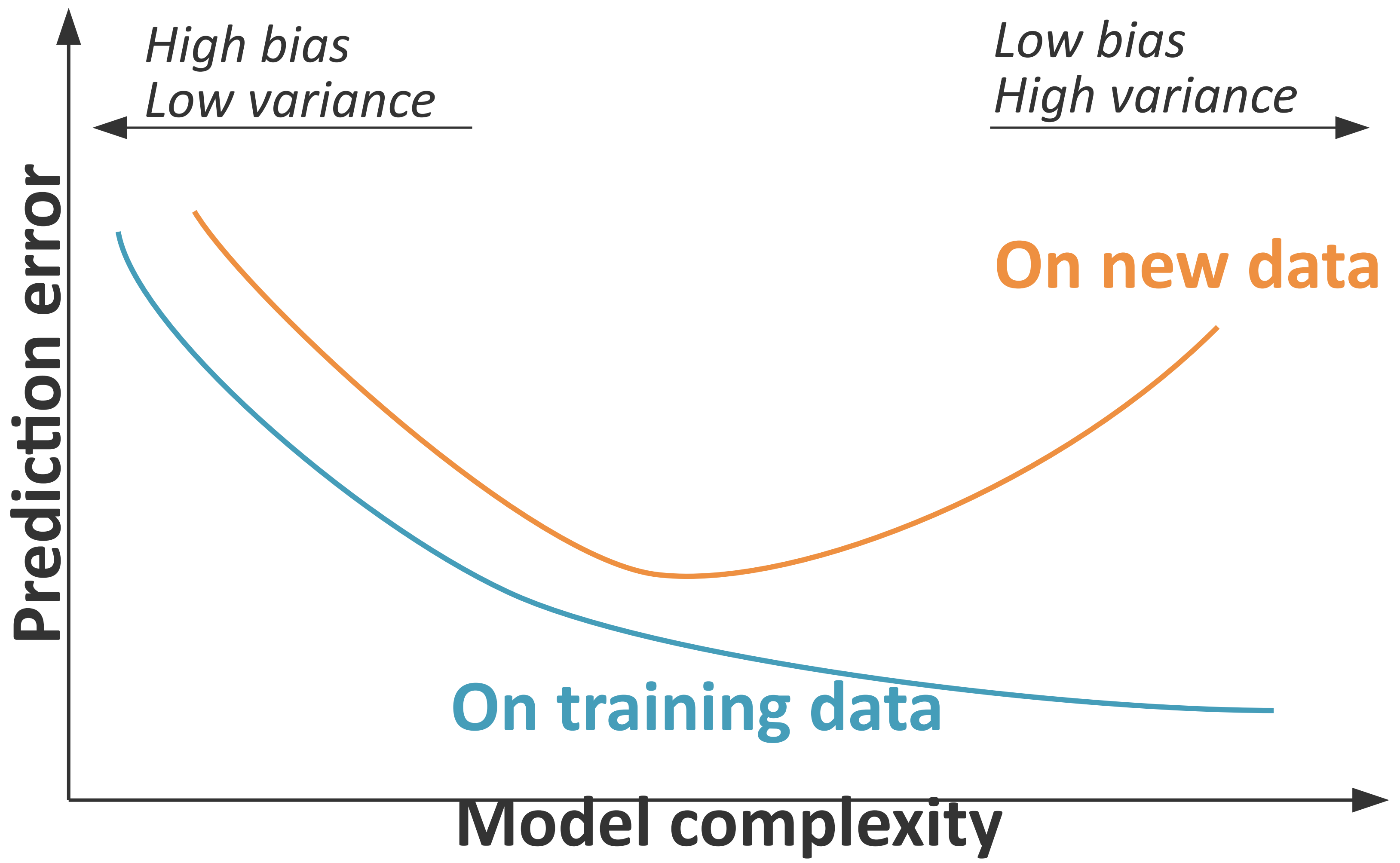
  - A more complex model has a higher variance.
  - **High variance can cause overfitting.**

Bias-variance decomposition

- $\text{Bias}(\hat{y}) = \mathbb{E}(\hat{y} - f(\boldsymbol{x}))$
- $\text{Var}(\hat{y}) = \mathbb{E}((\hat{y} - \mathbb{E}(\hat{y}))^2)$
- **Mean squared error:**

$$\begin{aligned} \text{MSE}(\hat{y}) &= \mathbb{E}(f(\boldsymbol{x}) - \hat{y})^2 \\ &= \text{Var}(\hat{y}) + \text{Bias}^2(\hat{y}) \end{aligned}$$

Generalization error vs. model complexity



Complexity of the hypothesis space: Vapnik-Chervonenkis dimension

VC dimension

- N points can be labeled in 2^N ways as +/-
- H **shatters** N if there exists f in H **consistent** for any of these labelings.

- **Vapnik-Chervonenkis dimension** of H = max number of points that can be shattered by H

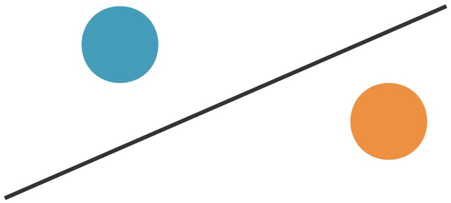
  $VC(H)=N$

In the plane:
What is the VC dimension of a line?
What is the VC dimension of an axis-aligned rectangle?

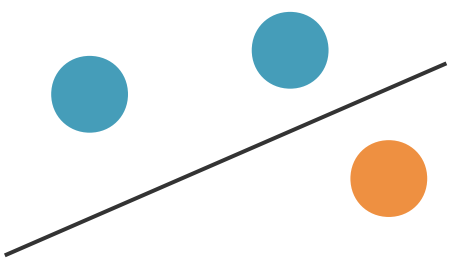
[Vapnik & Chervonenkis, 1970]

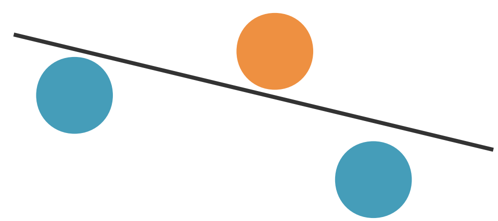
VC dimension of a line

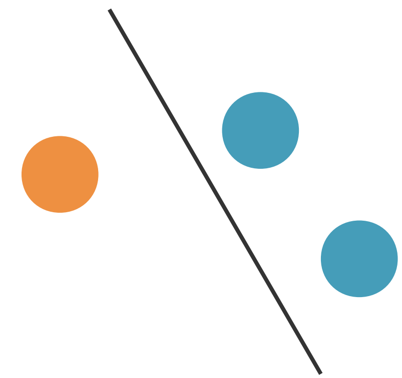
- Can a line shatter 2 points?



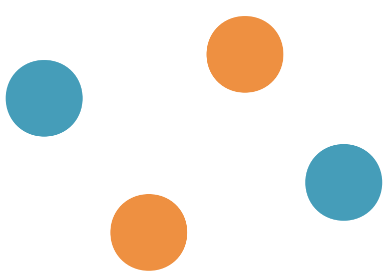
- Can a line shatter 3 points?





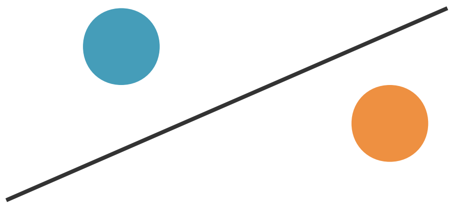


- Can a line shatter 4 points?

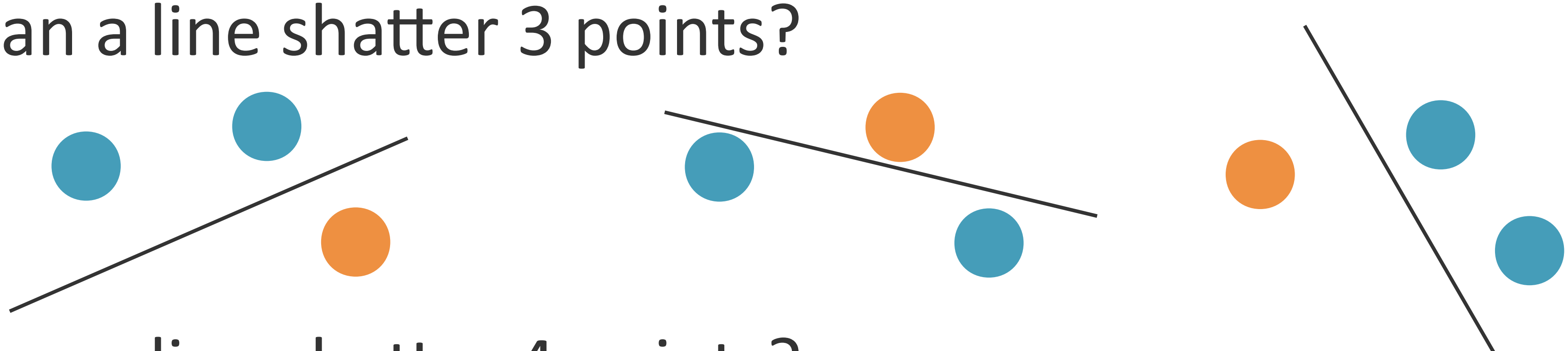


VC dimension of a line

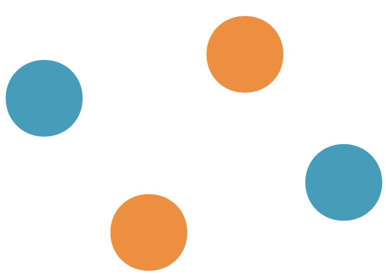
- Can a line shatter 2 points?



- Can a line shatter 3 points?



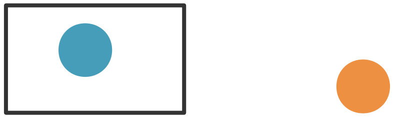
- Can a line shatter 4 points?



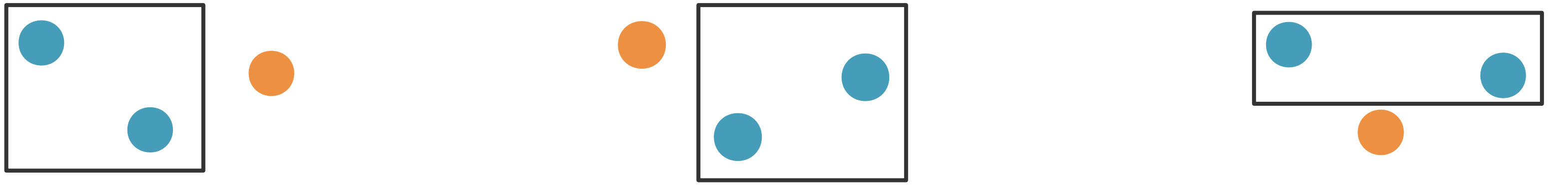
The VC dimension of a line is 3.

VC dimension of an axis-aligned rectangle

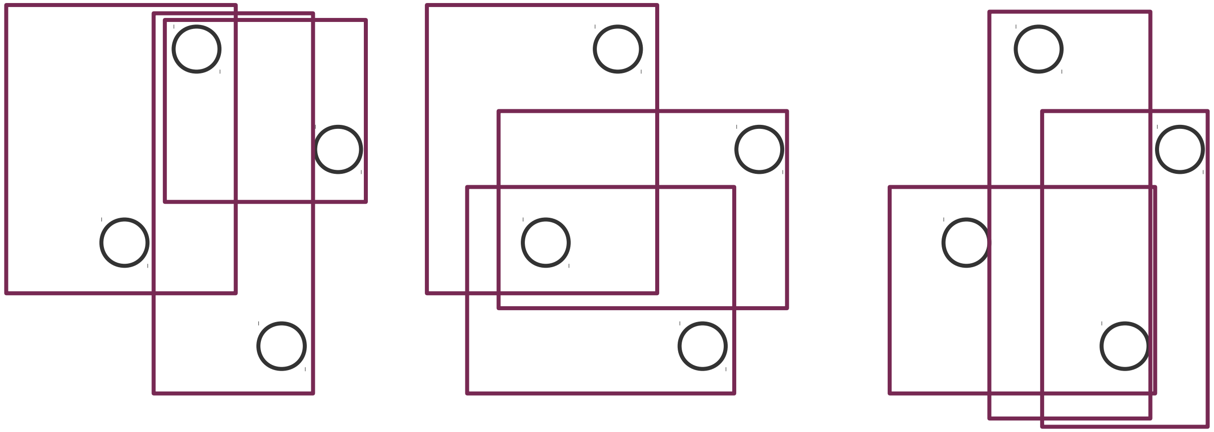
- Can an axis-aligned rectangle shatter 2 points?



- Can an axis-aligned rectangle shatter 3 points?



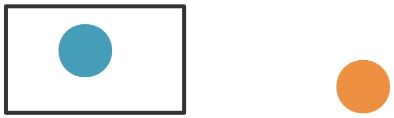
- Can an axis-aligned rectangle shatter 4 points?



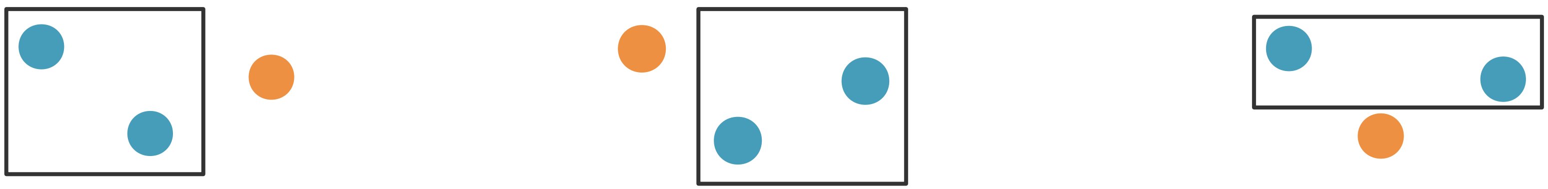
- Can an axis-aligned rectangle shatter 5 points?

VC dimension of an axis-aligned rectangle

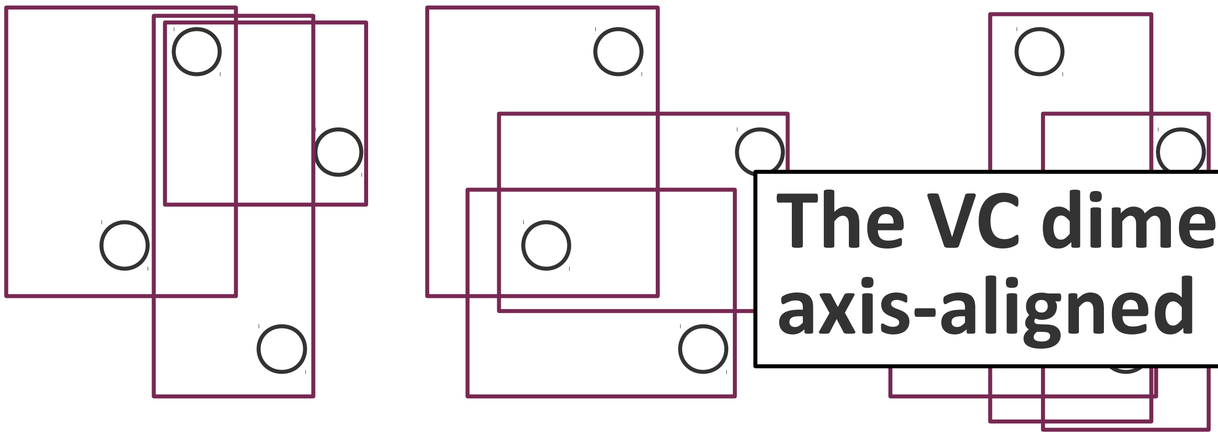
- Can an axis-aligned rectangle shatter 2 points?



- Can an axis-aligned rectangle shatter 3 points?



- Can an axis-aligned rectangle shatter 4 points?



The VC dimension of an axis-aligned rectangle is 4.

- Can an axis-aligned rectangle shatter 5 points?

VC dimension of an axis-aligned rectangle

- Using an axis-aligned rectangle, we can only guarantee learning classes over a world that contains no more than 4 data points.
- However, **the VC dimension is idp of the probability distribution of the data.**
  - the world changes smoothly
  - nearby instances have the same label most of the time
  - hence **we can still learn specific classes with H.**

Probably Approximately Correct Learning

PAC learning

Probably Approximately Correct learning

- We want f to be
  - **approximately correct**

    the probability of error is bounded by ε   (ε > 0)
  - **probably approximately correct**

    f is correct most of the time, i.e. with probability at least 1-δ

    (δ ≤ 1/2)

$$P\left(P(f(\boldsymbol{x}) \neq c(\boldsymbol{x})) \leq \epsilon\right) \geq (1-\delta).$$

[Vaillant, 1984]

PAC-learnable problem

- A hypothesis space H is **PAC-learnable** if there exists an algorithm that
  - Produces a probably approximately correct hypothesis
  - In polynomial time in 1/ε and in 1/δ
  - For any class C in H and any dataset D.

- **Sample complexity:** the number of instances N needed to learn it.

  Given a class C and examples drawn from a fixed probability distribution, we want to **find N such that**

$$P\left(P(f(\boldsymbol{x}) \neq c(\boldsymbol{x})) \leq \epsilon\right) \geq (1-\delta).$$

PAC learning of axis-aligned rectangles

- Let's consider f = S (tightest rectangle around positive examples)
- How many training examples N should we have, such that with probability at least (1 – δ), f has error at most ε ?

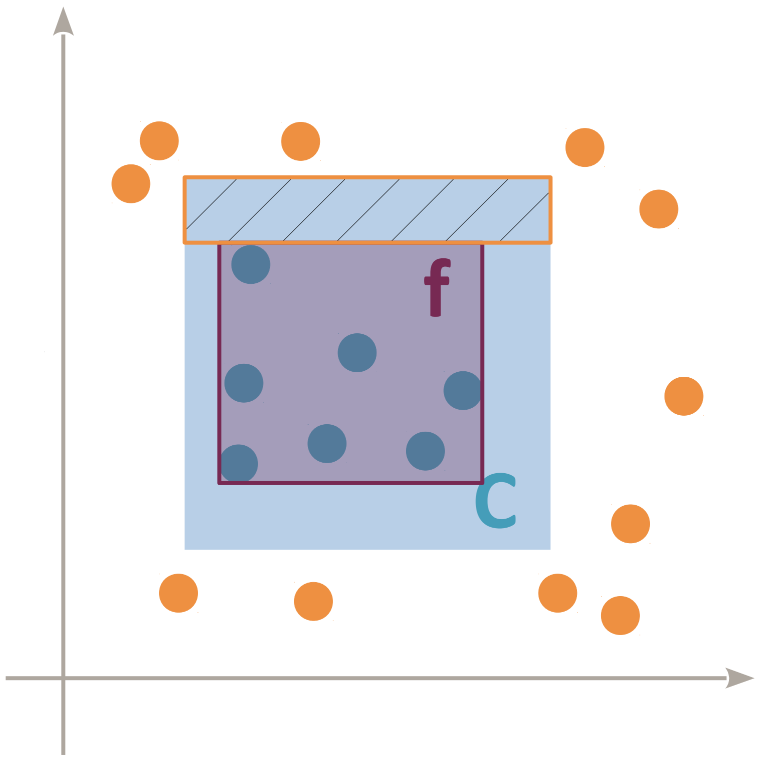
$$P\left(P(f(\boldsymbol{x}) \neq c(\boldsymbol{x})) \leq \epsilon\right) \geq (1-\delta).$$

- Let's show that **N ≥ (4/ε)log(4/δ)**
- If we want greater **accuracy** (ε↘)

  N must increase

- If we want greater **confidence** (δ↘)

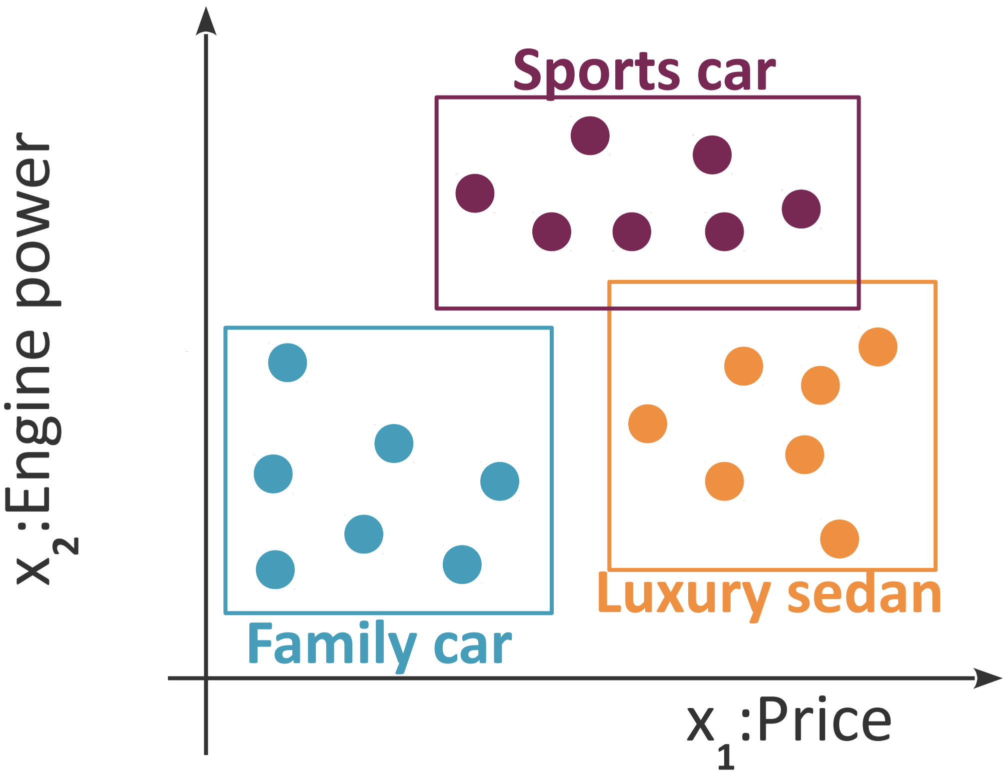
  N must increase



[Blumer et al., 1989]
[Kearns & Vazirani 1994]

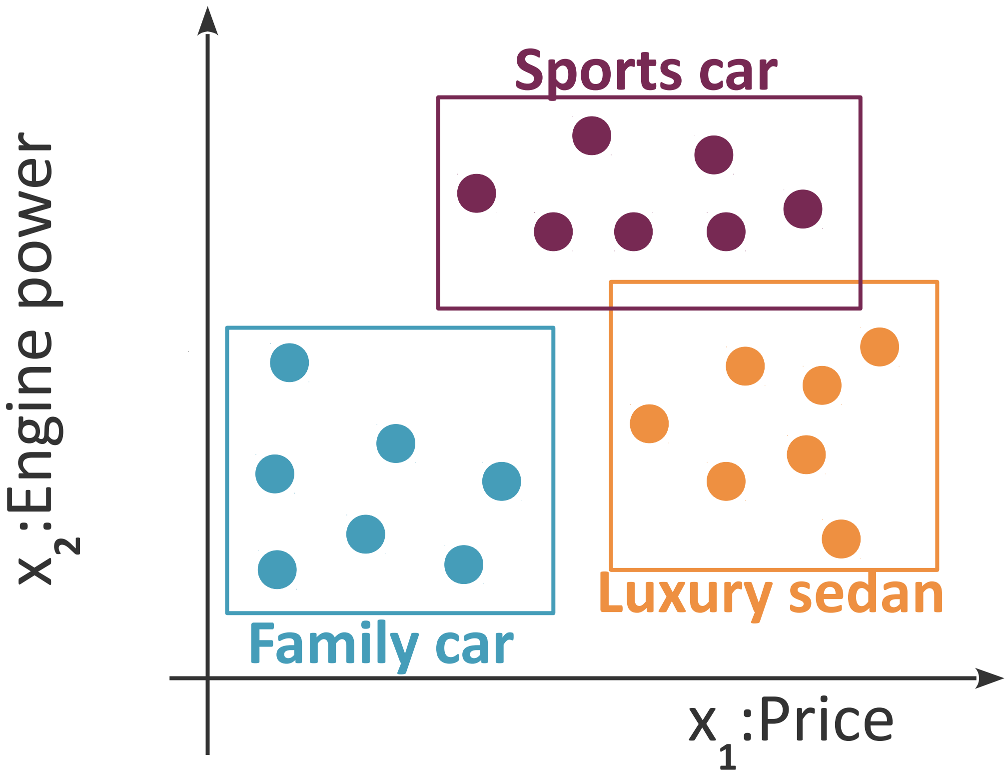
Binary classification isn't everything...

Multiple classes



How do we formulate this problem?

Multiple classes



$$y_k^i = \begin{cases} 1 & \text{if } \boldsymbol{x}^i \in \mathcal{C}_k \\ 0 & \text{if } \boldsymbol{x}^i \in \mathcal{C}_l, l \neq k \end{cases}$$

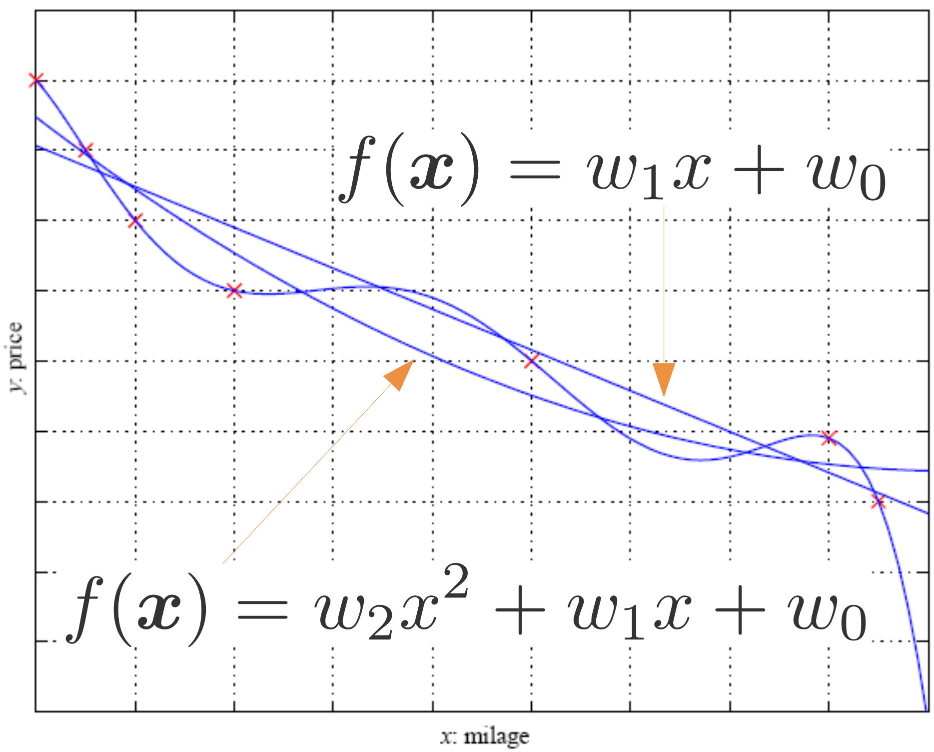
K hypotheses:

$$f_k(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f \text{ says } \boldsymbol{x} \in \mathcal{C}_k \\ 0 & \text{if } f \text{ says } \boldsymbol{x} \notin \mathcal{C}_k \end{cases}$$

Regression

$$\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\dots,n} \quad y^i \in \mathbb{R}$$

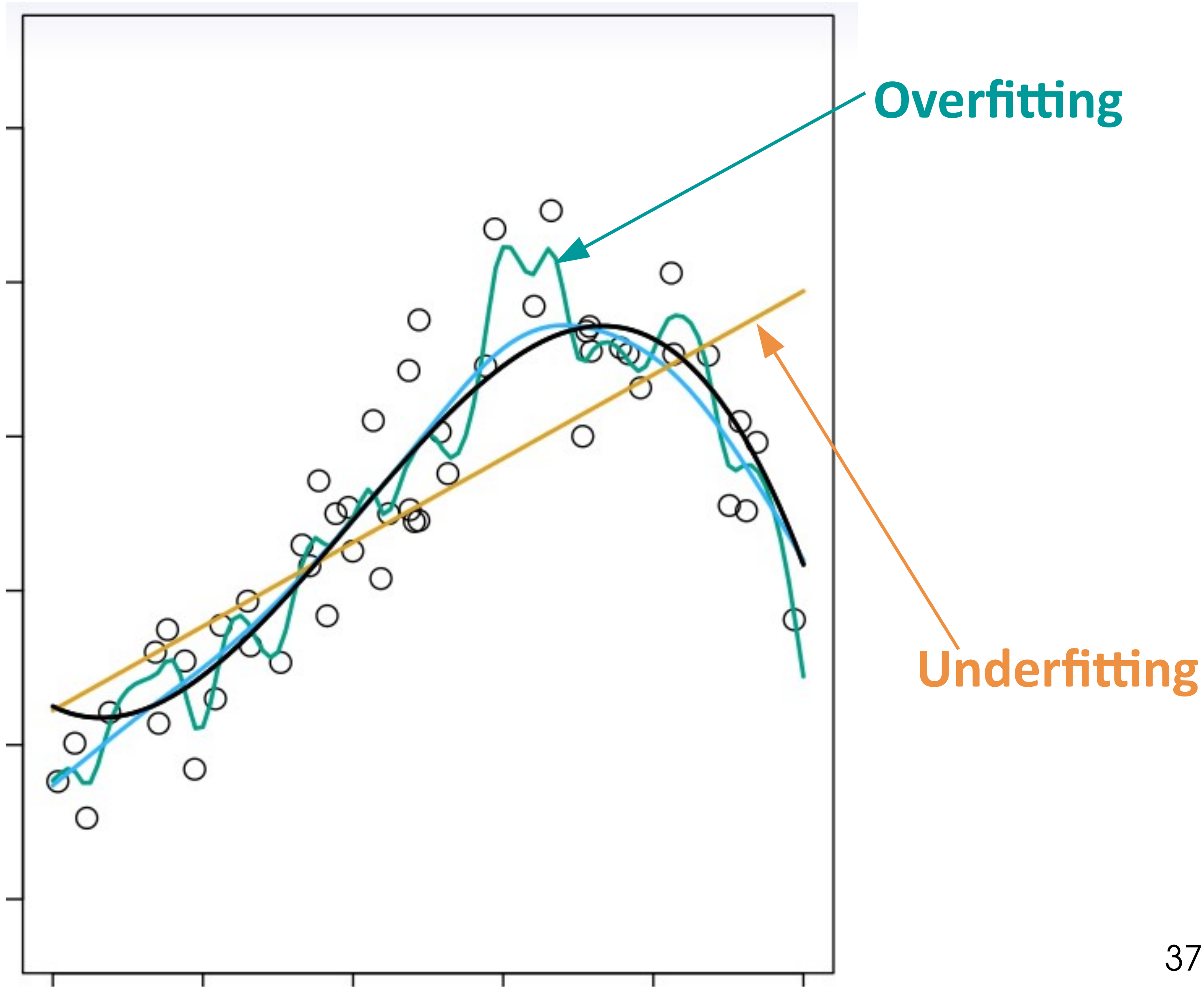
$$y^i = f(\boldsymbol{x}^i) + \epsilon$$



Empirical error:

$$E(f|X) = \frac{1}{n}\sum_{i=1}^{n}\left(y^i - f(\boldsymbol{x}^i)\right)^2$$

Overfitting & Underfitting (Regression)



Summary: 3 aspects of a supervised learner

Given an iid sample X={$\mathbf{x}^i$, y^i}, i=1...n,

our goal is to build a good and useful approximation to y.

- **Model**

  Define the hypothesis class $$f(\boldsymbol{x}|\theta) \in \mathcal{F}$$

- **Loss function** L

  Empirical error $$E(\theta|X) = \sum_{i=1}^{n} L(y^i, f(\boldsymbol{x}^i|\theta))$$

- **Optimization procedure** to minimize the empirical error

$$\theta^* = \arg\min_{\theta} E(\theta|X)$$

Before Sep. 14

- Do your **homework**

  http://tinyurl.com/ma2823-2016

  - Problem set to hand in
  - Set up your laptop for the lab.
- **Sign up to be a scribe**

  https://framadate.org/omVzzIPfaHHgm881