CS 340: Machine Learning
Lecture 3: Introduction to Supervised Learning

AD

January 2011

Supervised learning as function fitting

- We are given some training data

$$\mathcal{D} = \left\{ \left(\mathbf{x}^i, y^i \right) \right\}_{i=1}^{N}$$

- Consider a restricted set of mappings/parametric functions f in *hypothesis class* $\mathcal{H}$

$$f \in \mathcal{H} : \mathcal{X} \times \quad \Theta \quad \longrightarrow \mathcal{Y},$$

  we will predict using

$$\widehat{y}\left(\mathbf{x}\right) = f\left(\mathbf{x}; \theta\right)$$

  where $\theta \in \Theta$.

- **Learning**: Given $\mathcal{H}$, learn parameters θ given $\mathcal{D}$ so that predictions on non-labeled inputs (i.e. test set, real-world data) are as accurate as possible.

Definitions

- **Empirical/Training error**:

$$\text{Err\_Train} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(\widehat{y}\left(\mathbf{x}^i\right) \neq y^i\right).$$

- **Test error**:

$$\text{Err\_Test} = \frac{1}{N\_\text{test}}\sum_{i=1}^{N\_\text{test}} \mathbb{I}\left(\widehat{y}\left(\mathbf{x}^i_{\text{test}}\right) \neq y^i_{\text{test}}\right).$$

- **Generalization error:**

$$\mathbb{P}\text{roba}\left[\text{error}\right] = \mathbb{E}\left[\mathbb{I}\left(\widehat{y}\left(\mathbf{x}\right) \neq y\right)\right] = \int \mathbb{I}\left(\widehat{y}\left(\mathbf{x}\right) \neq y\right)\ p\left(\mathbf{x},y\right) d\mathbf{x}dy$$

  where the expectation is w.r.t the UNKNOWN probability density function $p\left(\mathbf{x},y\right)$ of $\left(\mathbf{x},y\right)$.

- Empirical/Training error are approximations of generalization error thanks to the Law of Large Numbers.

Law of Large Numbers

- Assume that some $\mathcal{Z}$-valued random variables $\left\{z^i\right\}$ are independent and identically distributed according to a probability density function $p(z)$ then for any function $g : \mathcal{Z} \to \mathbb{R}$ the law of large numbers states that

$$\lim_{N\to\infty} \underbrace{\frac{1}{N}\sum_{i=1}^{N} g\left(z^i\right)}_{\text{empirical average}} = \underbrace{\mathbb{E}\left[g(z)\right]}_{\text{statistical average}}$$

  where the expectation $\mathbb{E}\left[g(Z)\right]$ is defined by

$$\mathbb{E}\left[g(z)\right] = \int g(z)\, p(z)\, dz.$$

- This fundamental theorem states that the limit of the empirical average converges to the statistical average.
- In the supervised learning case, we have simply $z = (\mathbf{x}, y)$ and $g(z) = g(\mathbf{x}, y) = \mathbb{I}\left(\widehat{y}(\mathbf{x}) \neq y\right)$.

Binary classification: Credit card scoring

- Say you have training data of the following form

Income	Savings	Risk
100	50	Hi
100	100	Lo
50	75	Hi
500	93	Lo

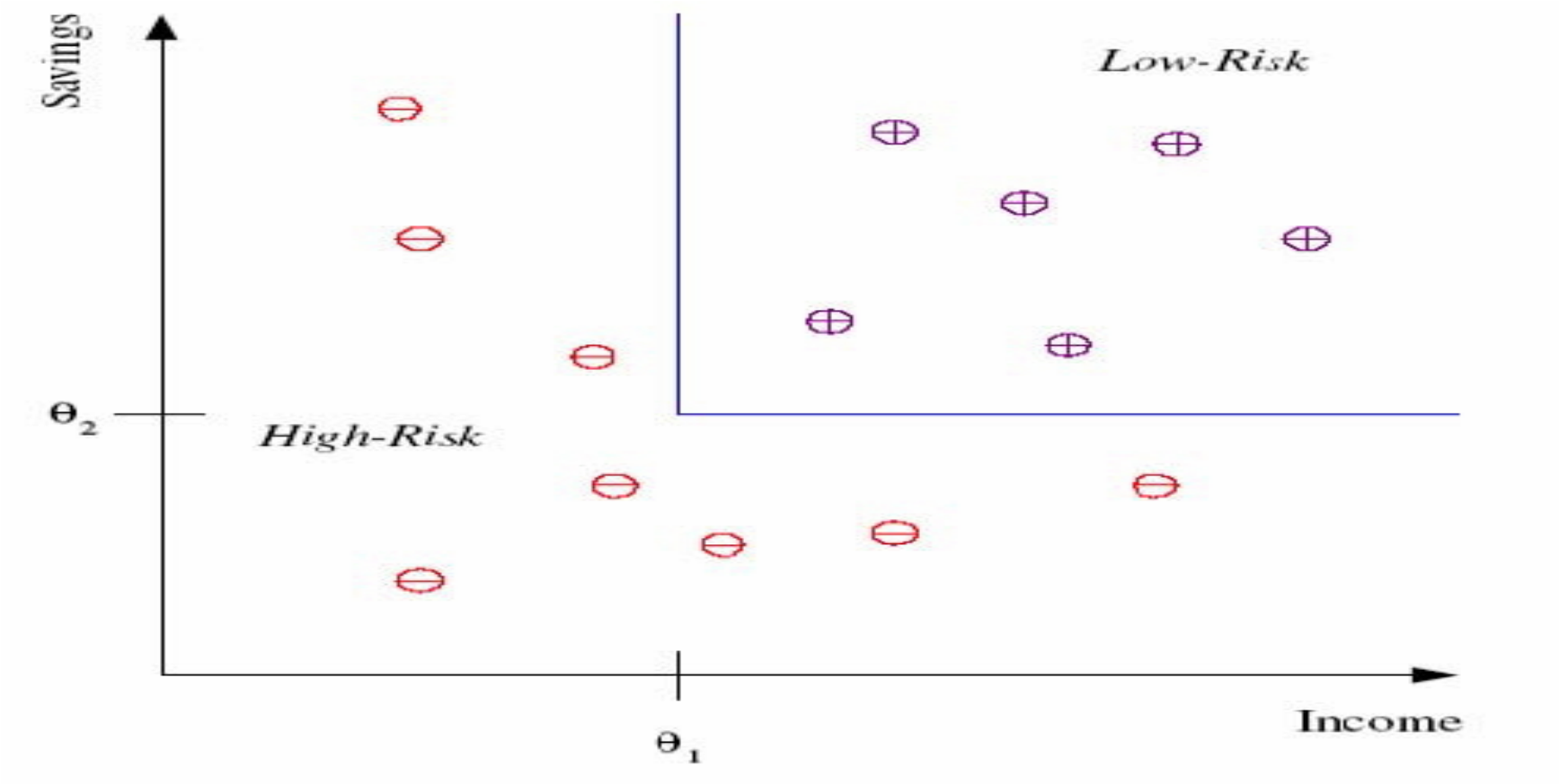
- Test data are of the form

Income	Savings	Risk
98	49	?
100	102	?
400	20	?

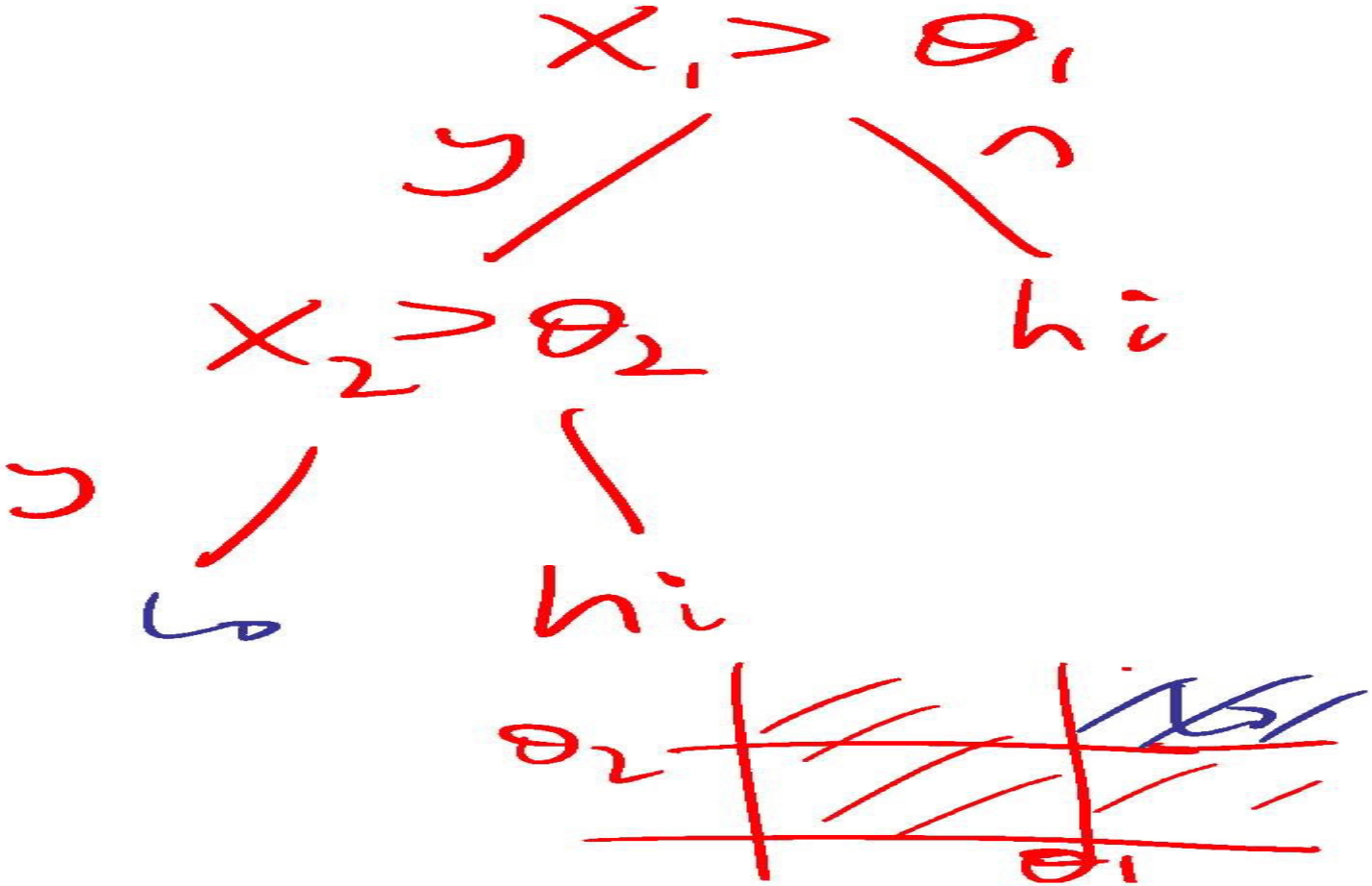
- In this case $\mathbf{x} = (x_1, x_2) \in (\mathcal{X} = \mathbb{R}^2)$ and $\mathcal{Y} = \{\text{high,low}\}$.

Example function

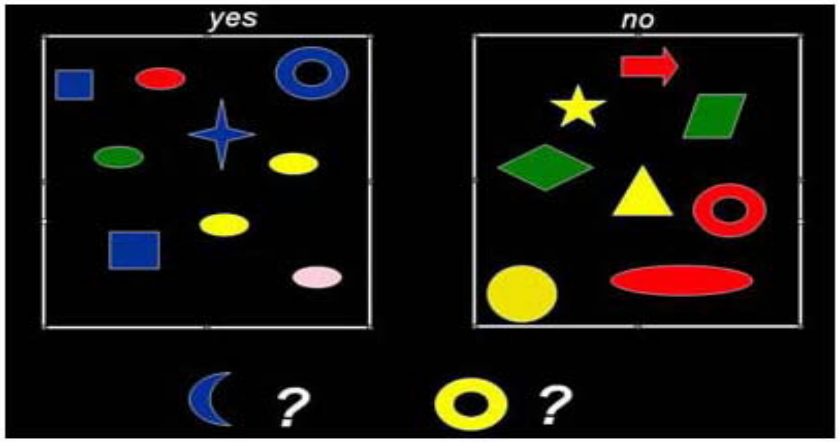
- $f(\mathbf{x};\theta) = f((\text{income,savings});(\theta_1,\theta_2)) = \text{IF } (x_1 = \text{income}) > \theta_1$ AND $(x_2 = \text{savings}) > \theta_2$ THEN low-risk ELSE high-risk.



Decision Trees



Classifying Shapes



p features (attributes)

Training set:
X: n by p
y: n by 1

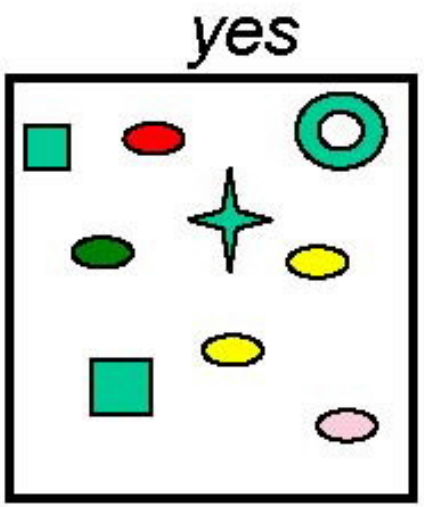
n cases

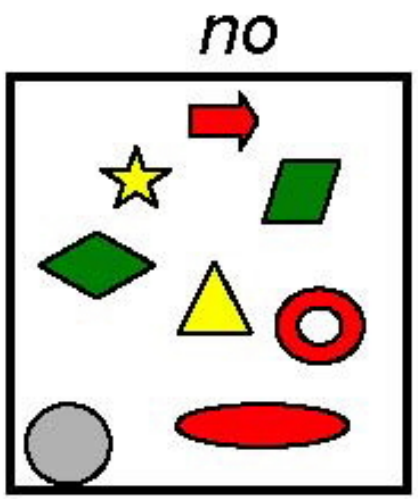
Color	Shape	Size	Label
Blue	Square	Small	Yes
Red	Ellipse	Small	Yes
Red	Ellipse	Large	No

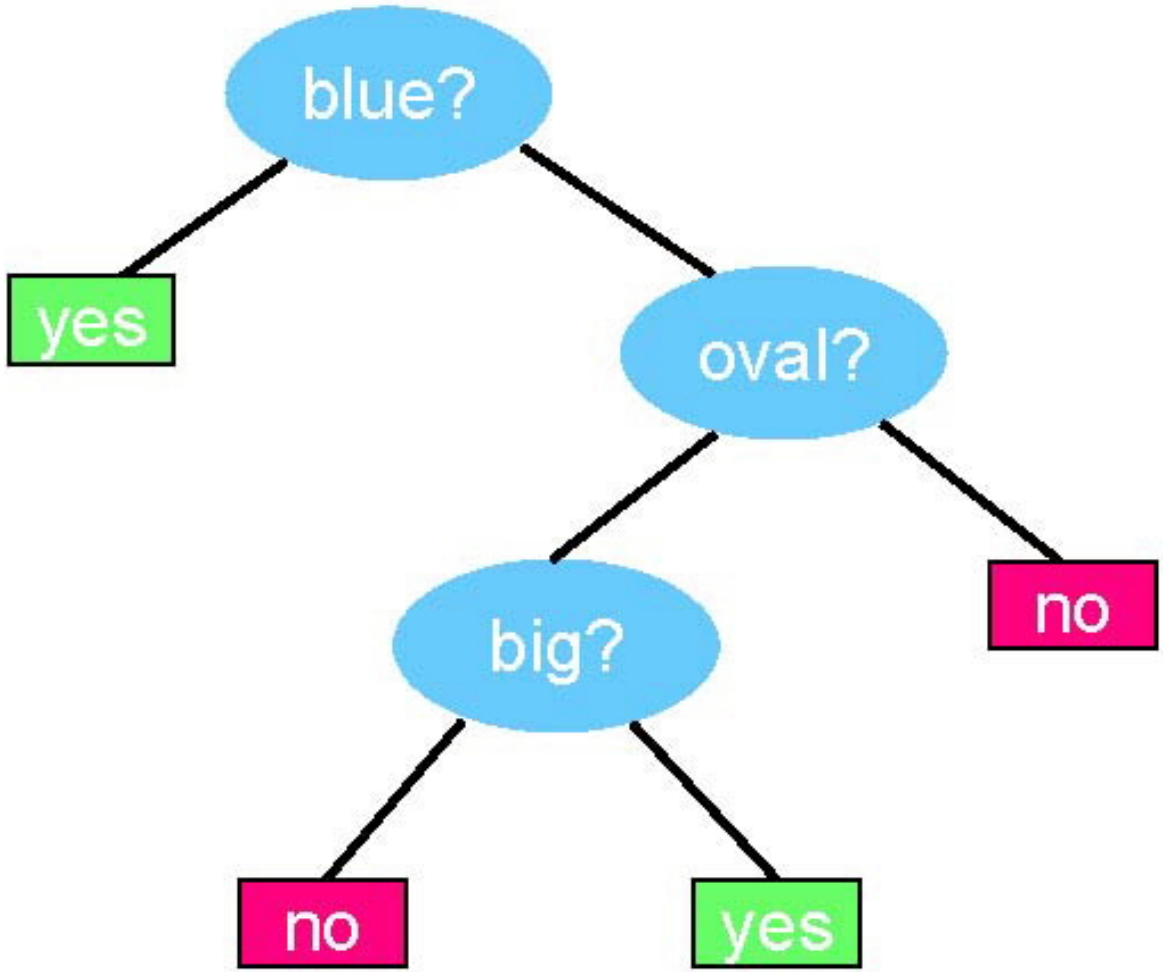
Test set

Blue	Crescent	Small	?
Yellow	Ring	Small	?

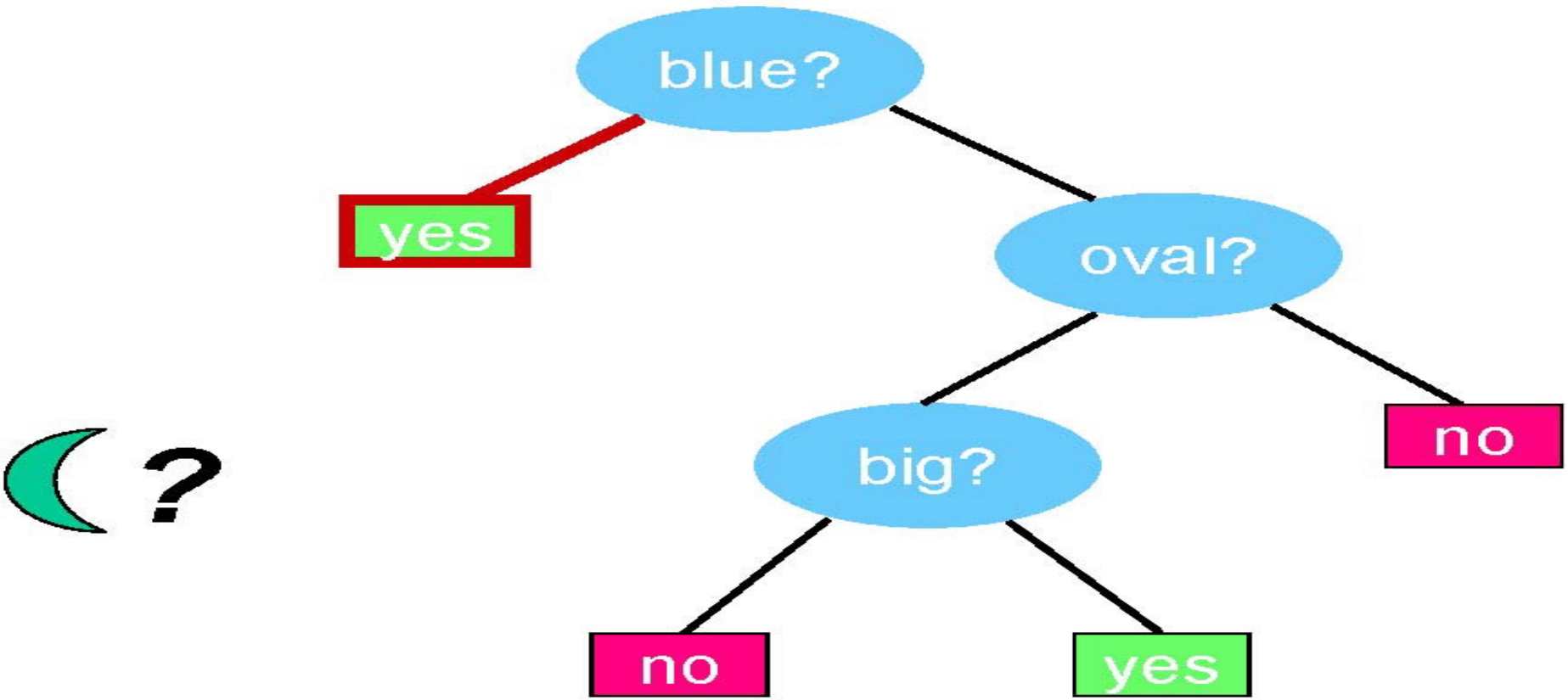
Hypothesis (Decision Trees)





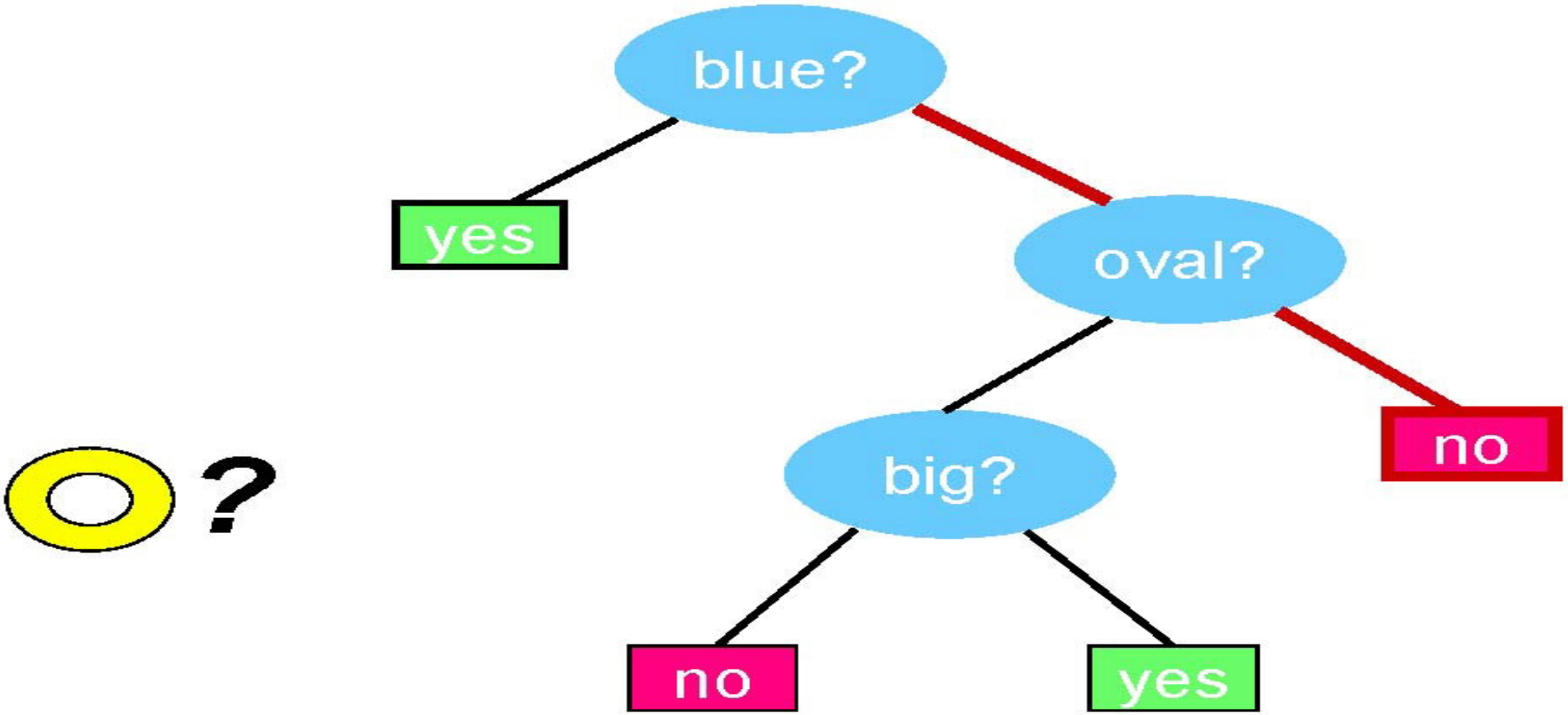


Decision Trees



- Generalization beyond training set can be difficult: Blue $\rightarrow$ yes: Reasonable?

Decision Trees



- Generalization beyond training set can be difficult: Yellow small ring $\longrightarrow$ no: Reasonable?

Decision Trees

- Decision trees are easy to use and easy to understand, and they can easily handle mixed discrete and continuous inputs.
- They can be very difficult to build (NP hard even in some simple cases).
- Optimal tree structure is sensitive to the particular patterns in the training data: not very stable to perturbations.
- Adding one more example of a certain kind can change the whole tree topology and the model does not generalize very well to novel inputs.
- No longer very popular in machine learning.

What's the right hypothesis class?

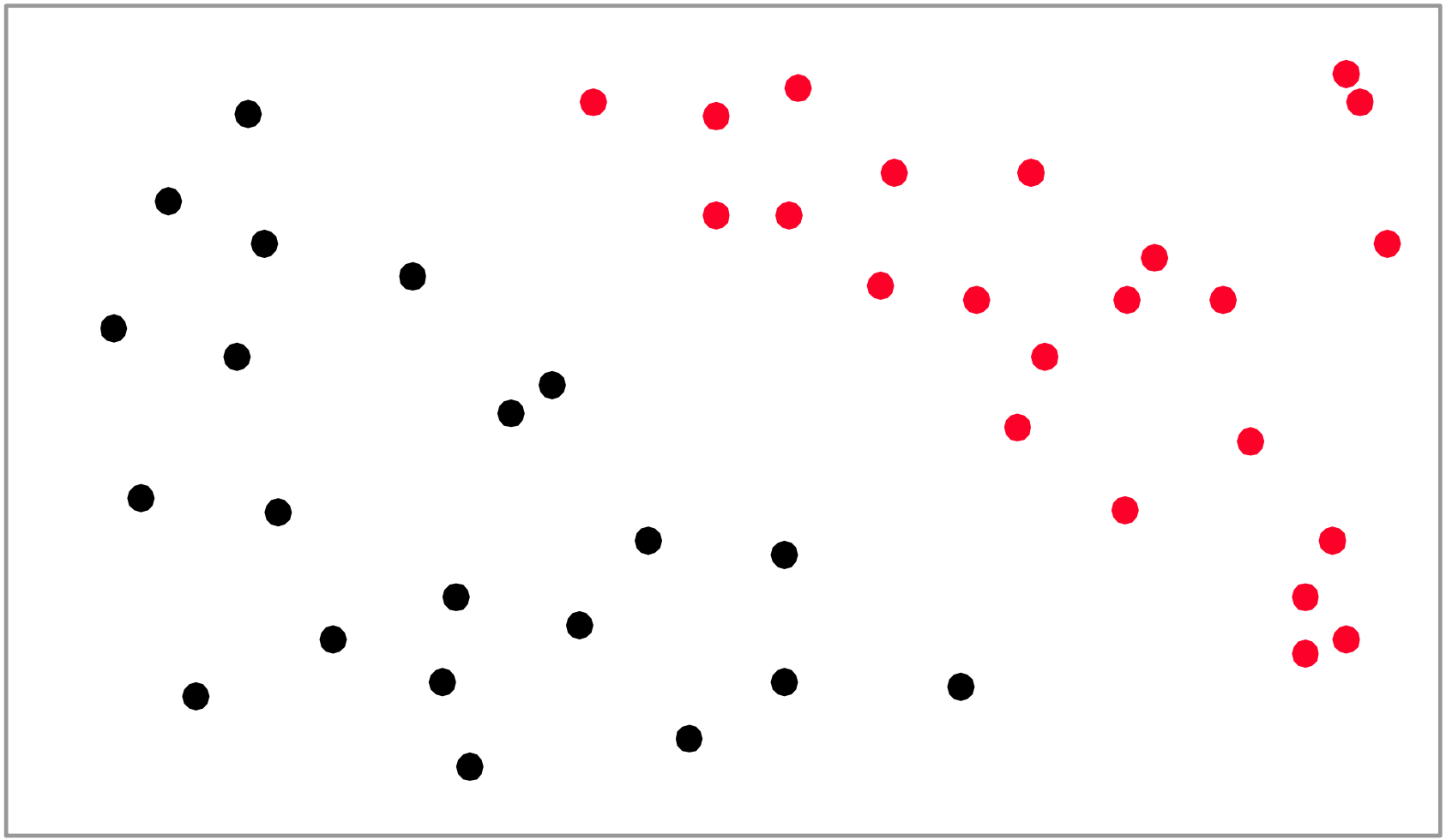


Figure: Training data where $\mathbf{x}^i \in \mathbb{R}^2$ and $y^i \in \{-1, 1\}$

Linearly separable data

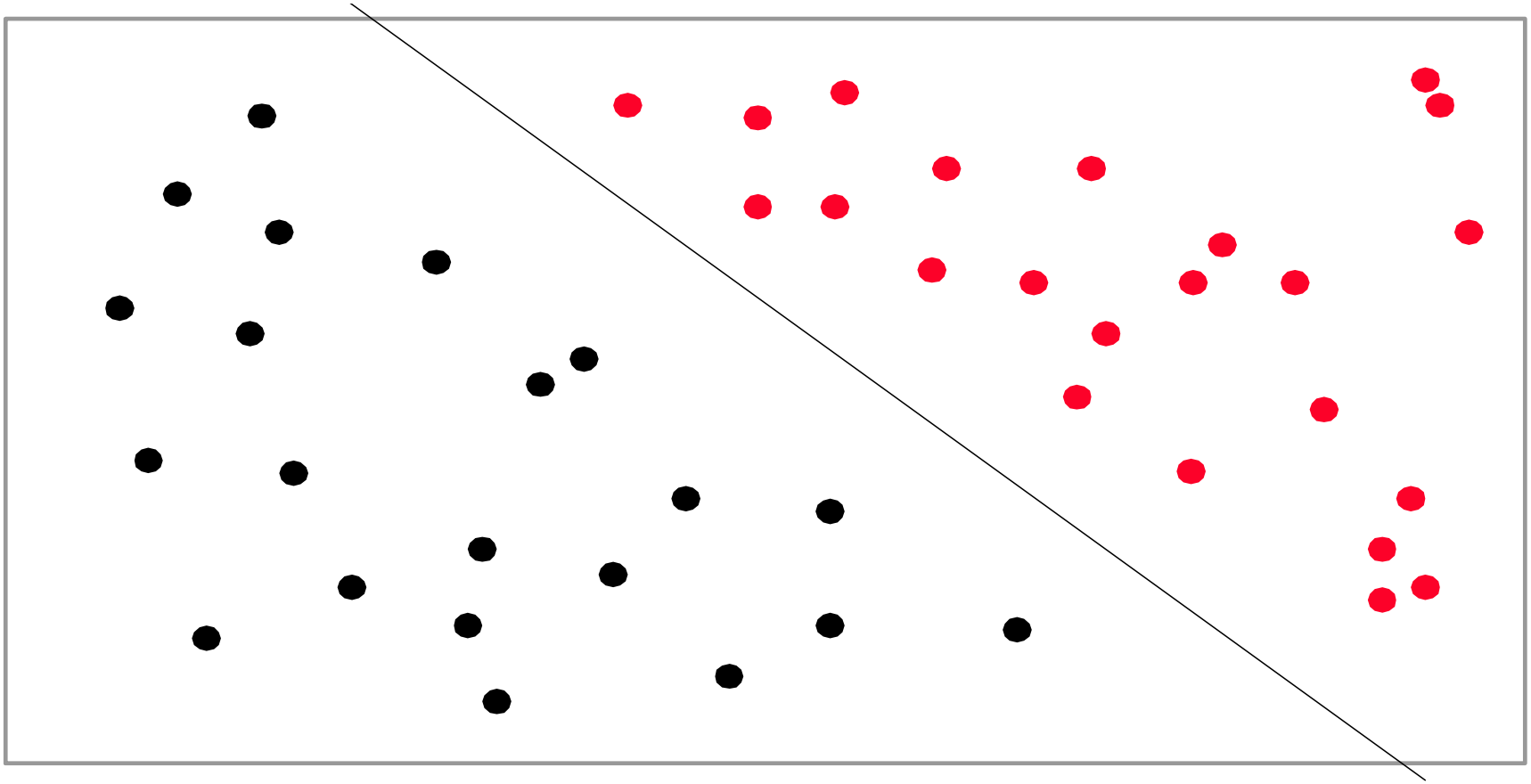
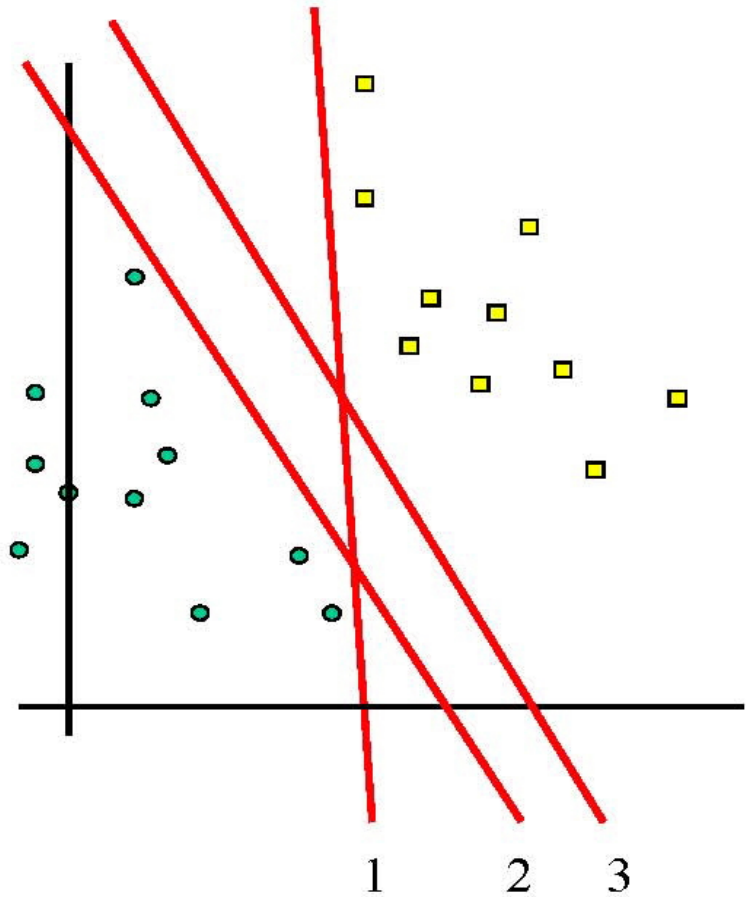
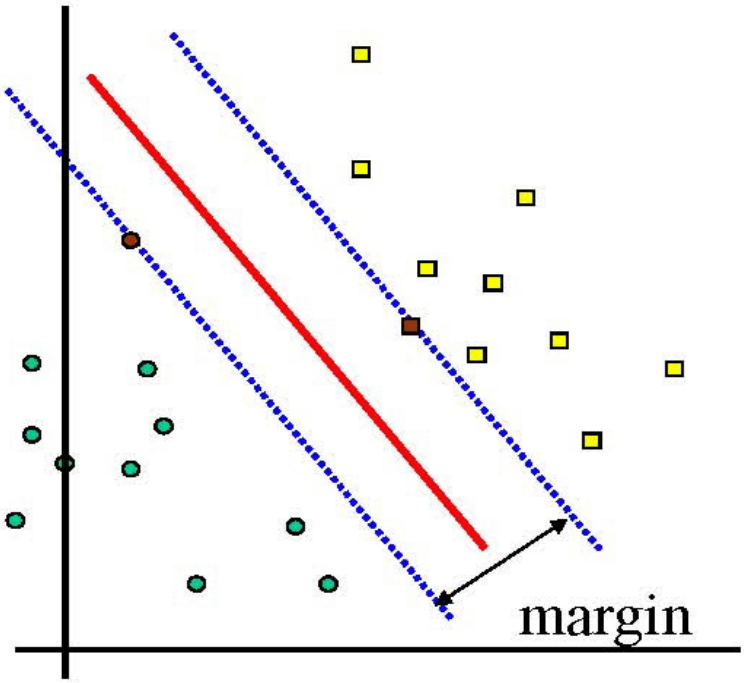


Figure: Linear classifier where $\mathbf{x}^i \in \mathbb{R}^2$ and $y^i \in \{-1, 1\}$

- Linearly separable means if f is a "linear" function of $\mathbf{x}$, we can perfectly fit the training data.

Which linear hypothesis is better?





- Even if the chosen hypothesis class is properly chosen, we will need to specify a criterion to select θ!

Not linearly separable data

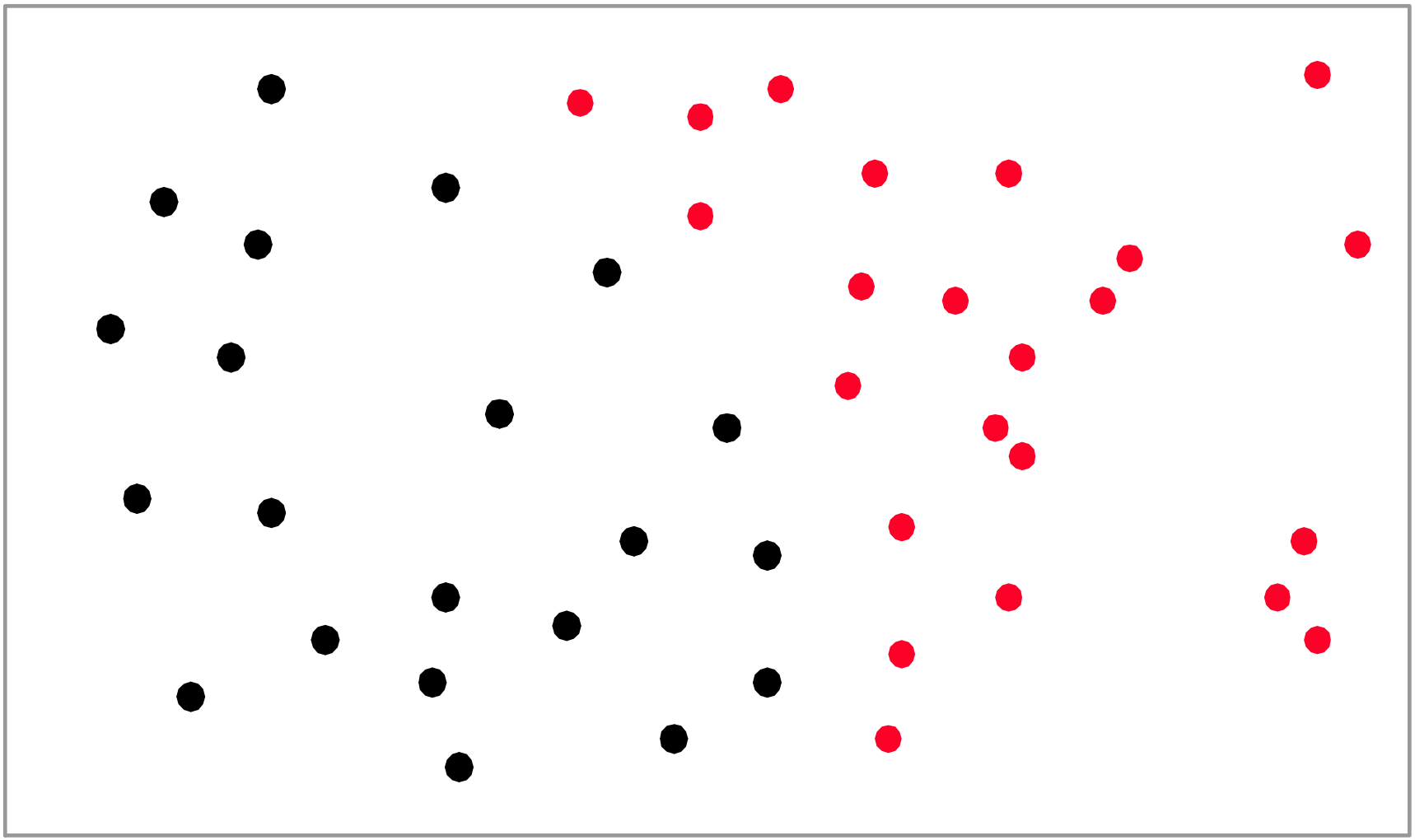
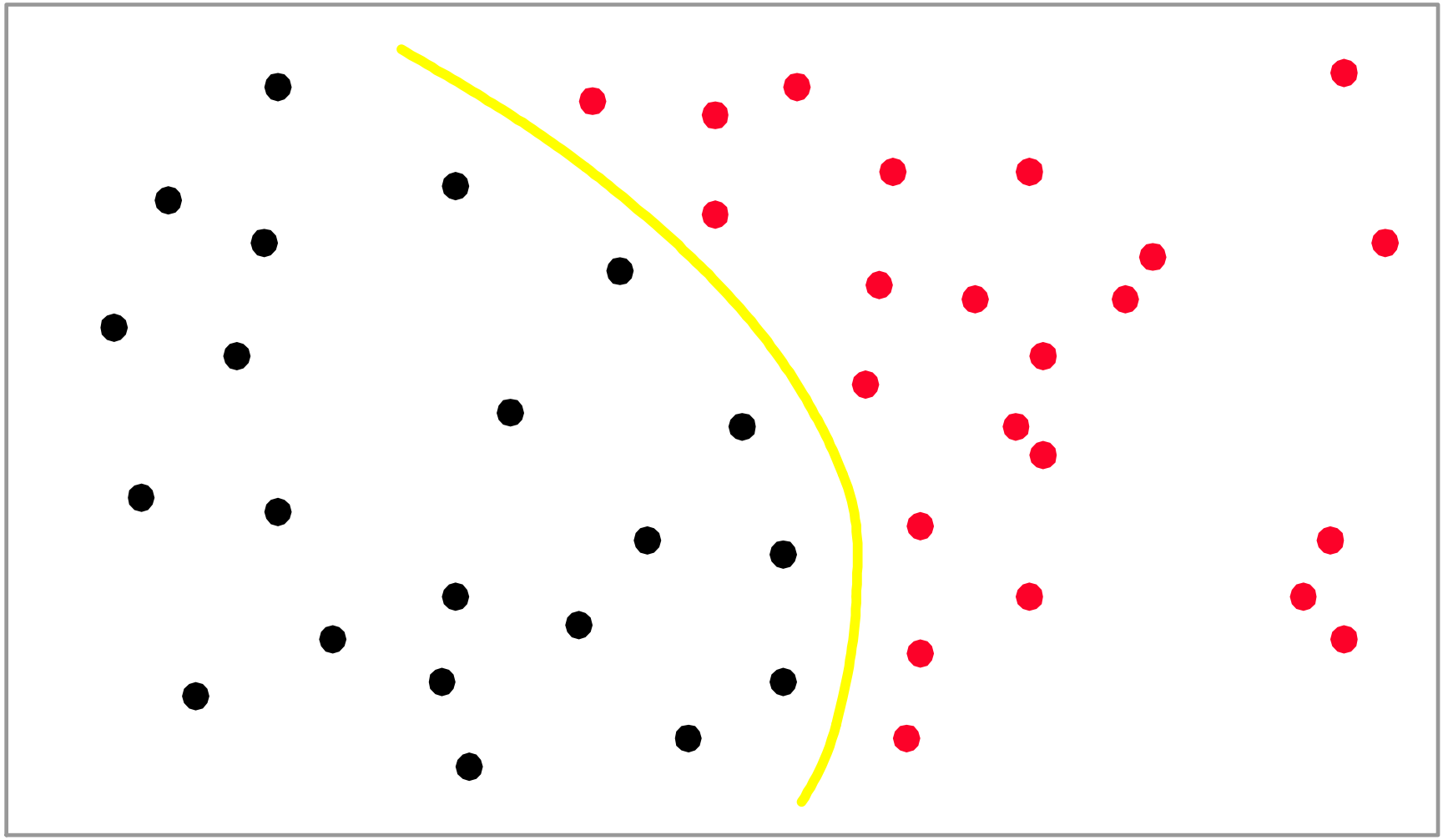


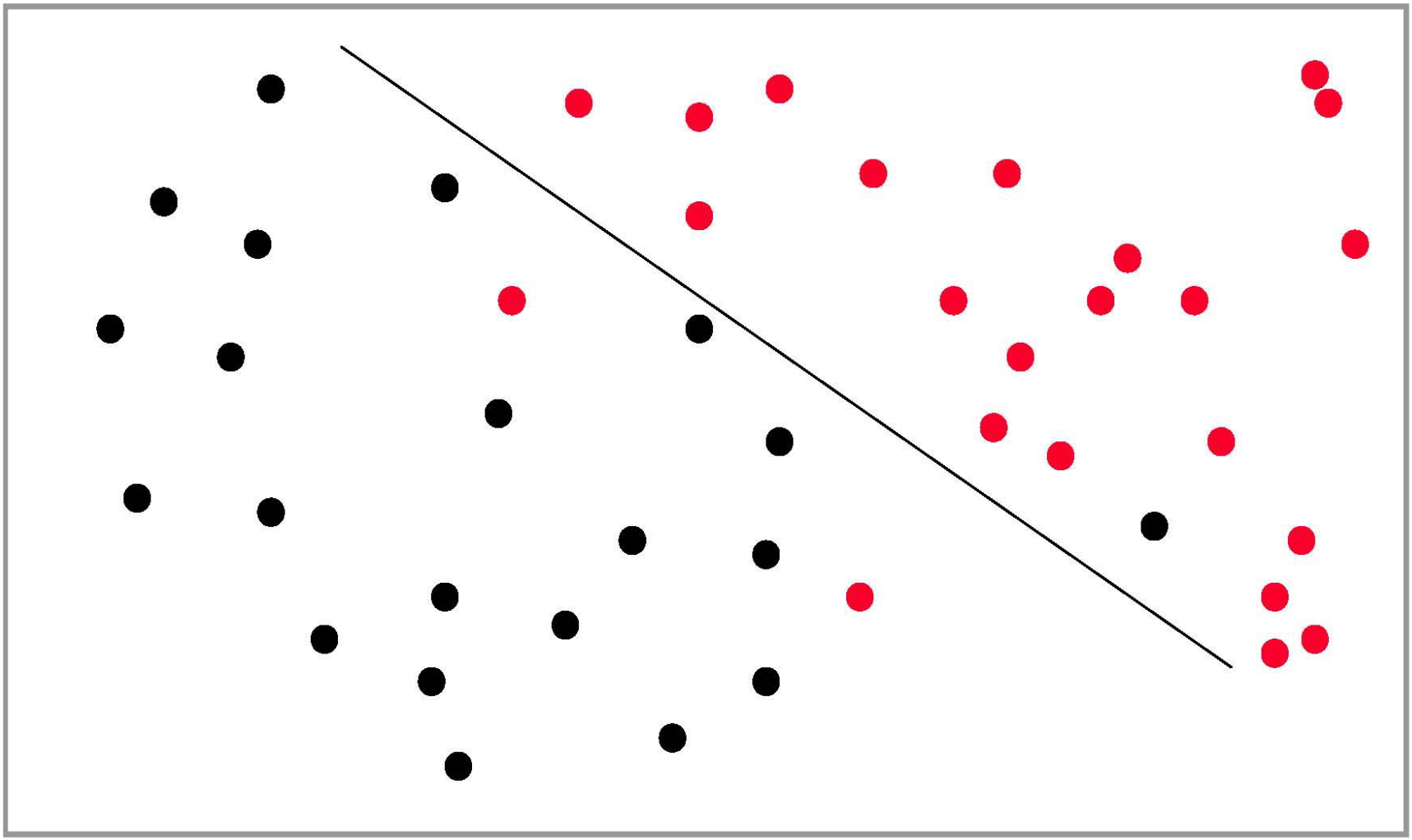
Figure: Non-linearly separable training data where $\mathbf{x}^i \in \mathbb{R}^2$ and $y^i \in \{-1, 1\}$

Quadratically separable

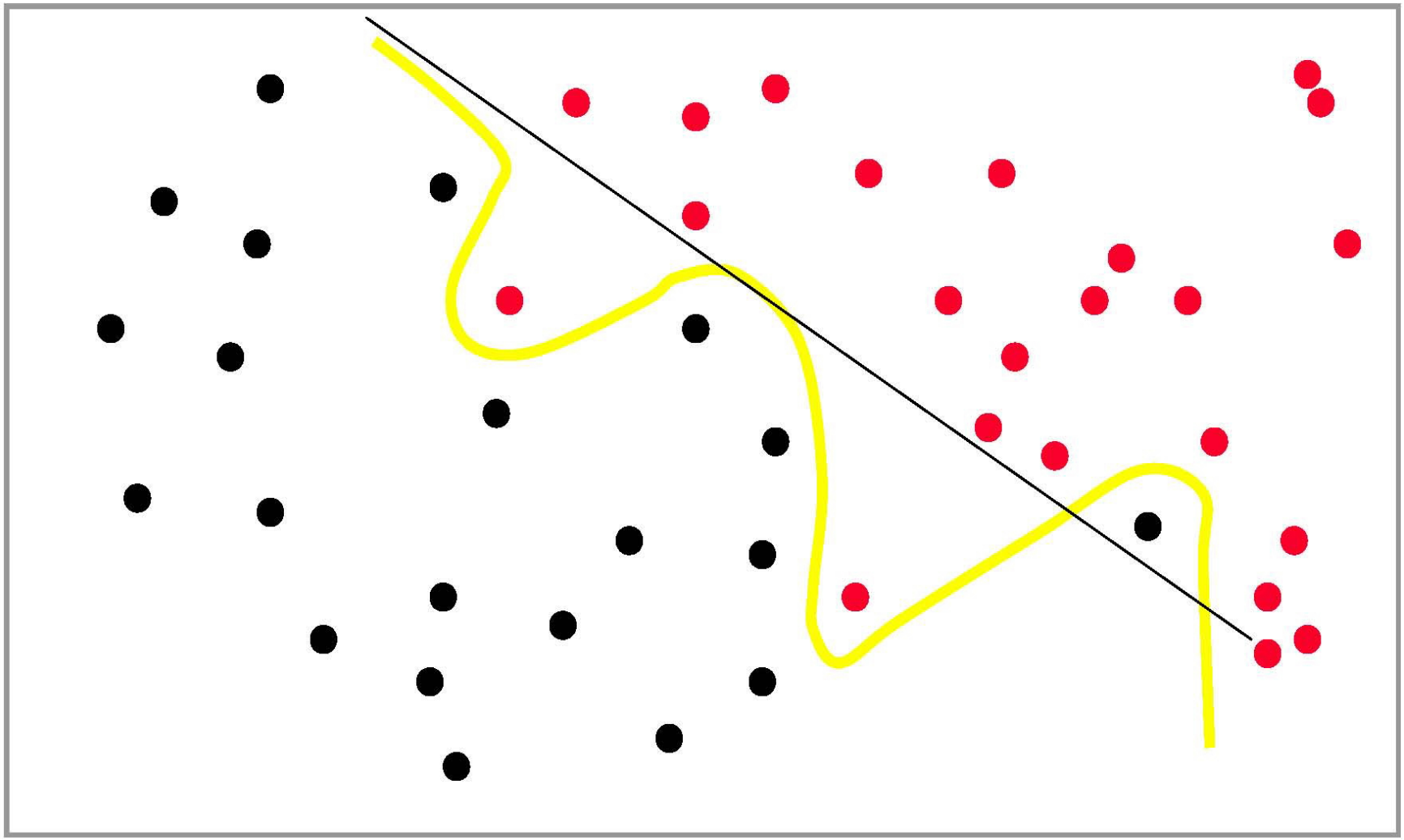


- $f\left(\mathbf{x};\theta\right) = \text{sgn}\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2\right)$.

Noisy/Mislabeled data and/or Features non-informative

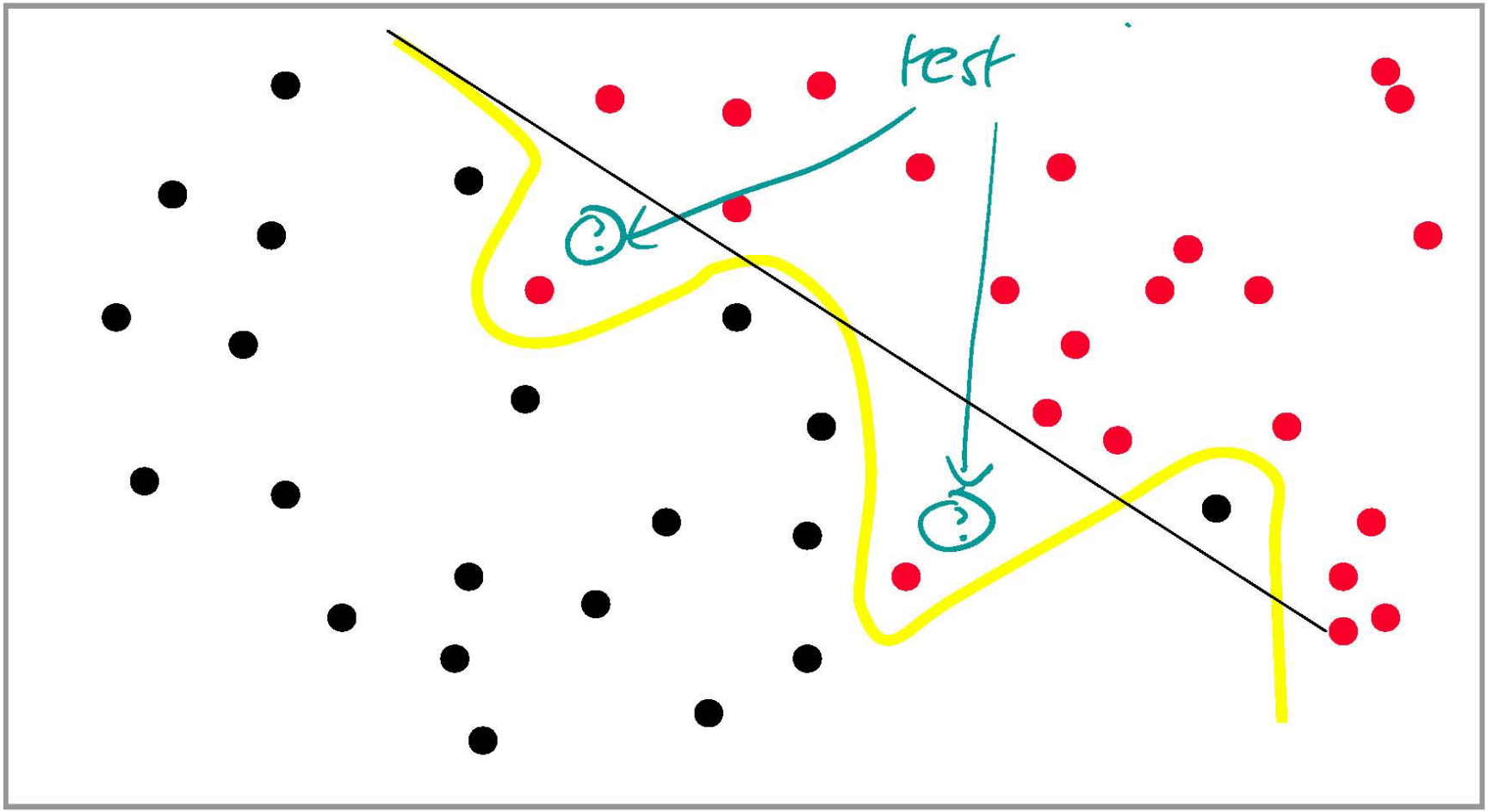


Overfitting

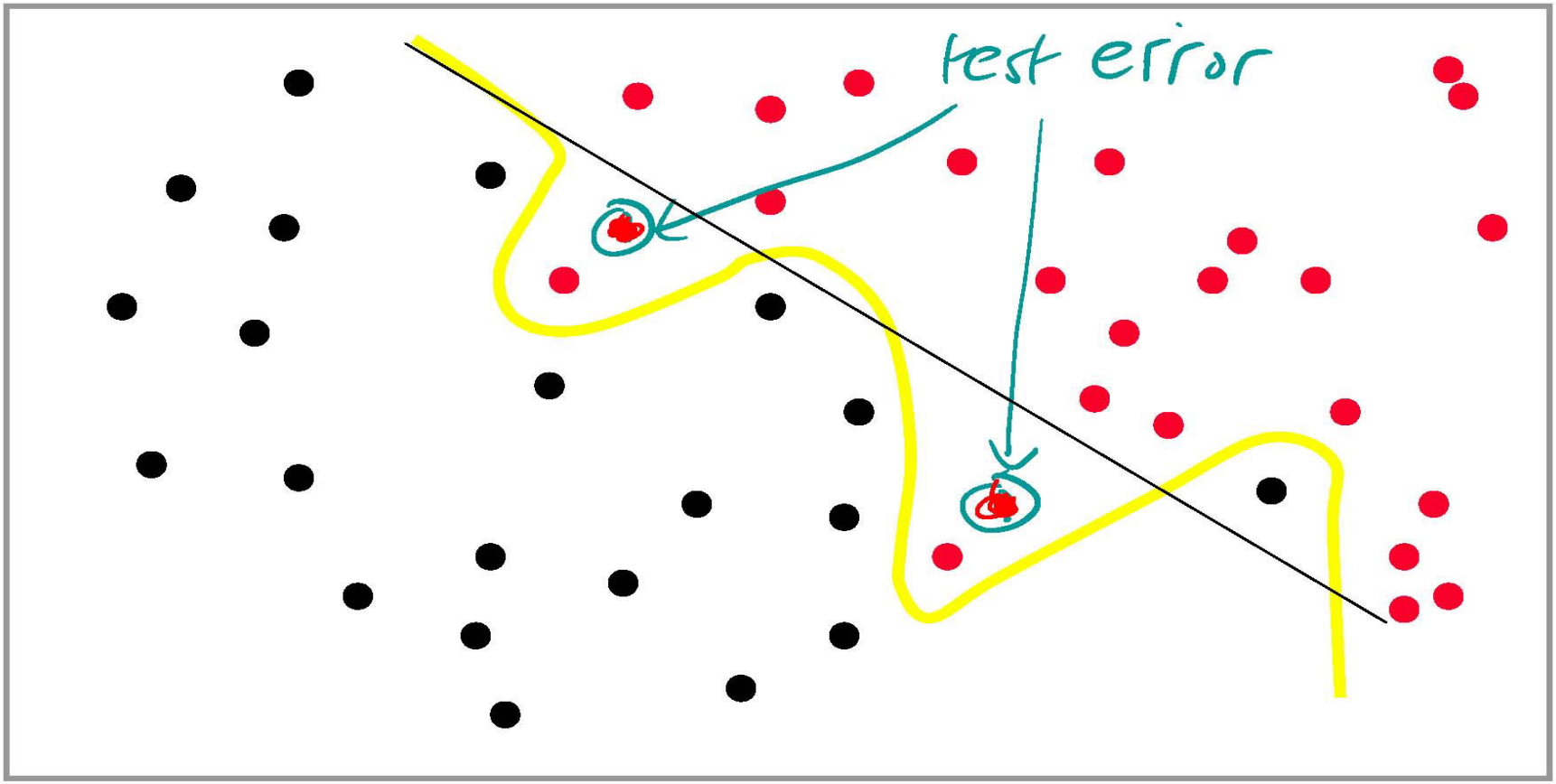


- An overly flexible function might models irrelevant details of training set.

Overfitted functions do not predict well

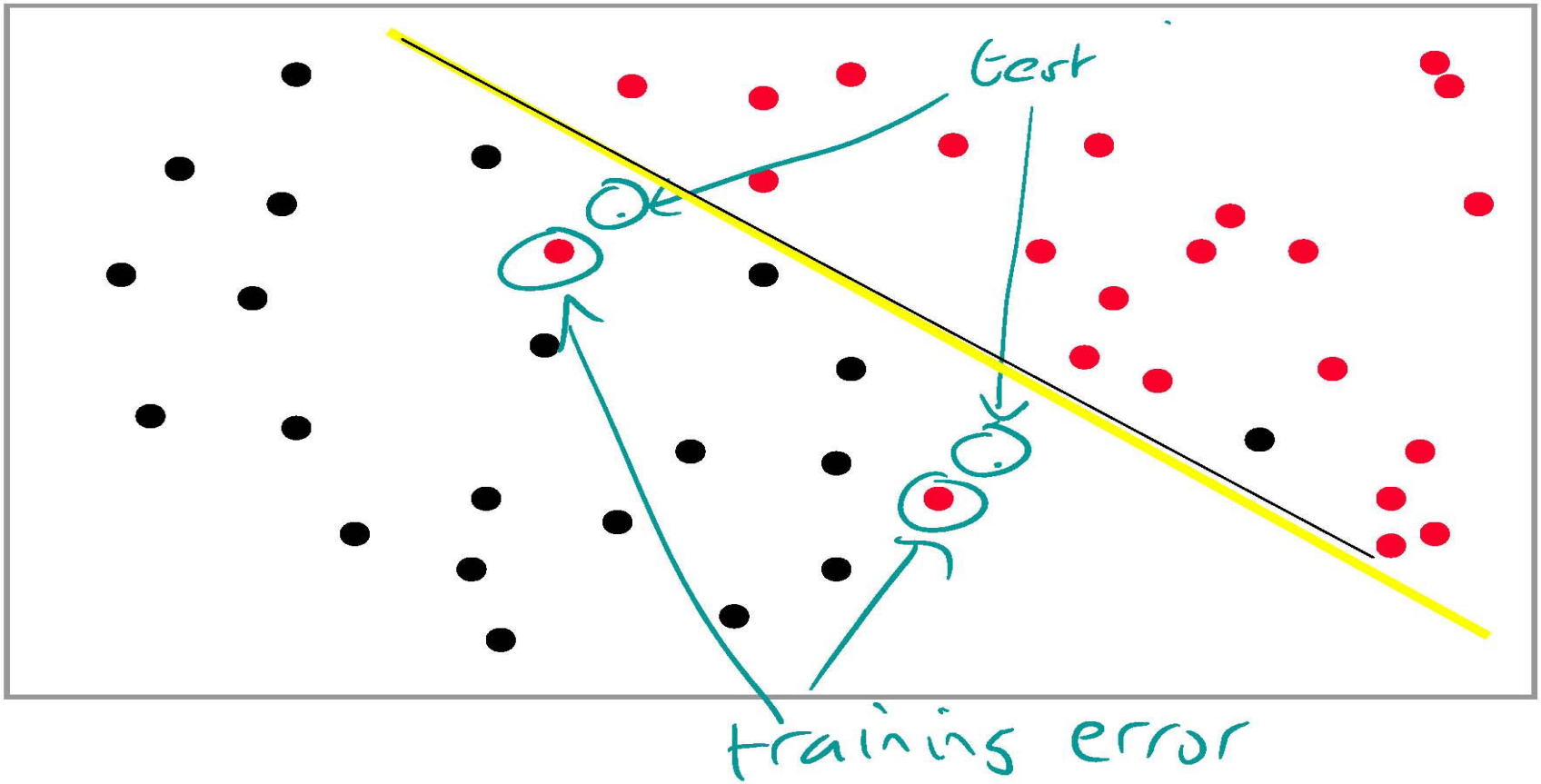


Overfitted functions do not predict well

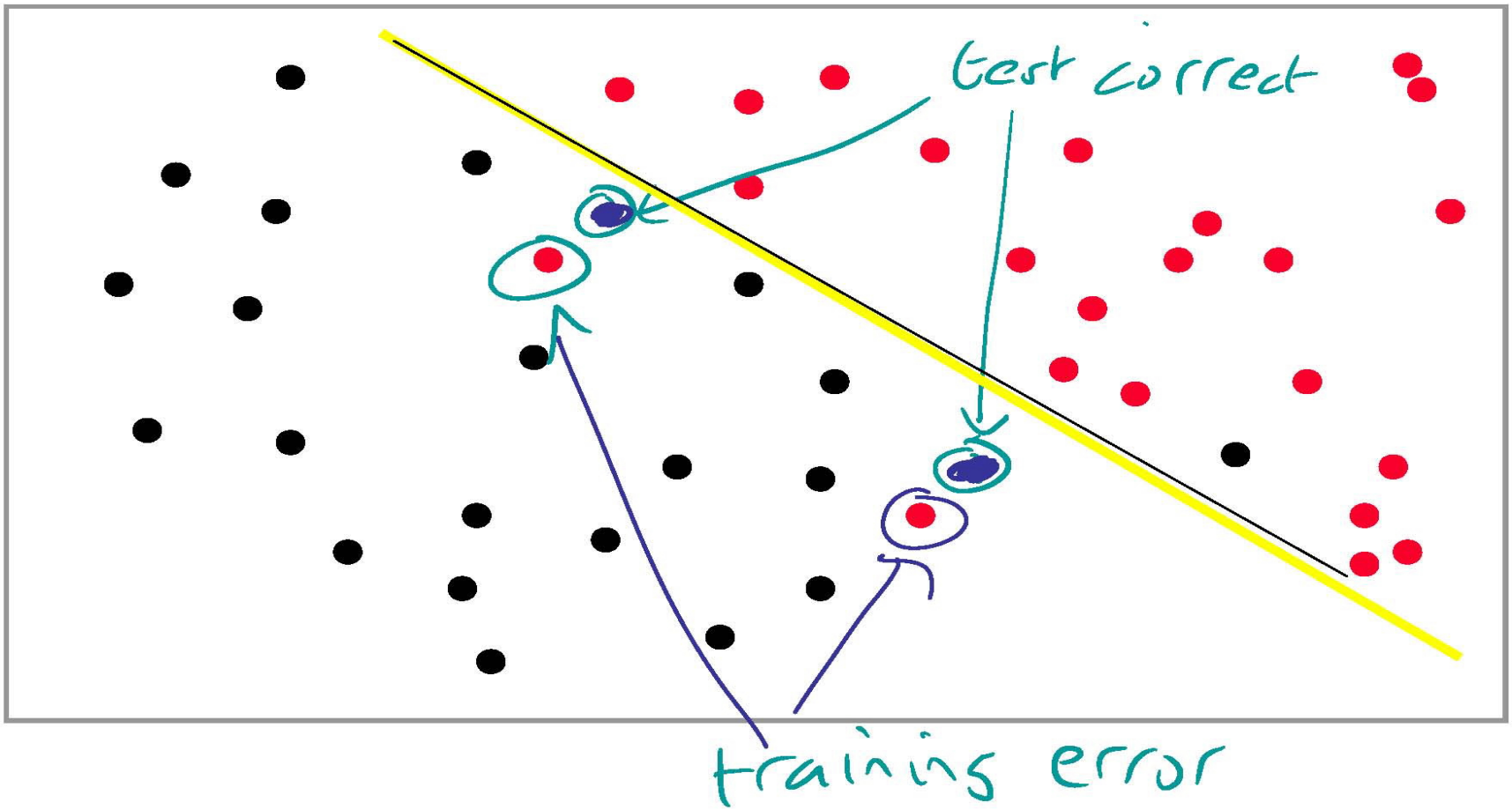


Test points are mis-predicted

Tradeoff simplicity for model fit

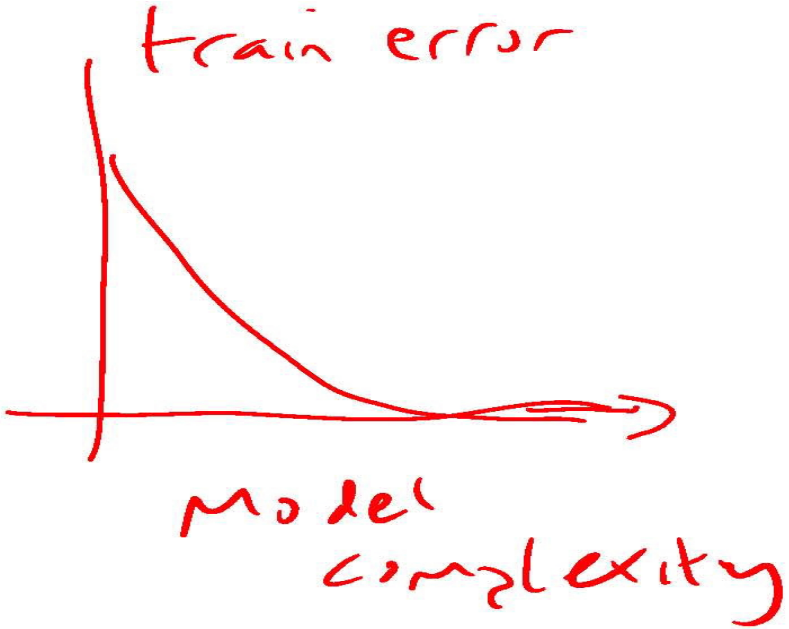


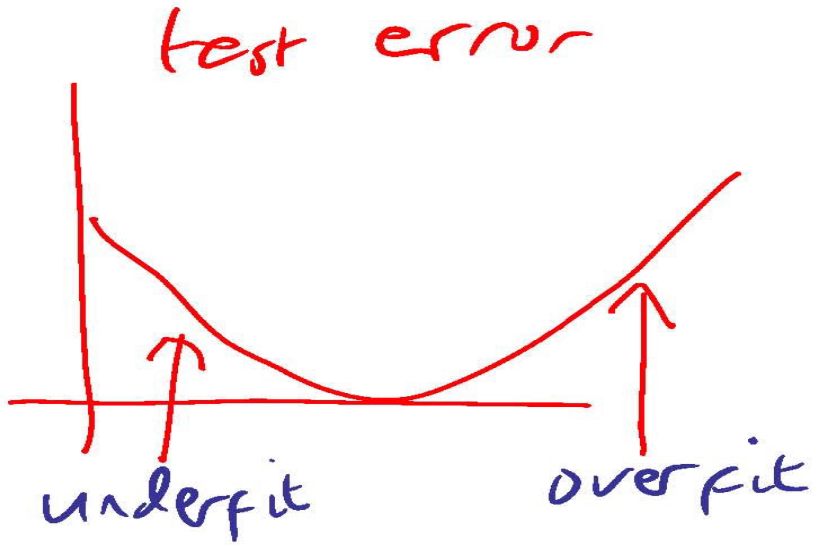
Tradeoff simplicity for model fit



Ockham's razor

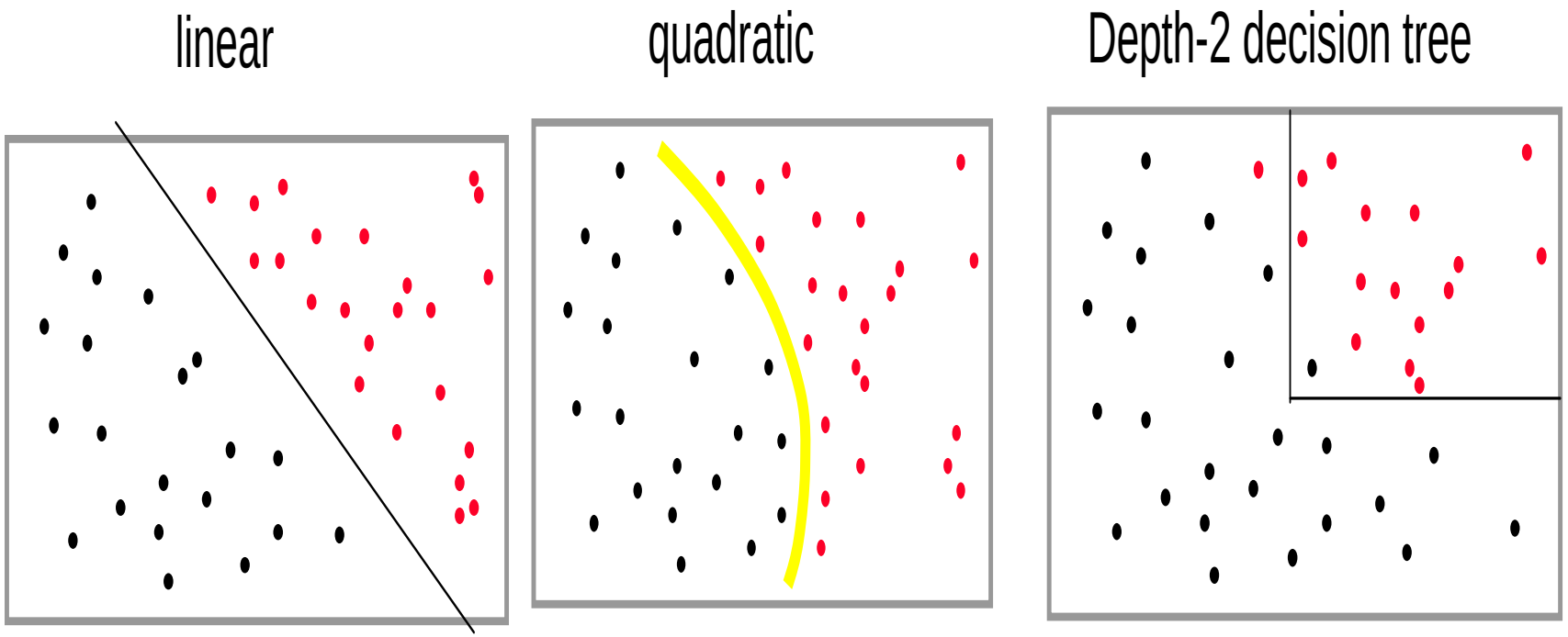
- If two models fit the data equally well, pick the simpler one
- In general, since our goal is to predict the test data, we may choose to incur errors on the training set if it results in a simpler function.





Function fitting

- Choose the right hypothesis classe $\mathcal{H}$ given training data $\mathcal{D}$.



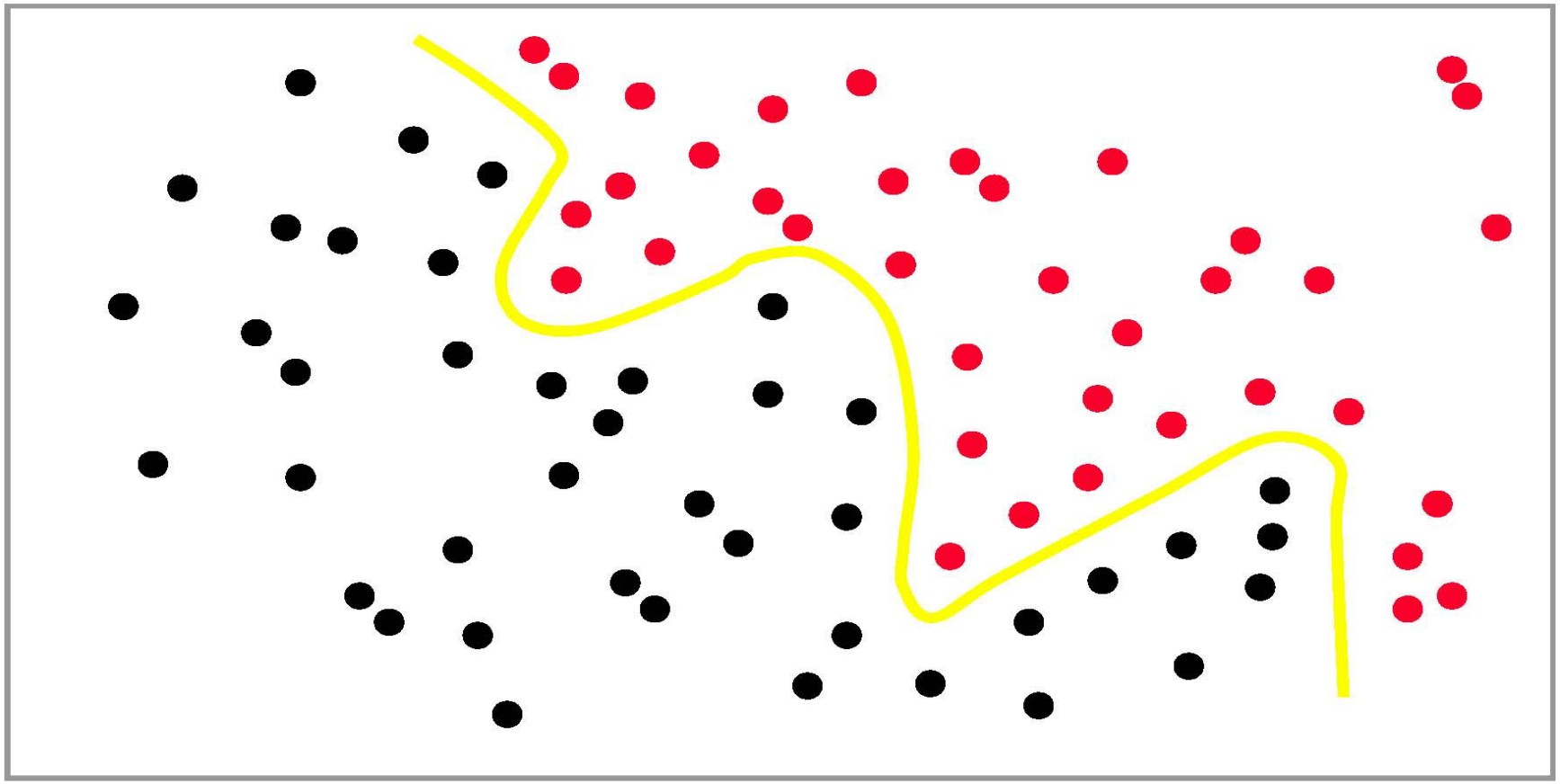
- Learn parameters θ of function $f(\mathbf{x};\theta)$ given $\mathcal{H}$ and $\mathcal{D}$; e.g.

$$f(\mathbf{x};\theta) = sgn\left(\theta^T \begin{pmatrix} 1 & \mathbf{x} \end{pmatrix}\right) = sgn\left(\theta_0 + \theta_1 x_1 + \theta_2 x_2\right)$$

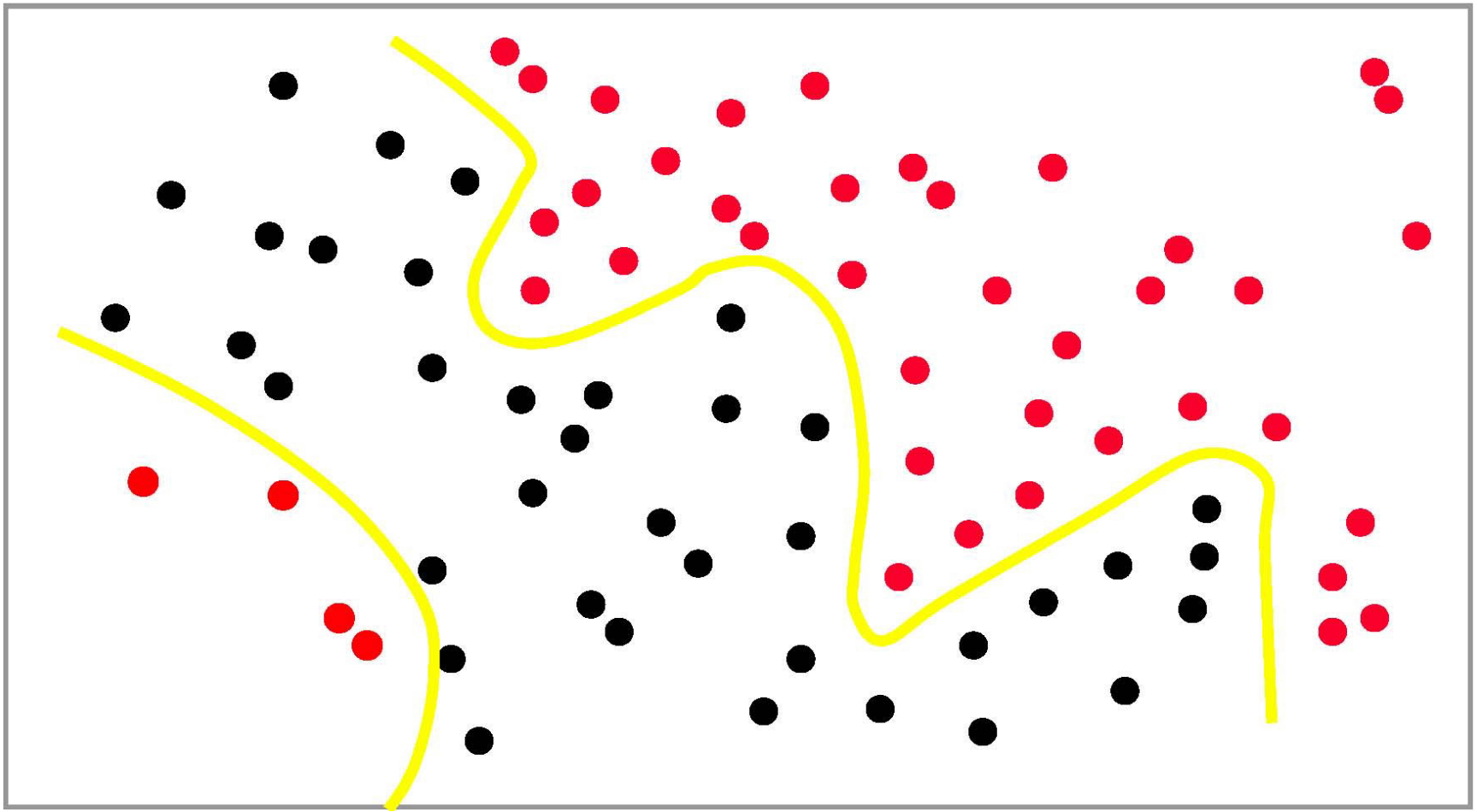
according to a given (typically statistical) criterion.

Hypothesis class depends on data

- More complex function is ok if we have more data, because we have more evidence for it.

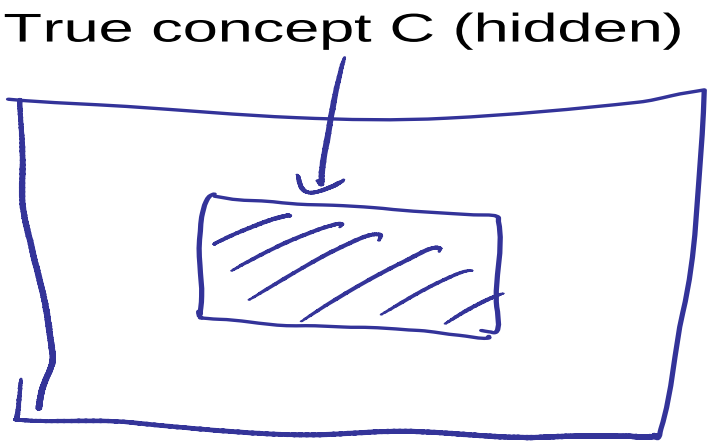


Decision regions might be discontinuous

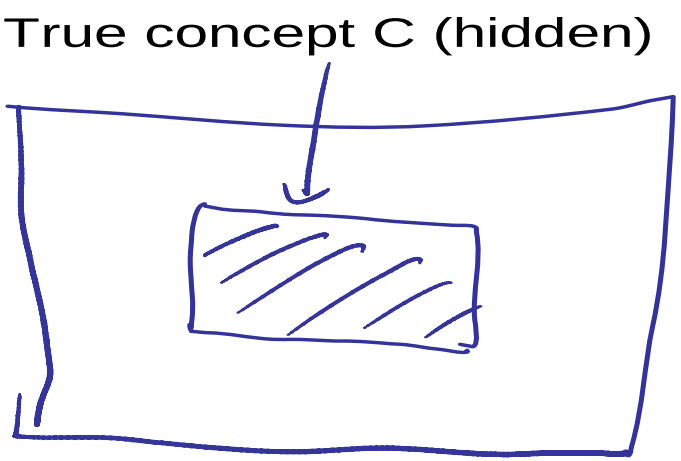


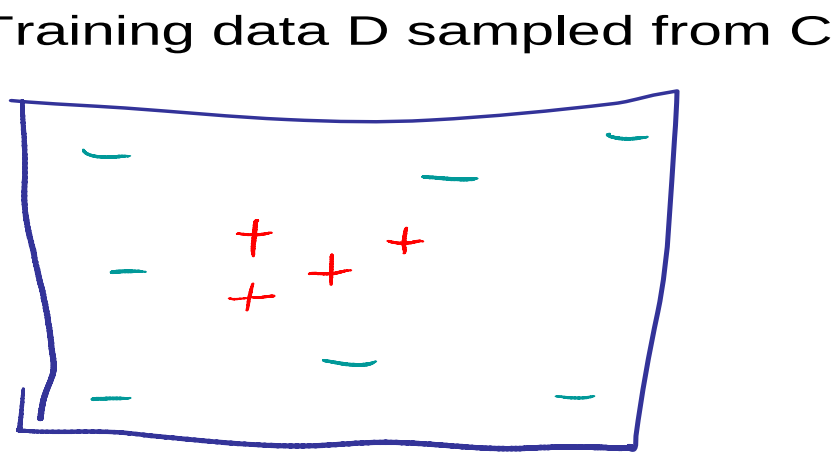
Another Simple Example

- Learn concept of "healthy levels" of cholestrol x_1 and insulin x_2 from positive and negative examples.
- $\mathcal{H}$ = space of rectangles in $\mathbb{R}^2$.
- Assume for the time being that the "true" mapping belongs also to $\mathcal{H}$.

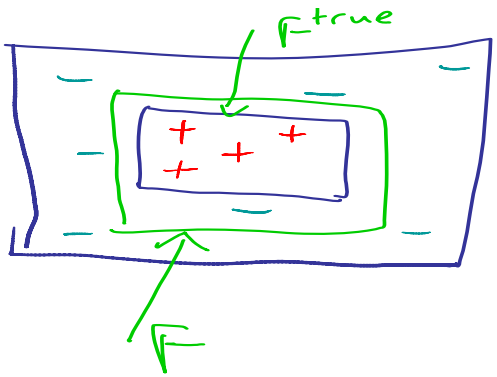


Training Data

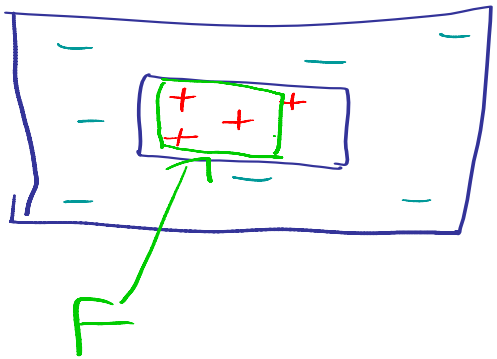




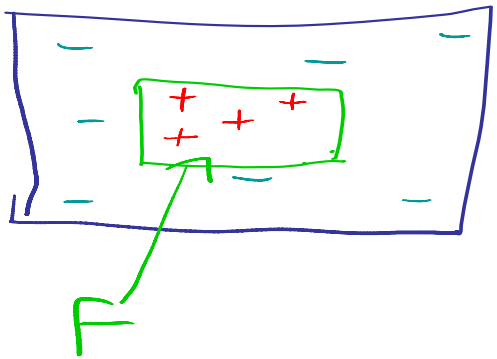
Learning: Inferring hidden concept/function



Too big

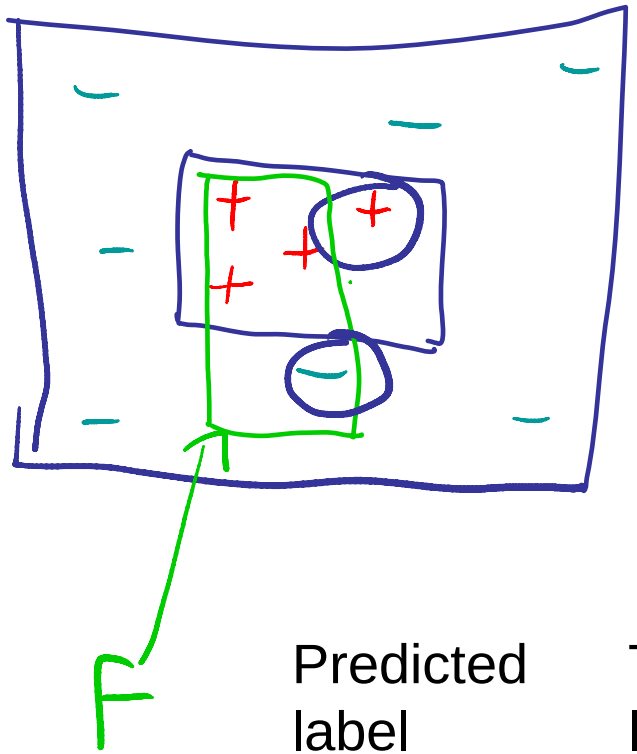


Too small



Just right

Empirical/Training Error

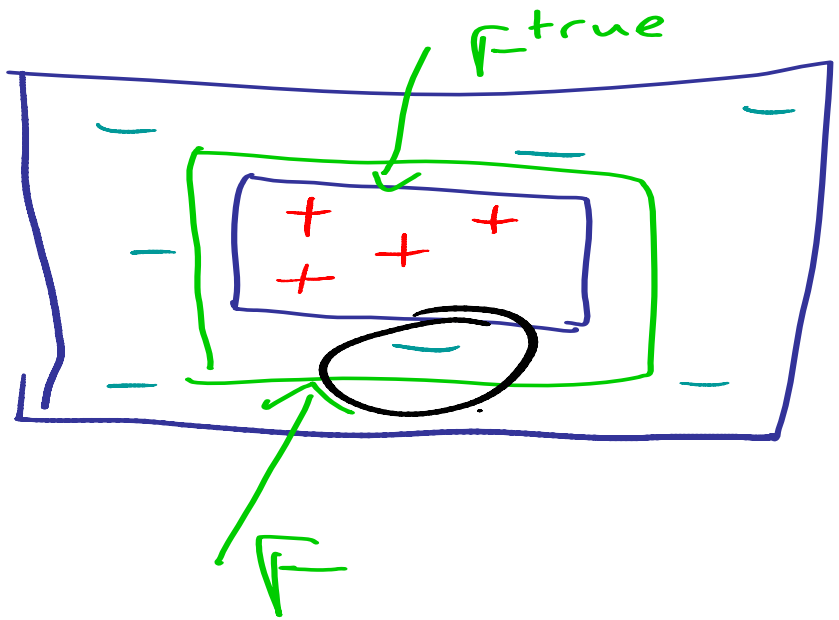


Predicted label

True label

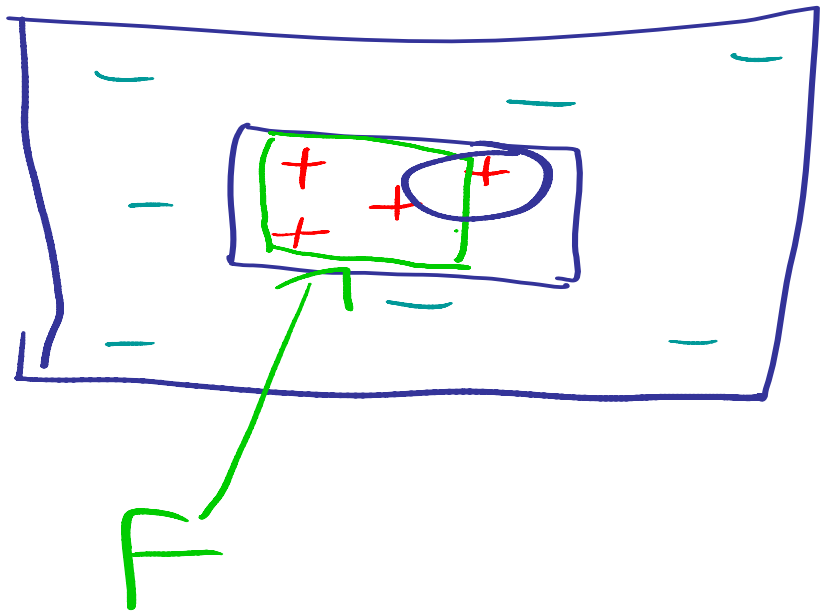
$$N_{err} = \sum_{n=1}^{N} I(\hat{y}(x_n) \neq y_n)$$

False Positive



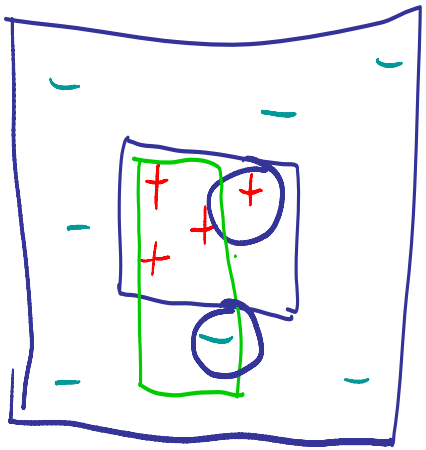
$$N\_\text{FalsePos} = \sum_{i=1}^{N} \mathbb{I}\left(\widehat{y}\left(\mathbf{x}_i\right) = 1 \wedge y_i = 0\right)$$

False Negative



$$N\_\text{FalseNeg} = \sum_{i=1}^{N} \mathbb{I}\left(\widehat{y}\left(\mathbf{x}_i\right) = 0 \wedge y_i = 1\right)$$

Generalization Error and Empirical Risk Minimization

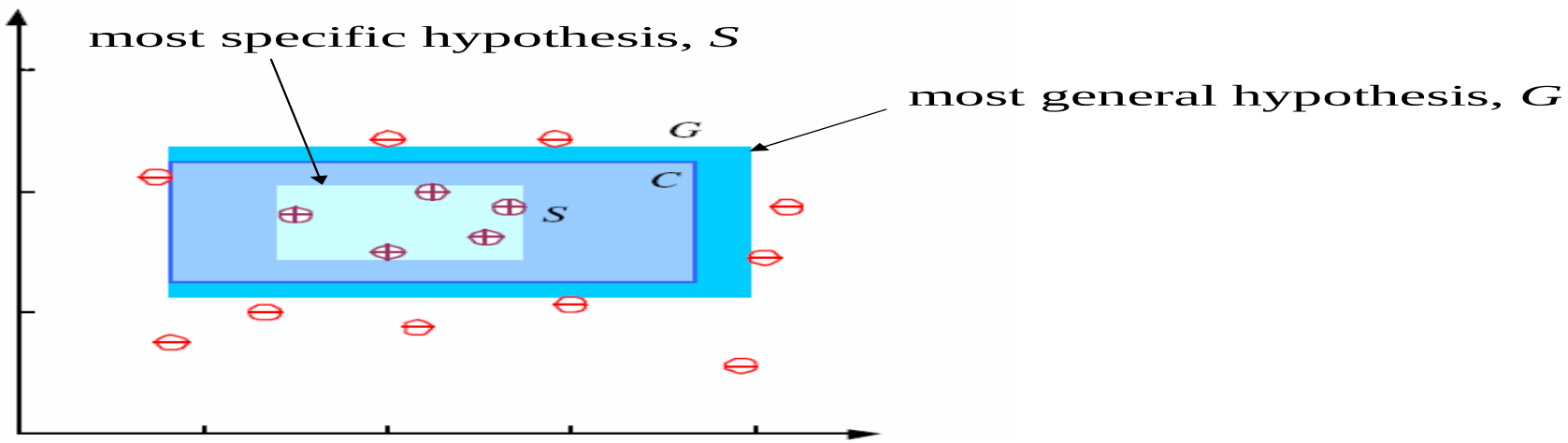


- Generalization error

$$\mathbb{P}\text{roba}\left[\text{error}\right] = \mathbb{E}\left[\mathbb{I}\left(\widehat{y}\left(\mathbf{x}\right) \neq y\right)\right] = \int \mathbb{I}\left(\widehat{y}\left(\mathbf{x}\right) \neq y\right)\ p\left(\mathbf{x},y\right) d\mathbf{x}dy$$

where the expectation is w.r.t the UNKNOWN probability density function $p\left(\mathbf{x},y\right)$ of $\left(\mathbf{x},y\right)$.

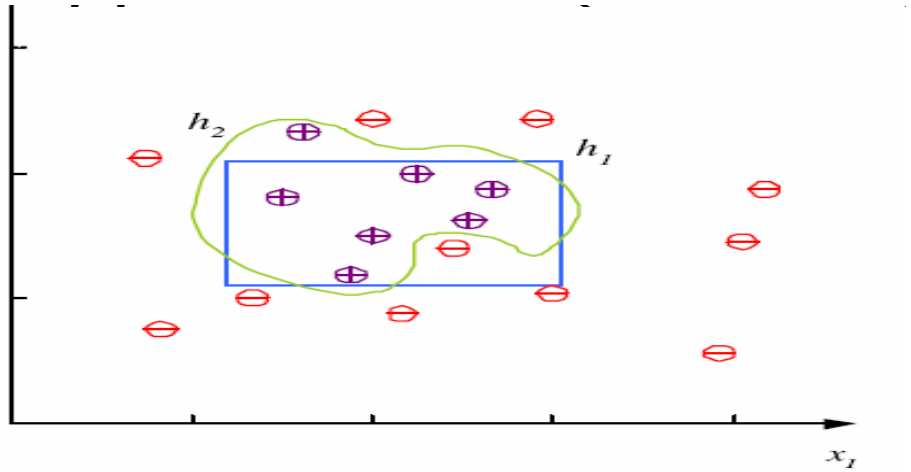
Version Space



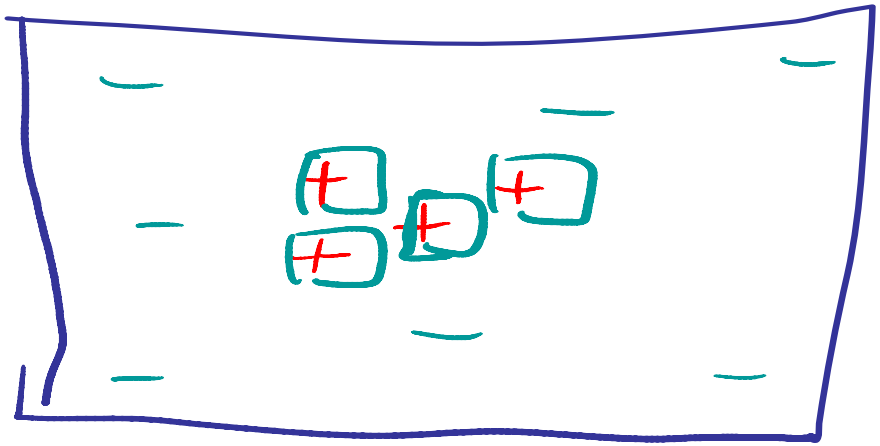
- There may be many functions which have zero training error, ranging from the most specific hypothesis to the most general. Which one we pick depends on our prior knowledge.

Lower Bound on Achievable Error Rate

- If the true concept (green blob) is a rectangle, we can fit it perfectly, and thus get 0 training error. But if the truth is more complex, we will just choose the best-fitting rectangular approximation (blue box) and so Err_Train=N_err$\neq$ 0.



Overfitting in Rectangle Land



- We can always make the empirical/training error be 0 by putting a little rectangle around every + in the training set.
- This may not lead to good generalization performance
- Hence we cannot use directly empirical error to select between models of different complexity.