9.54
class 8

Supervised learning

Optimization, regularization, kernels

Shimon Ullman + Tomaso Poggio

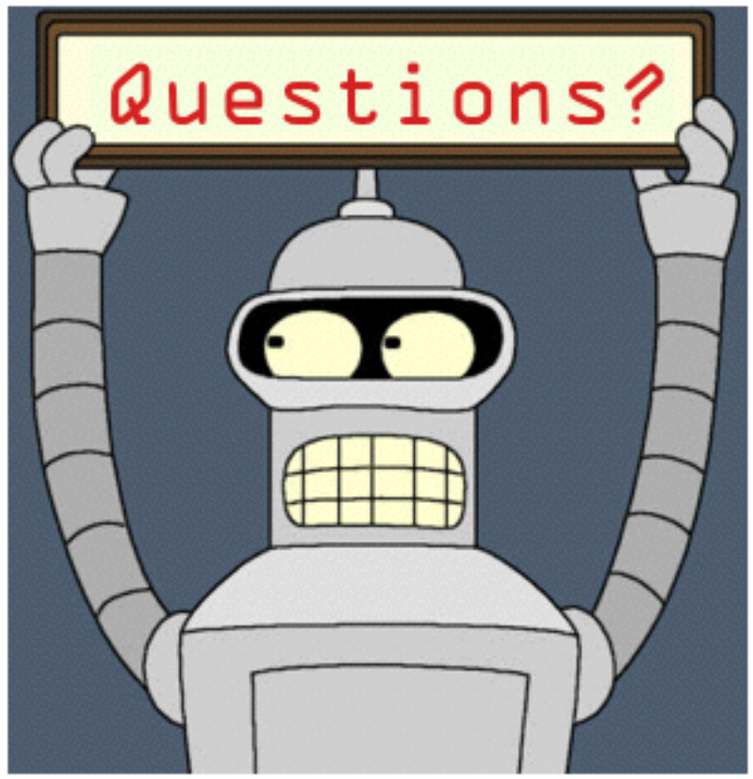
Danny Harari + Daneil Zysman + Darren Seibert

Center for Brains,
Minds & Machines

The Regularization Kingdom

- Loss functions and empirical risk minimization
- Basic regularization algorithms

this part of the lecture contains a bit of math, but we'll try to emphasize concepts and ideas. questions might help prevent this...

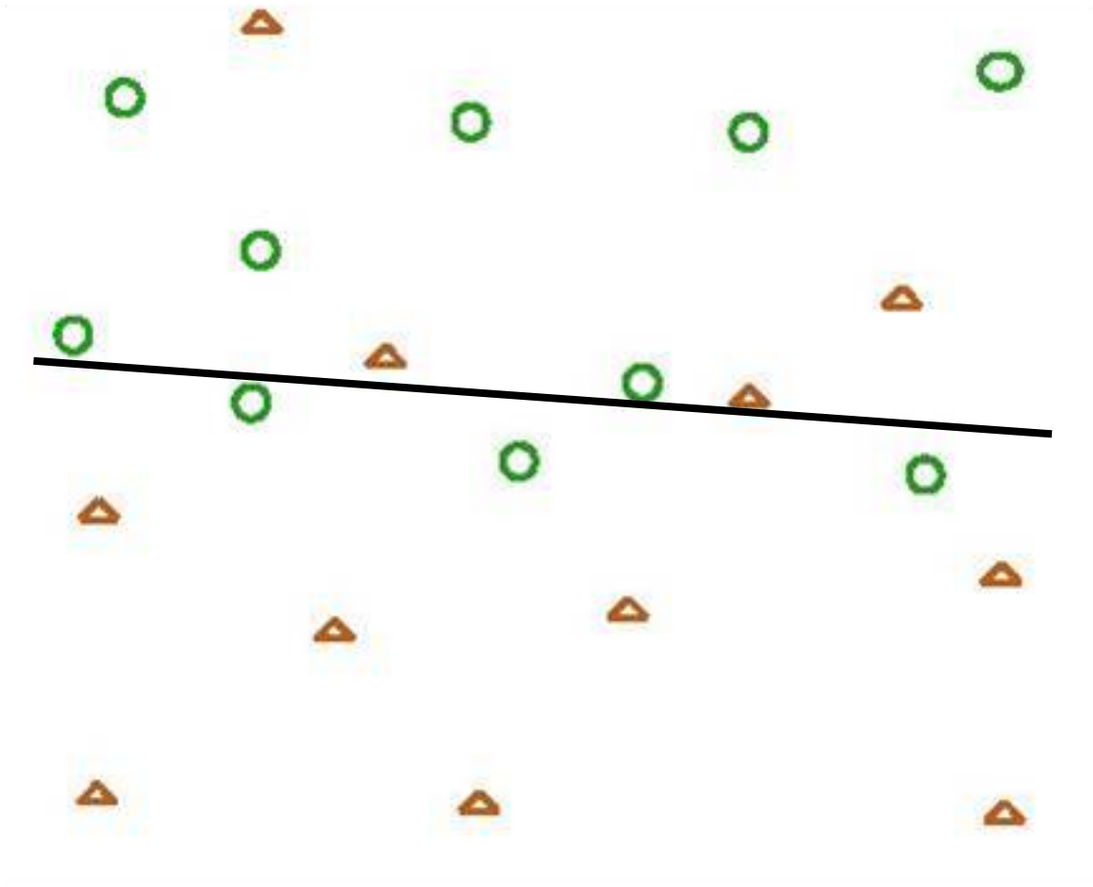


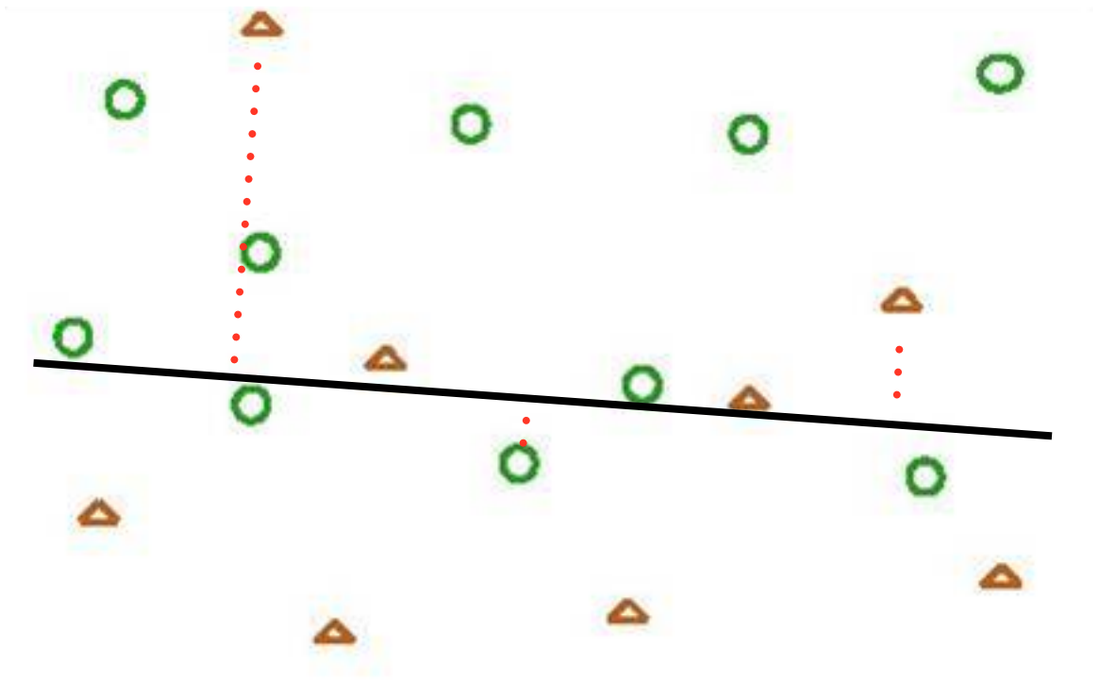
Given a Training Set

$$S = (x_1, y_1), \ldots, (x_n, y_n)$$

Find

$$f(x) \sim y$$

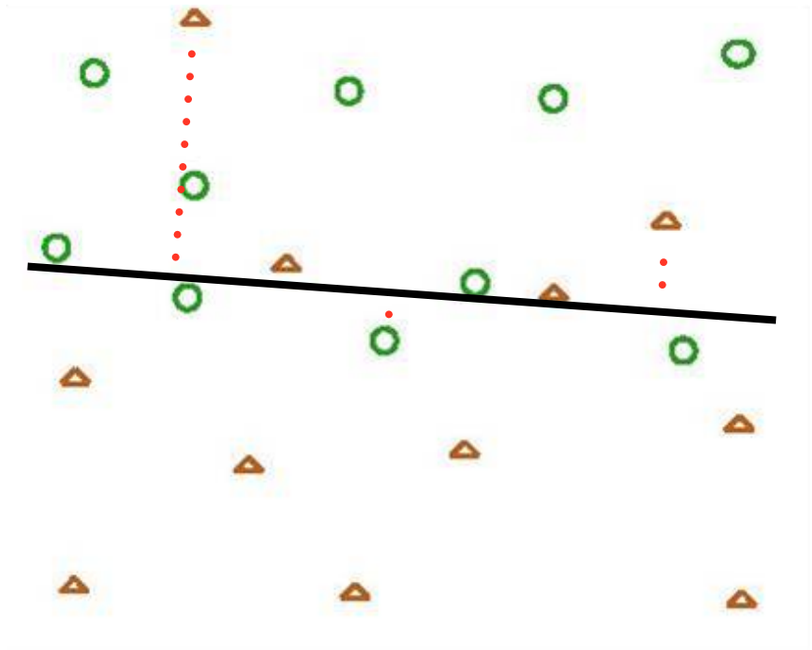


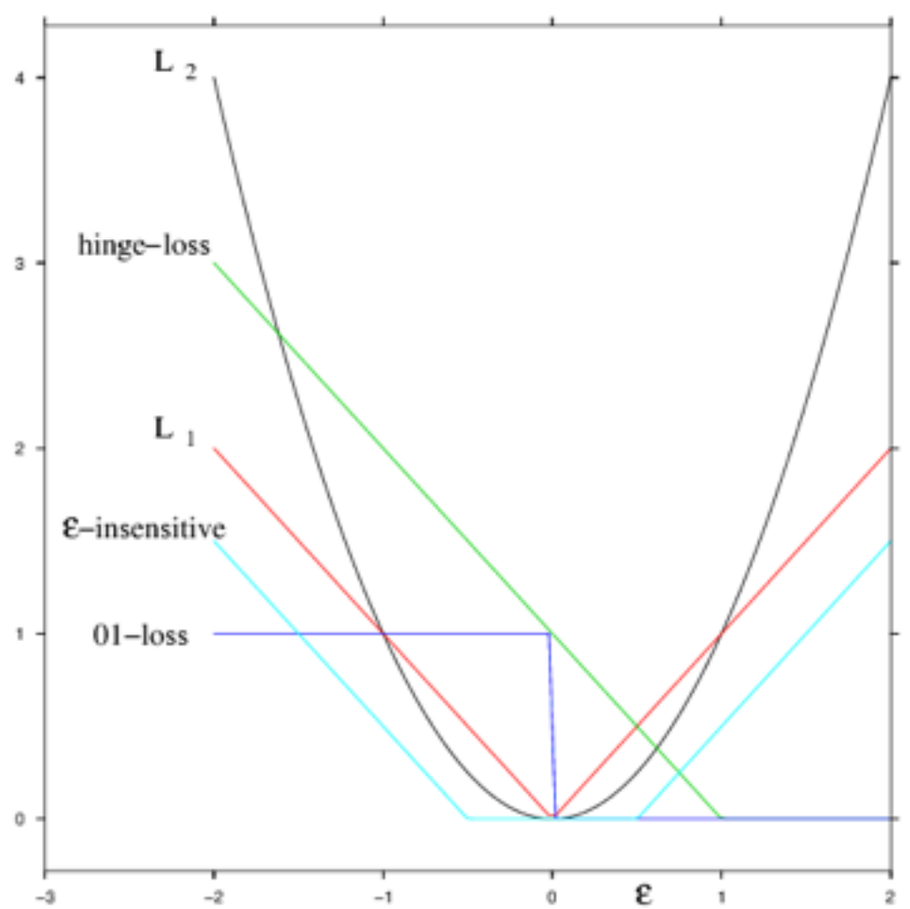


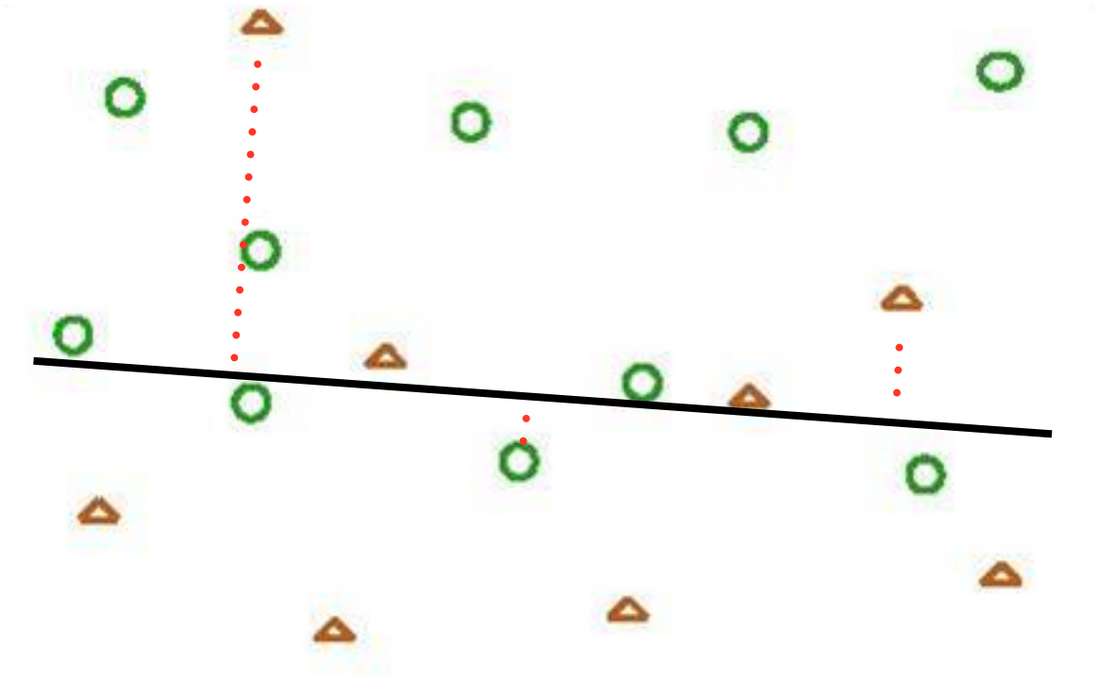
We need a way to measure errors

Loss function $V(f(x), y)$

- $0-1$**-loss** $V(f(x),y) = \theta(-yf(x))$ (θ is the step function)
- **square loss (L2)** $V(f(x),y) = (f(x)-y)^2 = (1-yf(x))^2$
- **absolute value (L1)** $V(f(x),y) = |f(x)-y|$
- Vapnik's ϵ-**insensitive loss** $V(f(x),y) = (|f(x)-y|-\epsilon)_+$
- **hinge loss** $V(f(x),y) = (1-yf(x))_+$
- **logistic loss** $V(f(x),y) = \log(1-e^{-yf(x)})$ logistic regression
- **exponential loss** $V(f(x),y) = e^{-yf(x)}$







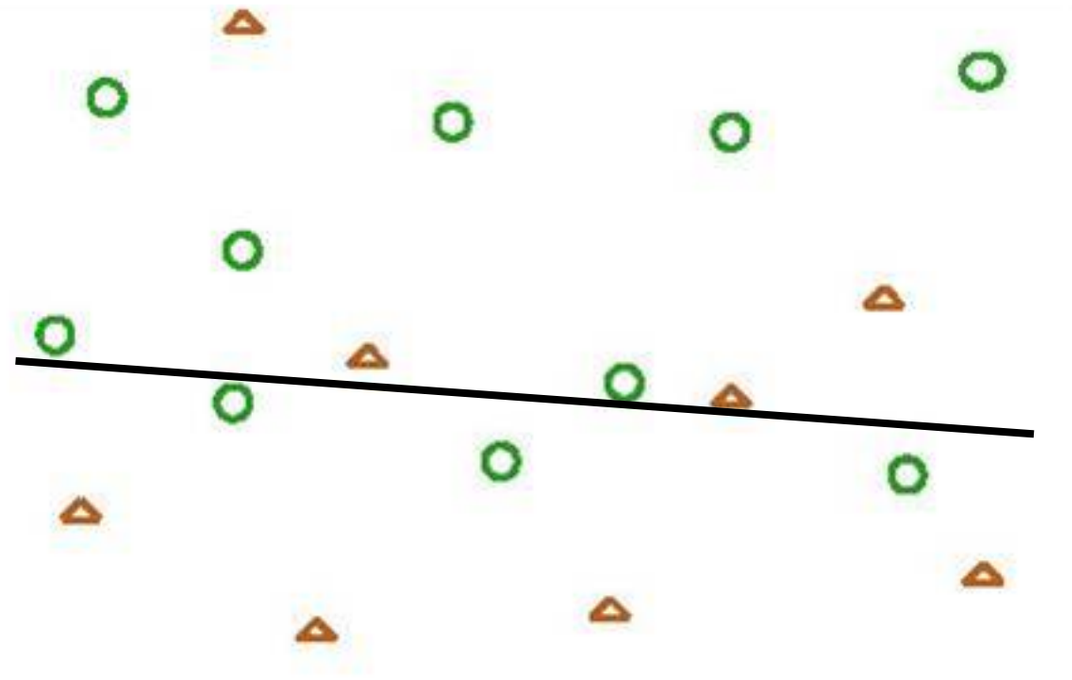
Given a loss function $V(f(x), y)$

We can define the Empirical Error

$$I_S[f] = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

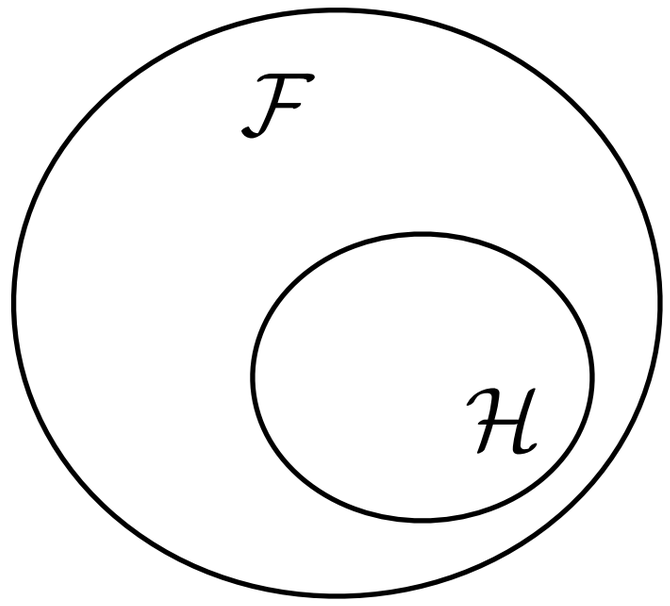
``Learning processes do not take place in vacuum."

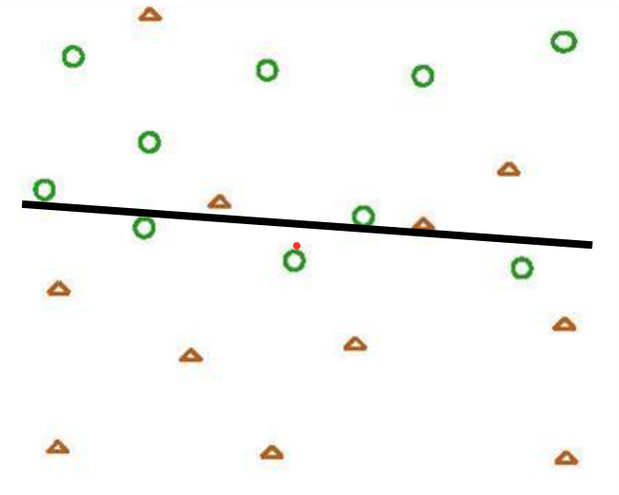
Cucker and Smale, AMS 2001



We need to fix a <u>Hypotheses Space</u>

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$

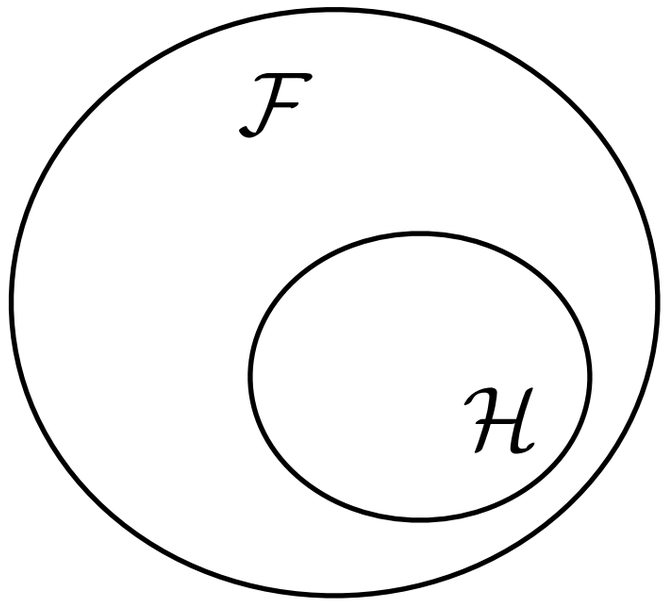


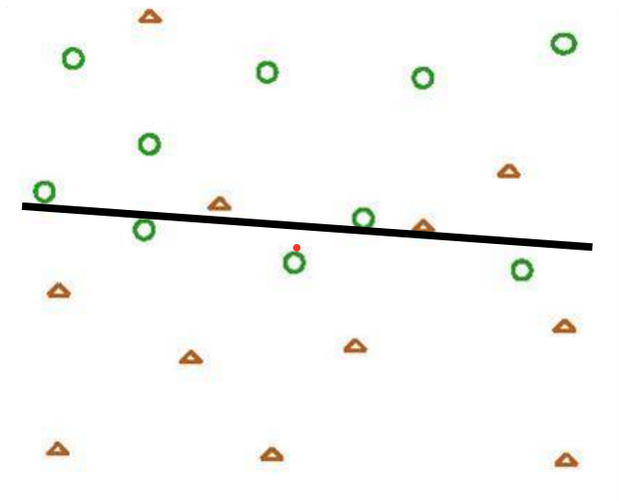


parametric

- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$





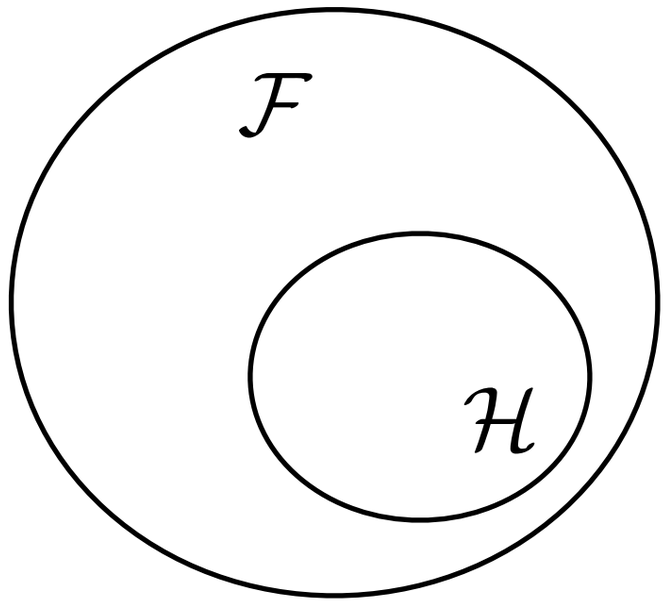
parametric

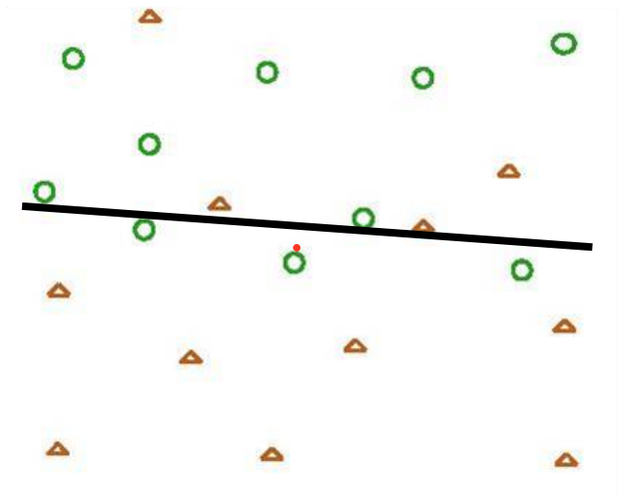
- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

semi-parametric

- **Generalized linear models** $f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j$

$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$





parametric

- **Linear model** $f(x) = \sum_{j=1}^{p} x^j w^j$

semi-parametric

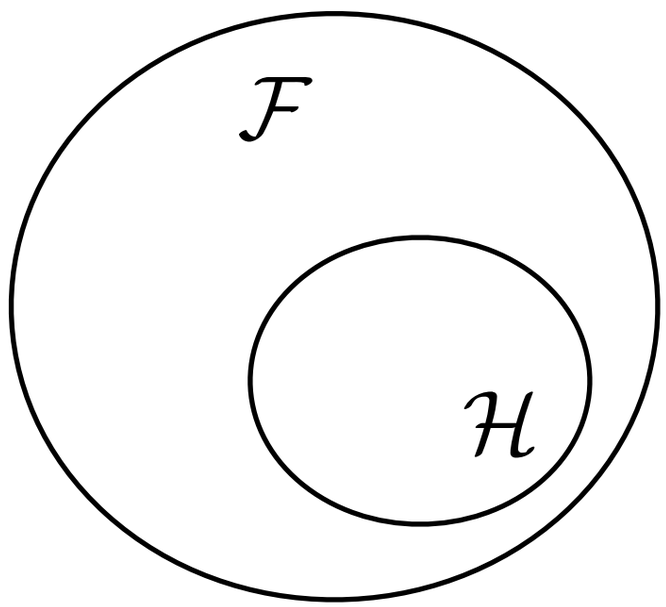
- **Generalized linear models** $f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j$

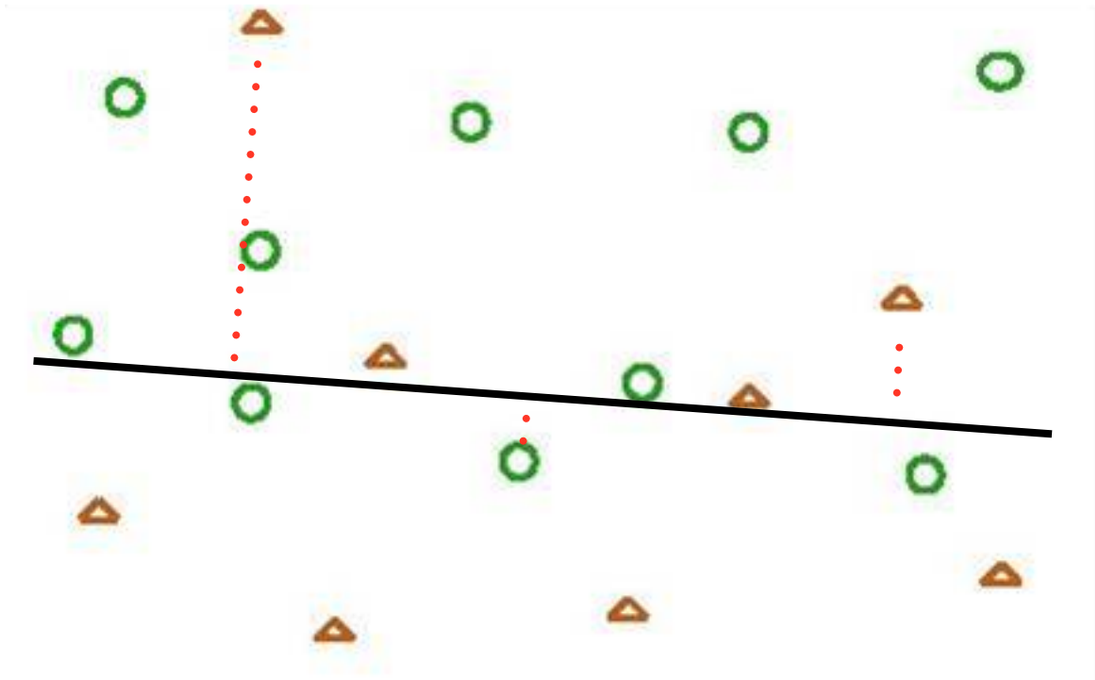
non-parametric

- **Reproducing kernel Hilbert spaces** $f(x) = \sum_{j \geq 1} \Phi(x)^j w^j = \sum_{i \geq 1} K(x, x_i) \alpha_i$

$K(x, x')$ is a symmetric positive definite function called reproducing kernel

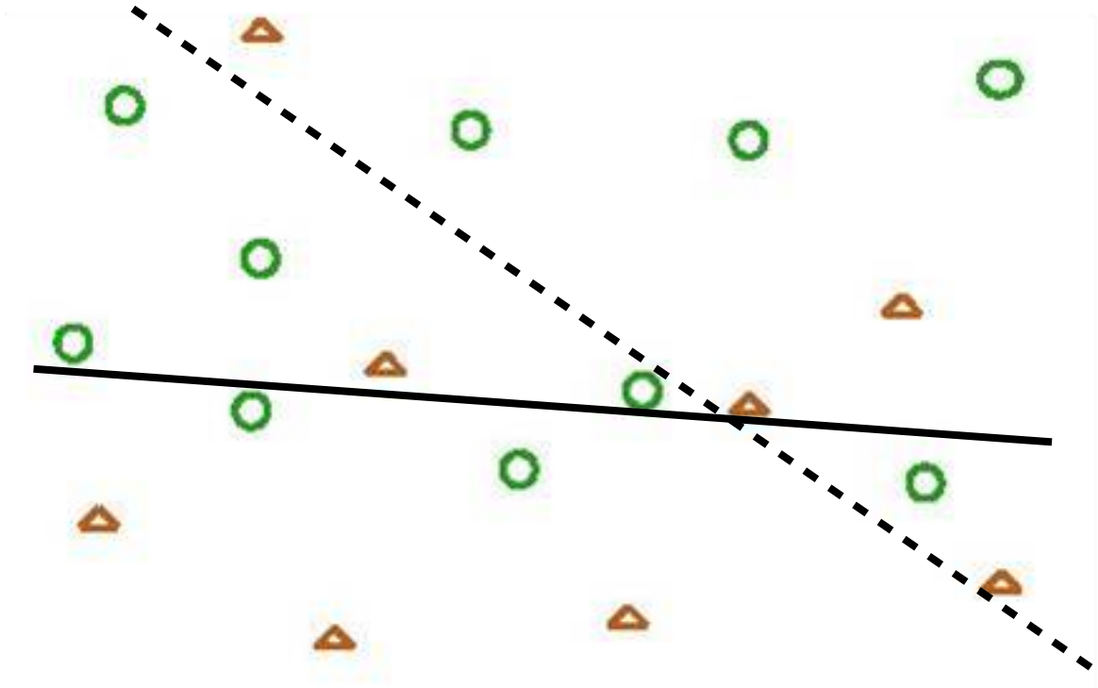
$$\mathcal{H} \subset \mathcal{F} = \{f \mid f : X \to Y\}$$





Empirical Risk Minimization (ERM)

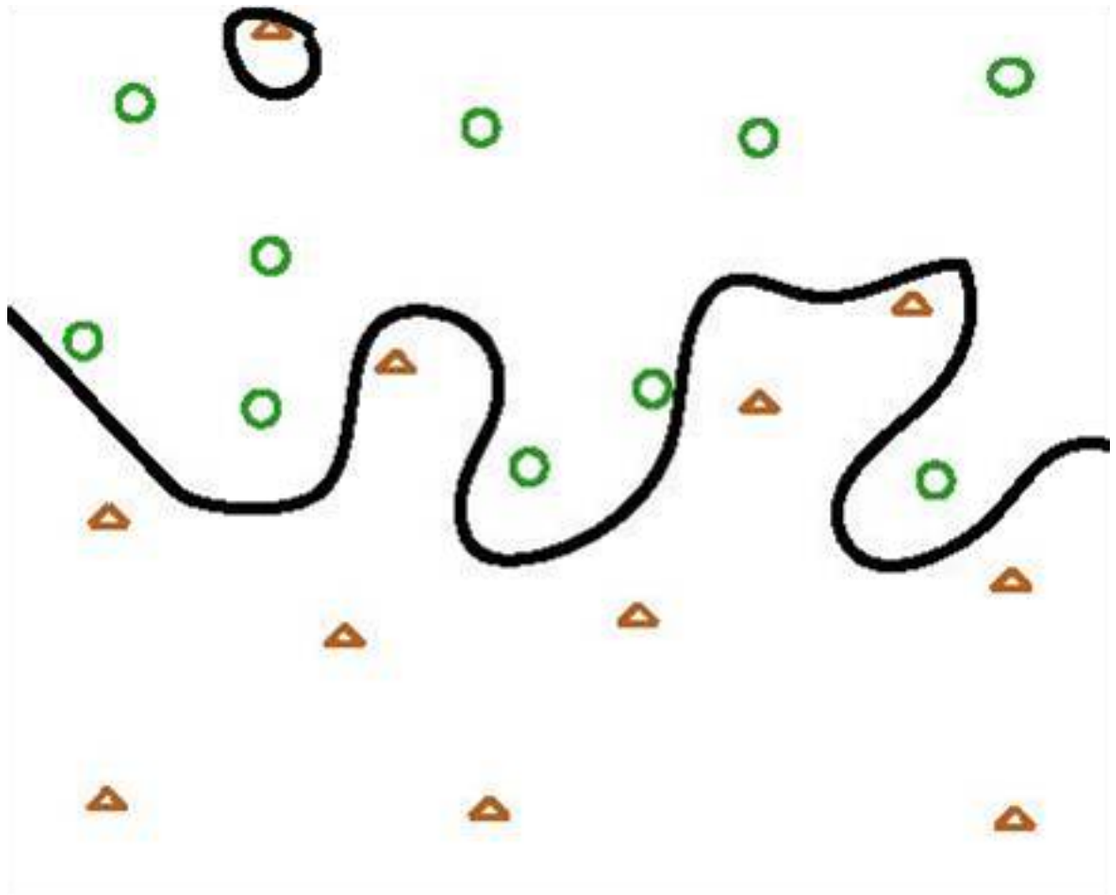
$$\min_{f \in \mathcal{H}} I_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$



Empirical Risk Minimization (ERM)

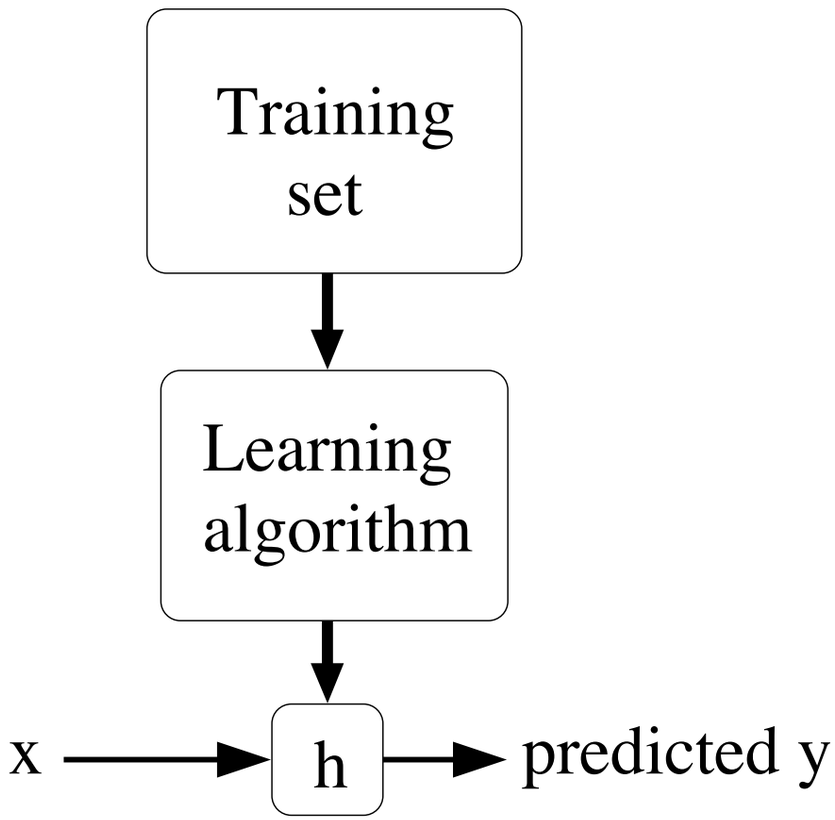
$$\min_{f \in \mathcal{H}} I_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

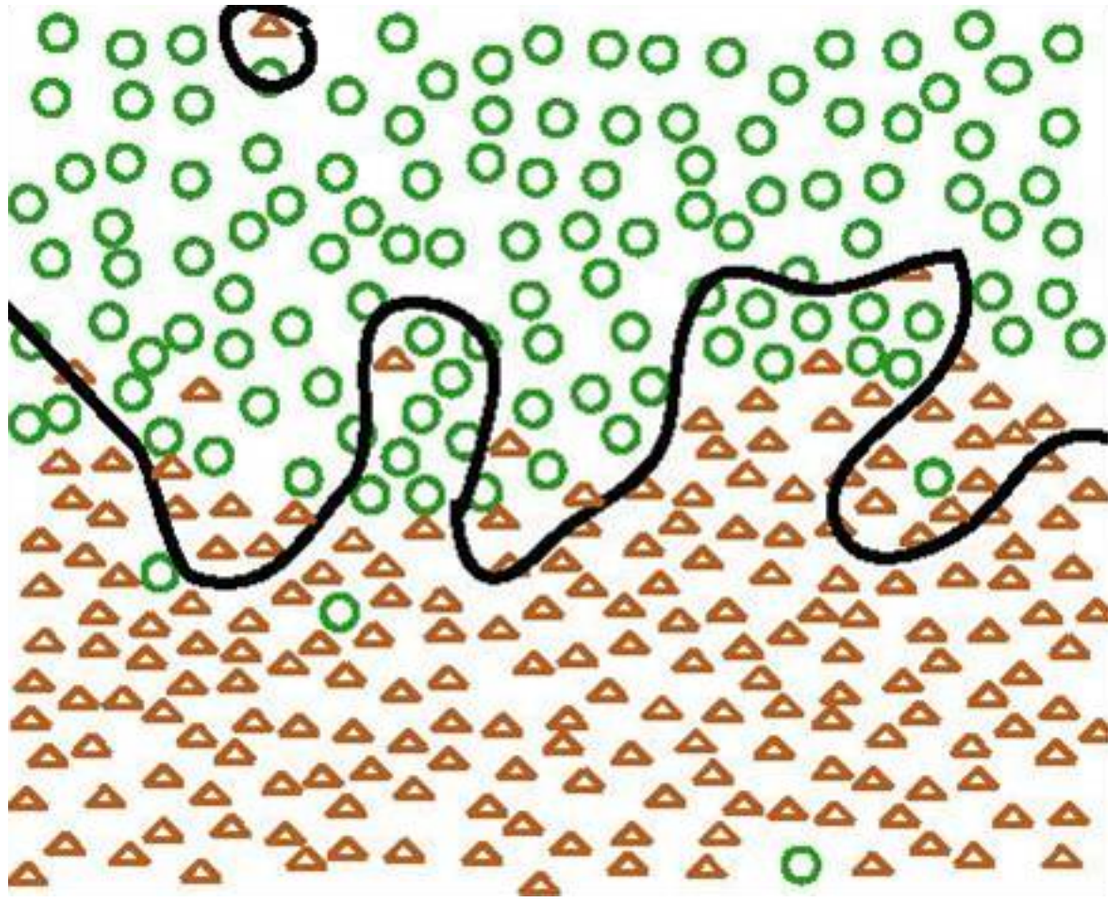
Which is a good solution?



Empirical Risk Minimization (ERM)

$$\min_{f \in \mathcal{H}} \mathcal{E}_S[f] = \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$





The training set

$$S = (x_1, y_1), \ldots, (x_n, y_n)$$

is sampled identically and independently (i.i.d) from a fixed unknown probability distribution $p(x, y) = p(x)p(y|x)$

Learning is an ill-posed problem

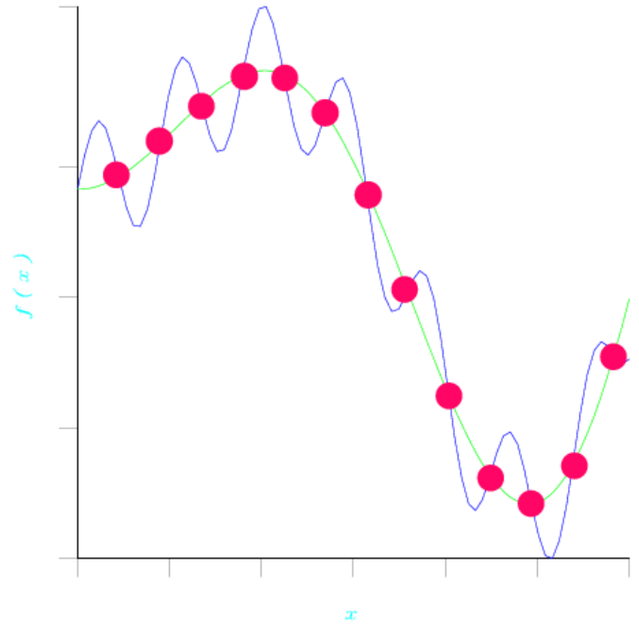
Jacques Hadamard

Ill posed problems often arise if one tries to infer general laws from few data

- the hypothesis space is too large
- there are not enough data

In general ERM leads to ill-posed solutions because

- the solution may be too complex
- it may be not unique
- it may change radically when leaving one sample out



Regularization Theory provides results and techniques to restore well-posedness, that is stability (hence generalization)

- Beyond drawings & intuitions (...) there is a deep, **rigorous mathematical foundation** of regularized learning algorithms (Cucker and Smale, Vapnik and Chervonenkis,).

Theory of learning is a synthesis of different fields, e.g. Computer Science (Algorithms, Complexity) and Mathematics (Optimization, Probability, Statistics).

- Central to the Theory of Machine Learning is the problem of understanding condition under which ERM can solve

$$\inf \mathcal{E}(f), \quad \mathcal{E}(f) = \mathbb{E}_{(x,y)} V(y, f(x))$$

Algorithms: The Regularization Kingdom

- loss functions and empirical risk minimization
- **basic regularization algorithms**

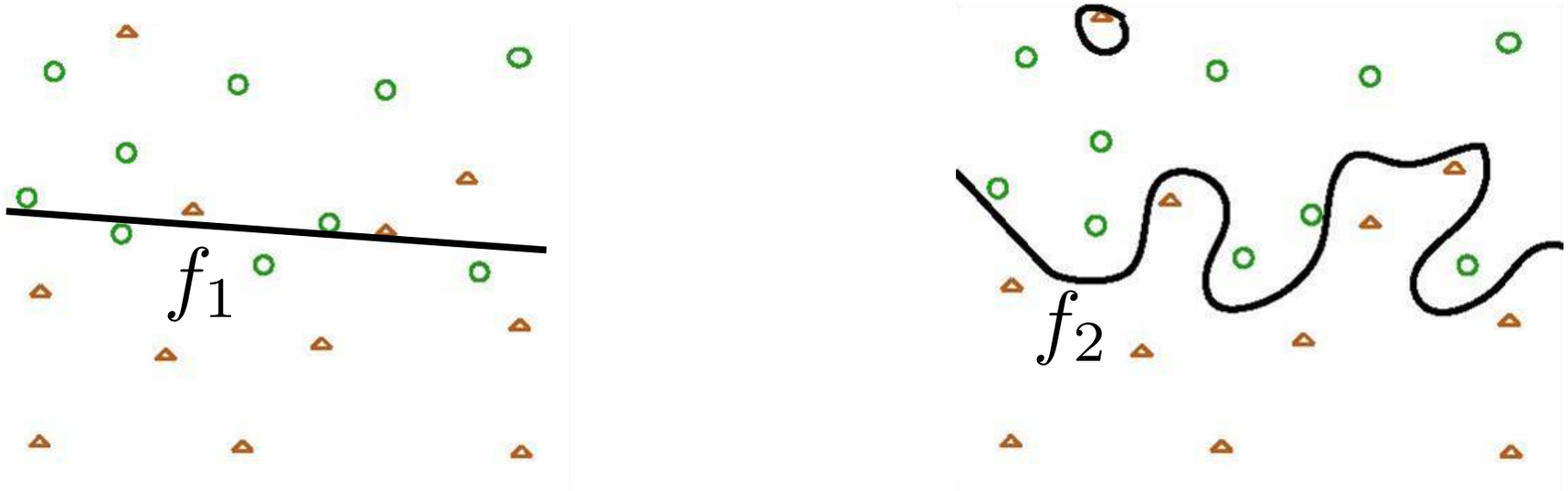
(Tikhonov) Regularization

$$\min_{f \in \mathcal{H}} \{ \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i)) + \lambda R(f)) \} \rightarrow f_S^{\lambda}$$

regularization parameter

regularizer

- The regularizer describes the *complexity* of the solution



$R(f_2)$ is bigger than $R(f_1)$

- The regularization parameter determines the trade-off between complexity and empirical risk

Stability and (Tikhonov) Regularization

Math

Consider $f(x) = w^T x = \sum_{j=1}^{p} w^j x^j$, and $R(f) = w^T w$,

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$\downarrow$$

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2 \right\}$$

$$\rightarrow$$

$$w^T = Y X^T (X X^T)^{-1}$$

$$\downarrow$$

$$w^T = Y X^T (X X^T + \lambda I)^{-1}$$

From Linear to Semi-parametric Models

$$\underbrace{f(x) = \sum_{j=1}^{p} x^j w^j}_{\text{linear model}} \Longrightarrow \underbrace{f(x) = \sum_{j=1}^{p} \Phi(x)^j w^j}_{\text{generalized linear model}}$$

If instead of a linear model we have a generalized linear model we simply have to consider

$$X_n = \begin{pmatrix} \Phi(x_1)^1 & \dots & \dots & \dots & \Phi(x_1)^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi(x_n)^1 & \dots & \dots & \dots & \Phi(x_n)^p \end{pmatrix}$$

From Parametric to Nonparametric Models

How about nonparametric models?

Math

Some simple linear algebra shows that

$$w^T = YX^T(XX^T)^{-1} = Y(X^TX)^{-1}X^T = CX^T$$

since $X^T(XX^T)^{-1} = (X^TX)^{-1}X^T$

Then

$$f(x) = w^Tx = CX^Tx = \sum_i^n c_i x_i^T x$$

We can compute C_n or w_n depending whether $n \leq p$.

The above result is the most basic form of the <u>Representer Theorem.</u>

From Linear to Nonparametric Models

Math

Note that

$$f(x) = \sum_{j=1}^{p} w_n^j x^j = \sum_{i=1}^{n} \underbrace{x_i^T x}_{\sum_{j=1}^{p} x_i^j x^j} \quad c_i$$

We can now consider a truly non parametric model

$$f(x) = \sum_{j \geq 1} w^j \Phi(x)^j = \sum_{i=1}^{n} \underbrace{K(x, x_i)}_{\sum_{j \geq 1} \Phi(x_i)^j \Phi(x)^j} c_i$$

From Linear to Nonparametric Models

Math

We can now consider a truly non parametric model

$$f(x) = \sum_{j \geq 1} w^j \Phi(x)^j = \sum_{i=1}^{n} \underbrace{K(x, x_i)}_{\sum_{j \geq 1} \Phi(x_i)^j \Phi(x)^j} c_i$$

We have

$$C_n = (\underbrace{X_n X_n^T}_{(X_n X_n^T)_{i,j} = x_i^T x_j} + \lambda n I)^{-1} Y_n \longrightarrow C_n = (\underbrace{K_n}_{(K_n)_{i,j} = K(x_i, x_j)} + \lambda n I)^{-1} Y_n$$

Kernels

- **Linear kernel**

$$K(x, x') = x^T x'$$

- **Gaussian kernel**

$$K(x, x') = \mathrm{e}^{-\frac{\|x-x'\|^2}{\sigma^2}}, \qquad \sigma > 0$$

- **Polynomial kernel**

$$K(x, x') = (x^T x' + 1)^d, \qquad d \in \mathbb{N}$$

- **Inner Product kernel/Features**

$$K(x, x') = \sum_{j=1}^{p} \Phi(x)^j \Phi(x')^j \qquad \Phi : X \to \mathbb{R}^p.$$

Reproducing Kernel Hilbert Spaces

Given K, $\exists!$ Hilbert space of functions $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ such that, Math

- $K_x := K(x, \cdot) \in \mathcal{H}$, for all $x \in X$, and
- $f(x) = \langle f, K_x \rangle$, for all $x \in X$, $f \in \mathcal{H}$.

The norm of a function $f(x) = \sum_{i=1}^{n} K(x, x_i) c_i$ is given by

$$\|f\|^2 = \sum_{i,j=1}^{n} K(x_j, x_i) c_i c_j$$

and is a natural complexity measure.

Note: An RKHS is equivalently defined as a Hilbert space where the evaluation functionals are continuous.

Extensions: Other Loss Functions

For most loss functions the solution of Tikhonov regularization is of the form

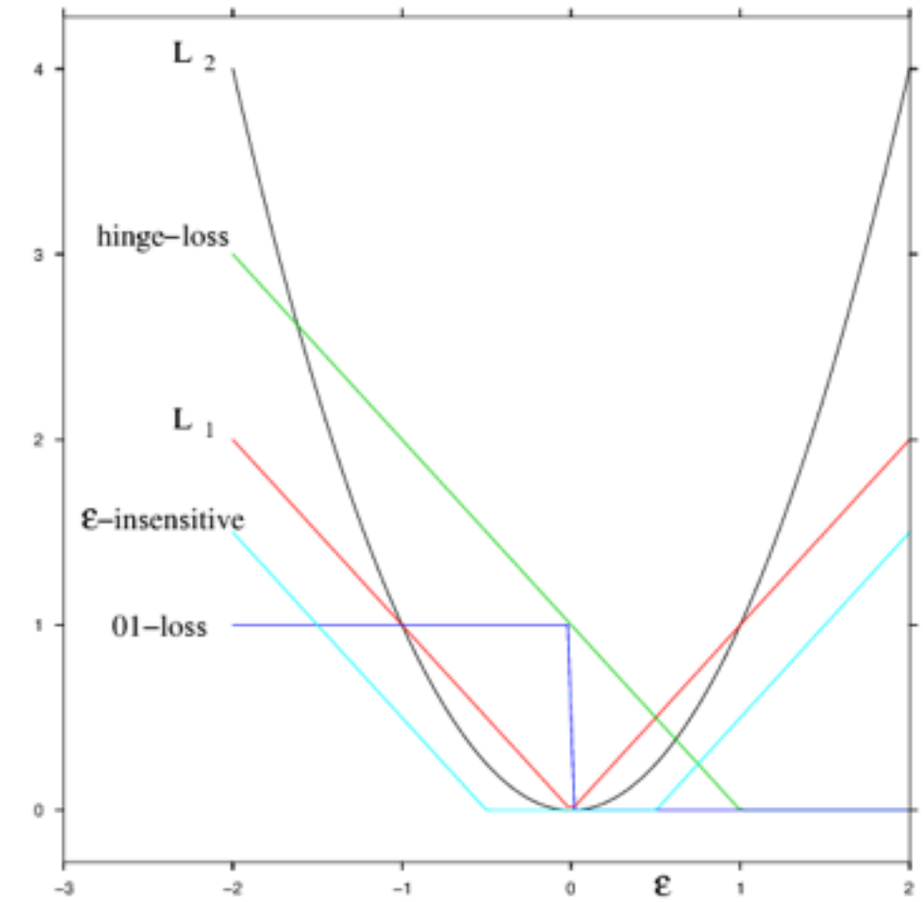
$$f(x) = \sum_{i=1}^{n} K(x, x_i)c_i.$$

- $V(f(x), y) = (f(x) - y)^2$, RLS
- $V(f(x), y) = (|f(x) - y| - \epsilon)_+$ SVM regression
- $V(f(x), y) = (1 - yf(x))_+$ SVM classification
- $V(f(x), y) = \log(1 - e^{-yf(x)})$ logistic regression
- $V(f(x), y) = e^{-yf(x)}$ boosting

Extensions: Other Loss Functions (cont)

By changing the loss function we change the way we compute the coefficients in expansion

$$f(x) = \sum_{i=1}^{n} K(x, x_i) c_i.$$

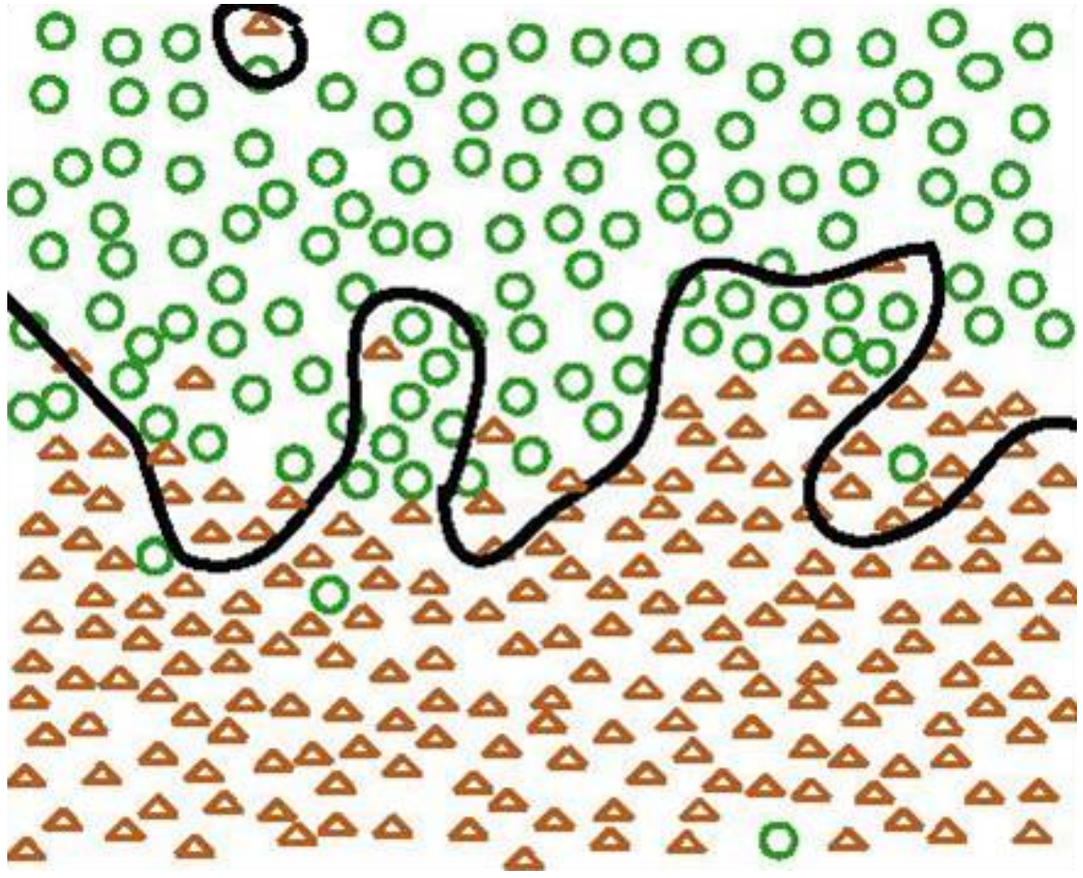


- **Regularization** avoids overfitting, ensures **stability** of the solution and **generalization**
- There are many **different** instance of **regularization beyond Tikhonov**, e.g. early stopping...

$$\min_f \quad \underbrace{I_S[f]}_{\text{data fit term}} \quad + \lambda \underbrace{R(f)}_{\text{complexity/smoothness term}}$$

- Regularization ensures stability of the solution and generalization
- There are different instance of regularization beyond Tikhonov, e.g. early stopping

Conclusions



- Regularization Theory provides results and techniques to **avoid overfitting** (stability is key to generalization)
- Regularization provide a **core** set of **concepts** and **techniques** to solve a variety of problems
- Most algorithms can be seen as a form of regularization

Learning, Generalization and Overfitting

Predictivity or Generalization

Given the data, the goal is to learn how to make decisions/predictions about future data / data not belonging to the training set. **Generalization** is the key requirement emphasized in Learning Theory. This emphasis makes it different from Bayesian or traditional statistics (especially explanatory statistics).

Expected Risk

A good function – we will also speak about *hypothesis* – should incur in only *a few* errors. We need a way to quantify this idea.

Expected Risk

The quantity

$$I[f] = \int_{X \times Y} V(f(x), y) p(x, y) dx dy.$$

is called the expected error and measures the loss averaged over the unknown distribution.

This is really the expected error in the future.

Basic definitions

- $p(x, y)$ probability distribution,
- S_n training set,
- $V(f(x), y)$ loss function,
- $I_n[f] = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$, empirical risk,
- $I[f] = \int_{X \times Y} V(f(x), y) p(x, y) dxdy$, expected risk,

Empirical risk and Generalization

Empirical Risk

The empirical risk is a natural proxy (how good?) for the expected risk

$$I_n[f] = \frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i).$$

Generalization Error

How good a proxy is captured by the generalization error,

$$\mathbb{P}\{|I[f_n] - I_n[f_n]| \leq \epsilon\} \geq 1 - \delta,$$

for $n = n(\epsilon, \delta)$.

Under which conditions is the empirical error a good proxy for the expected error?

Hypotheses Space

In many learning algorithms (not all!) we need to choose a suitable space of hypotheses $\mathcal{H}$.

The **hypothesis space** $\mathcal{H}$ is the space of functions that we allow our algorithm to "look at". For many algorithms (such as optimization algorithms) it is the space the algorithm is allowed to search. lit is important to choose the hypothesis space as a function of the amount of data *n* available.

Hypotheses Space

Examples: linear functions, polynomial, RBFs, Sobolev Spaces...

Learning algorithm

A learning algorithm A is then a map from the data space to $\mathcal{H}$,

$$A(S_n) = f_n \in \mathcal{H}.$$

A learning algorithm should be well-posed, eg stable

In addition to the key property of generalization, a "good" learning algorithm should also be *stable*: f_S should depend continuously on the training set S. In particular, changing one of the training points should affect less and less the solution as n goes to infinity. Stability is a good requirement for the learning problem and, in fact, for any mathematical problem. We open here a small parenthesis on stability and well-posedness.

General definition of Well-Posed and Ill-Posed problems

A problem is **well-posed** if its solution:

- exists
- is unique
- depends continuously on the data (e.g. it is *stable*)

A problem is **ill-posed** if it is not well-posed. In the context of this class, well-posedness is mainly used to mean *stability* of the solution.

ERM

Given a training set S and a function space $\mathcal{H}$, empirical risk minimization is the class of algorithms that look at S and select f_S as

$$f_S = \arg\min_{f \in \mathcal{H}} I_S[f]$$

.

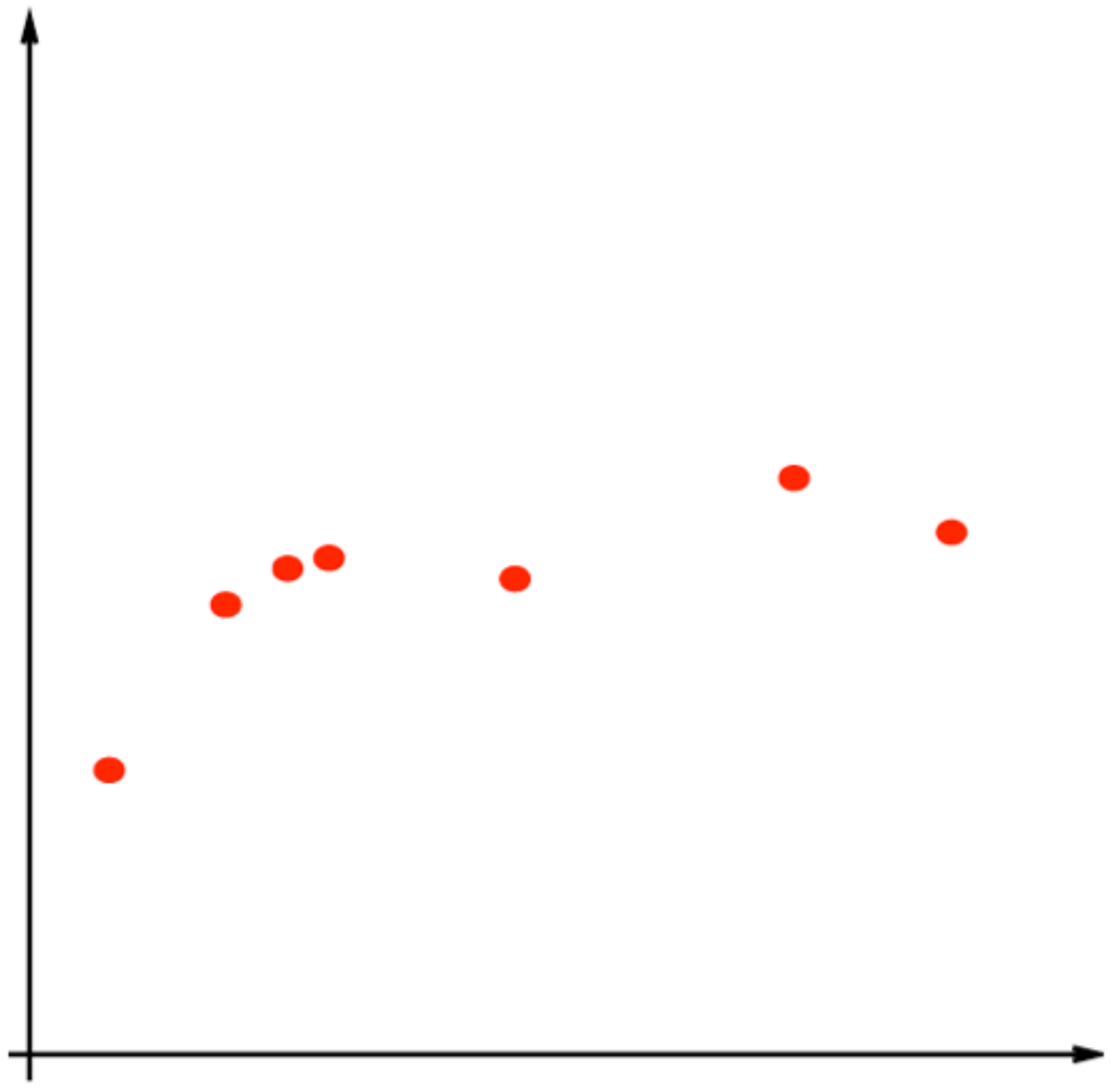
For example linear regression is ERM when $V(z) = (f(x) - y)^2$ and H is space of linear functions $f = ax$.

Generalization and Well-posedness of Empirical Risk Minimization

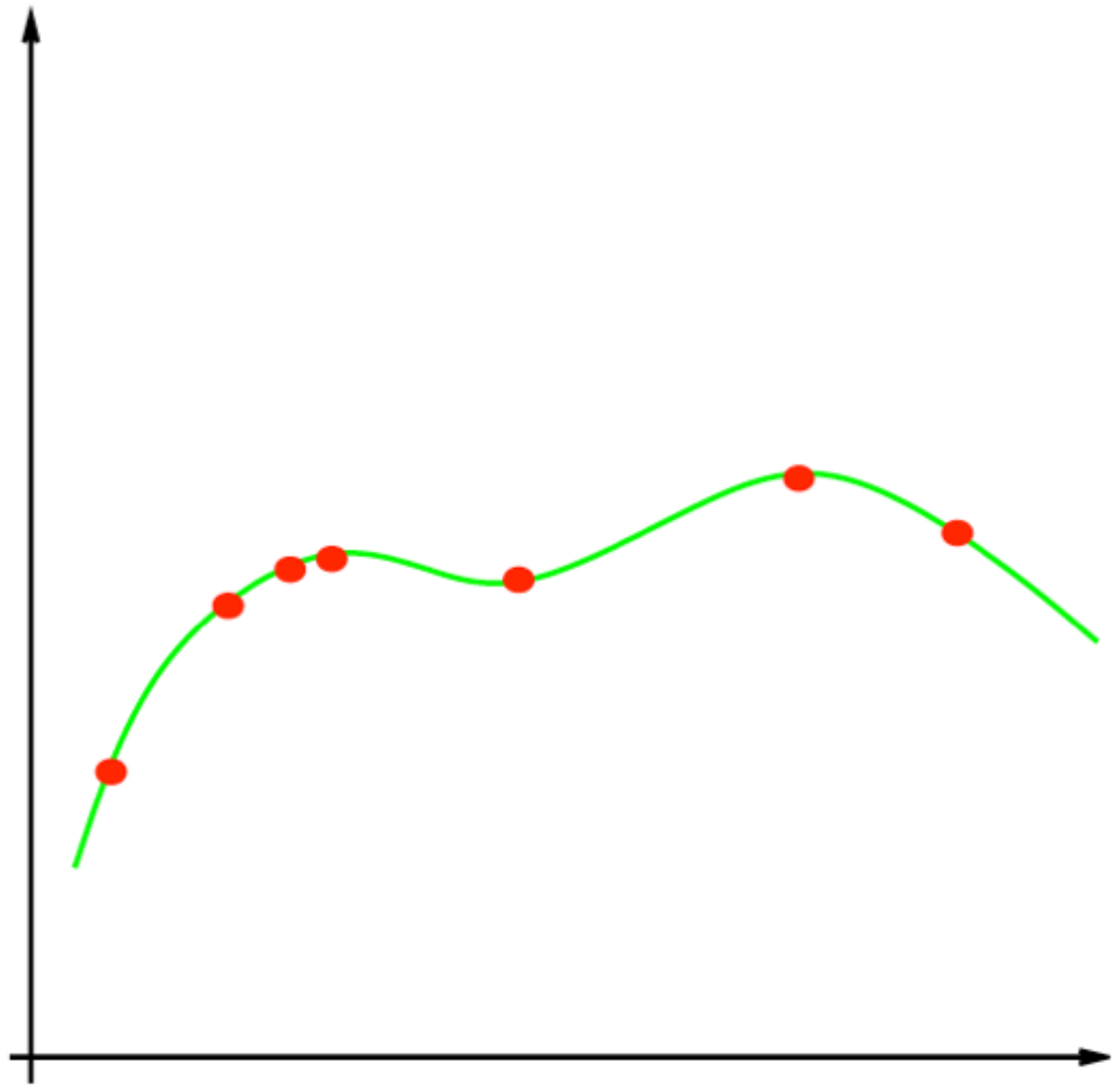
For ERM to represent a “good” class of learning algorithms, the solution should

- *generalize*
- exist, be unique and – especially – be *stable* (well-posedness), according to *some* definition of stability.

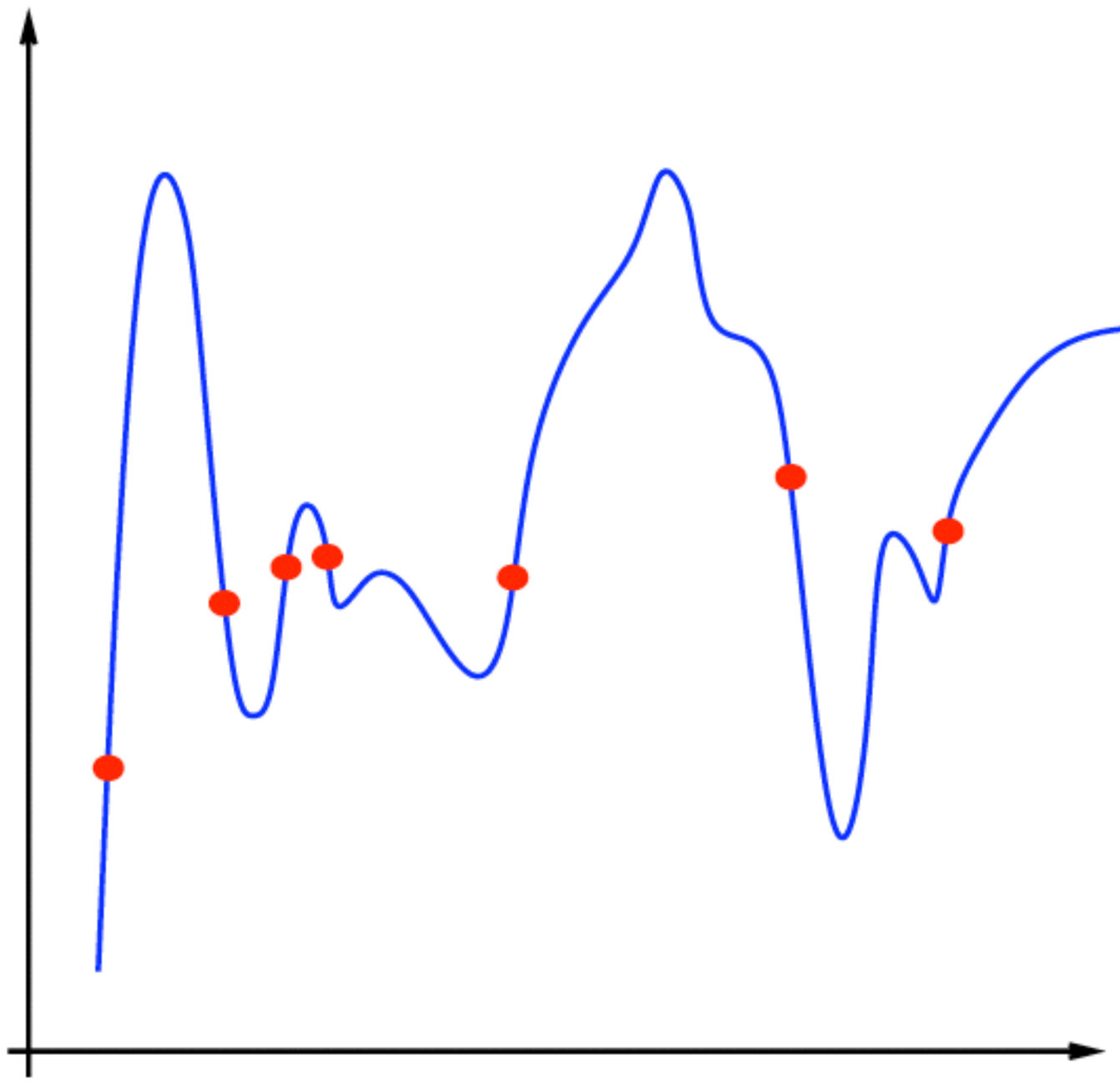
ERM and generalization: given a certain number of samples...



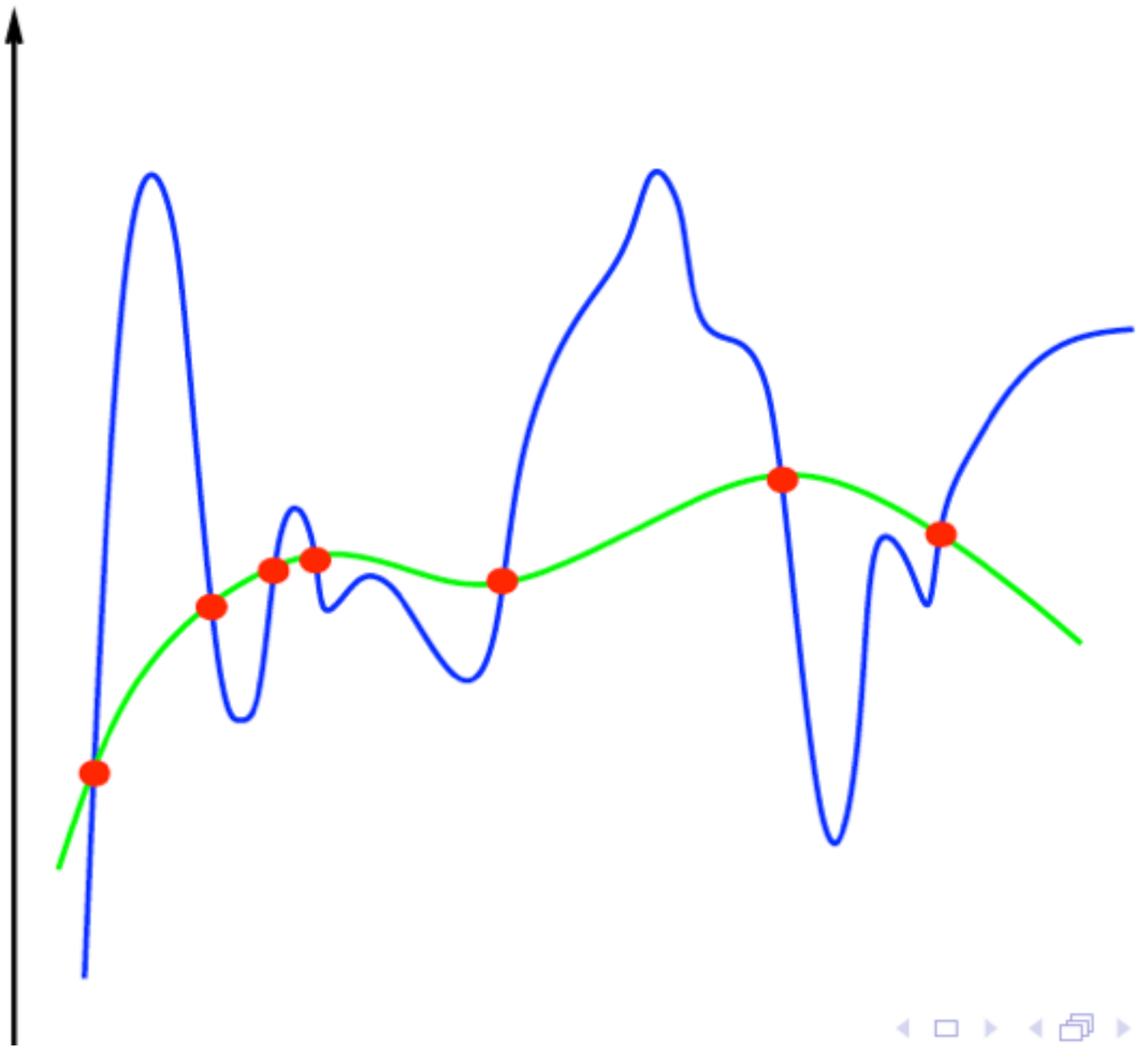
...suppose this is the “true” solution...



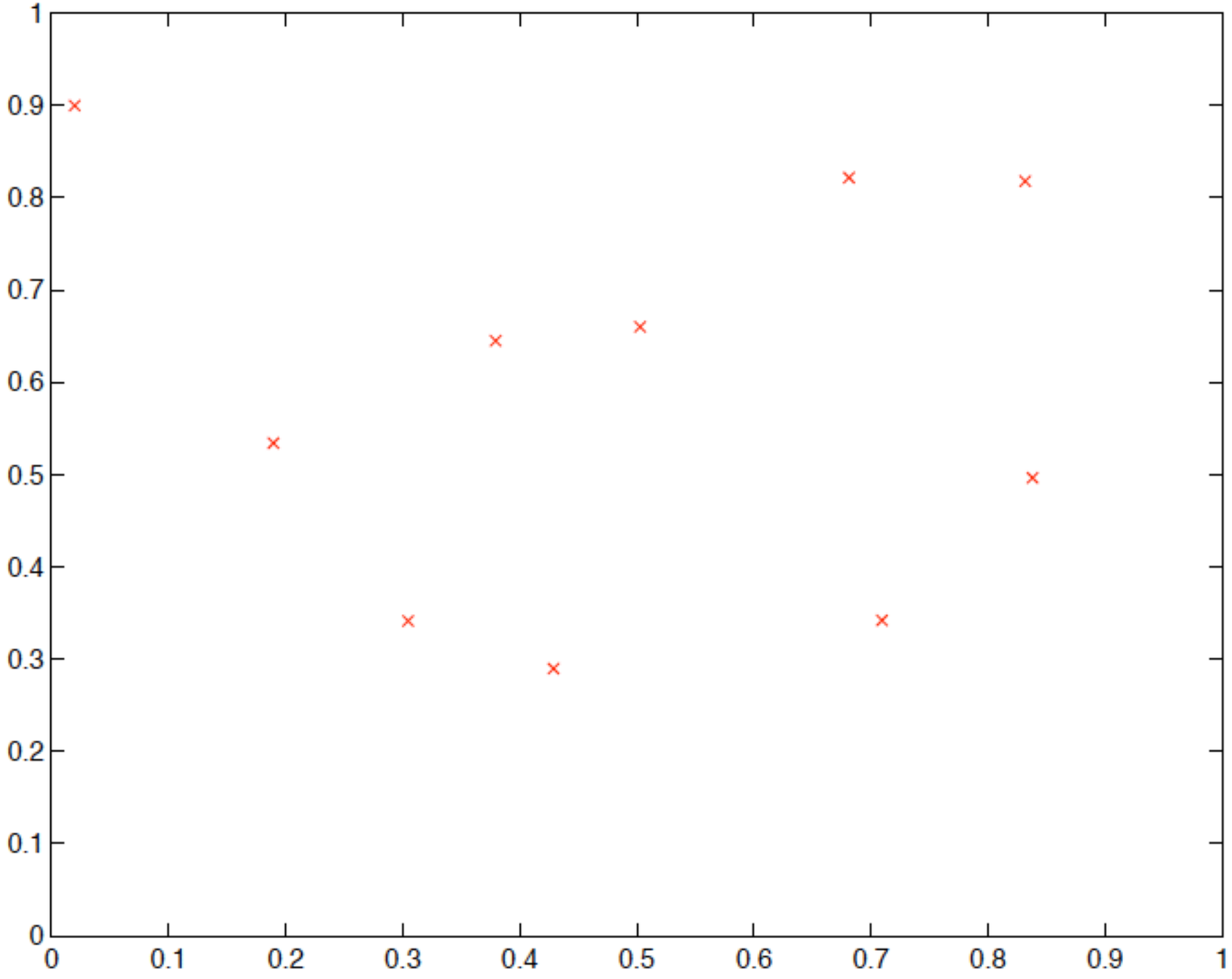
... but suppose ERM gives this solution.



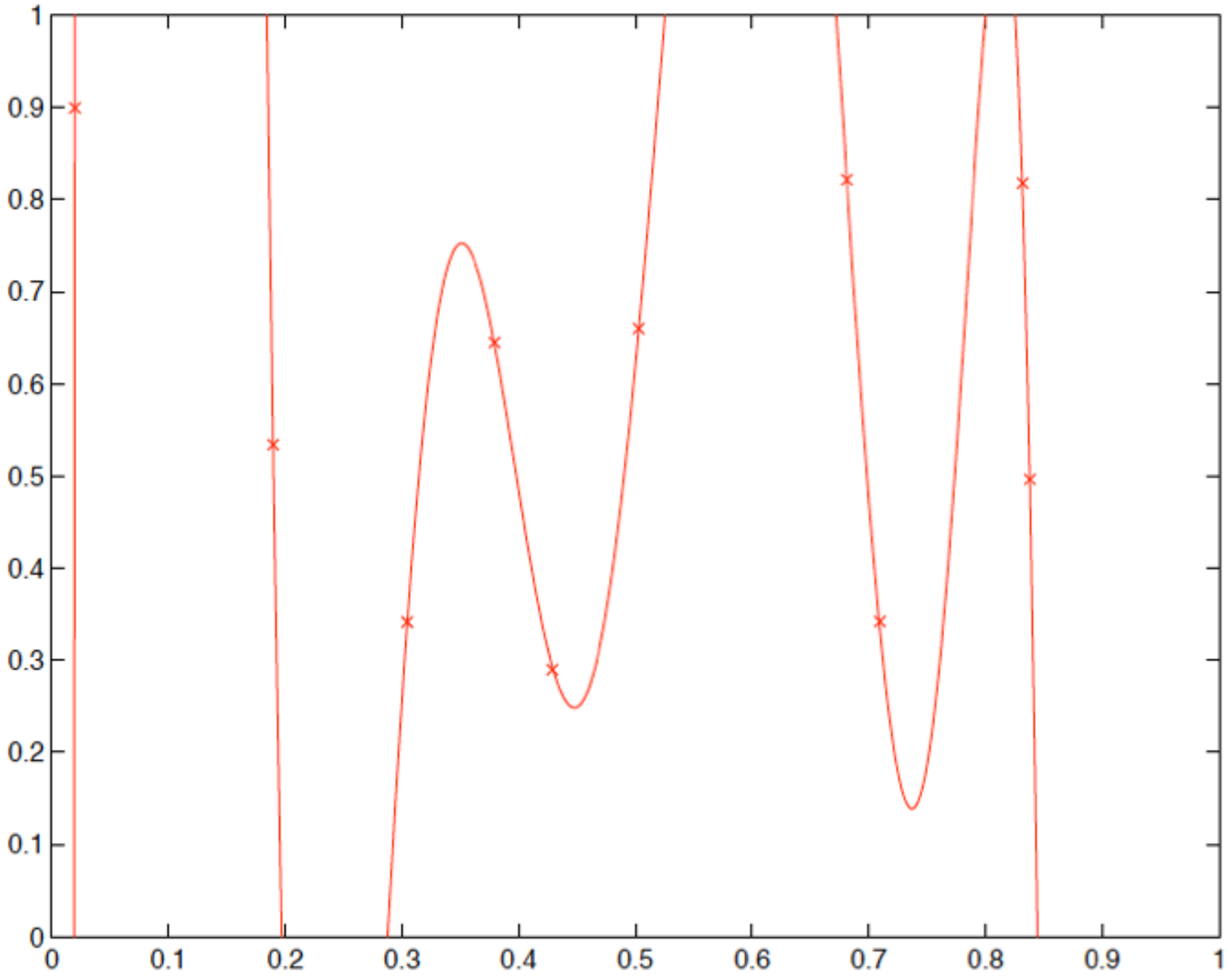
Under which conditions the ERM solution converges with increasing number of examples to the true solution? In other words...what are the conditions for generalization of ERM?



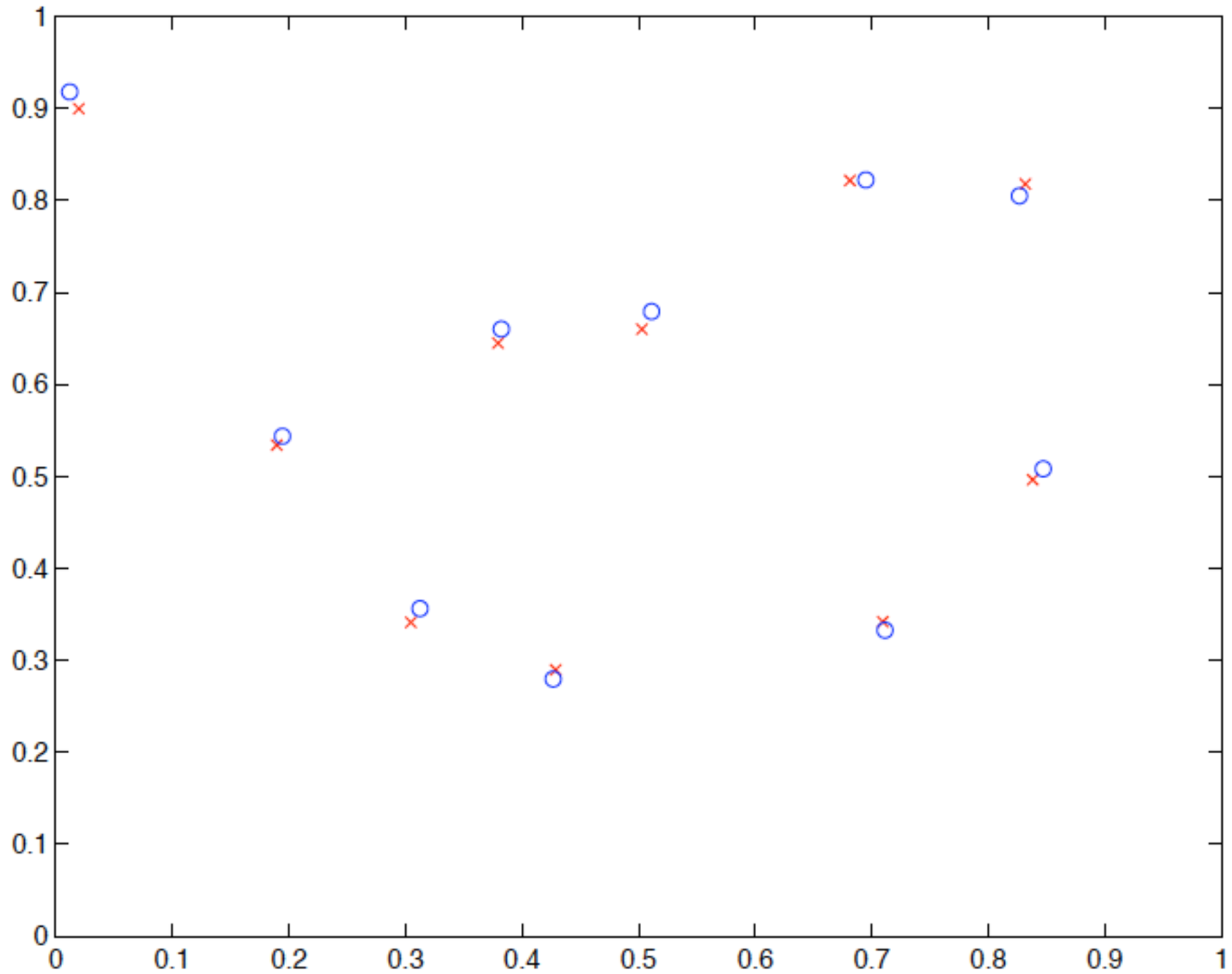
ERM and stability: given 10 samples...



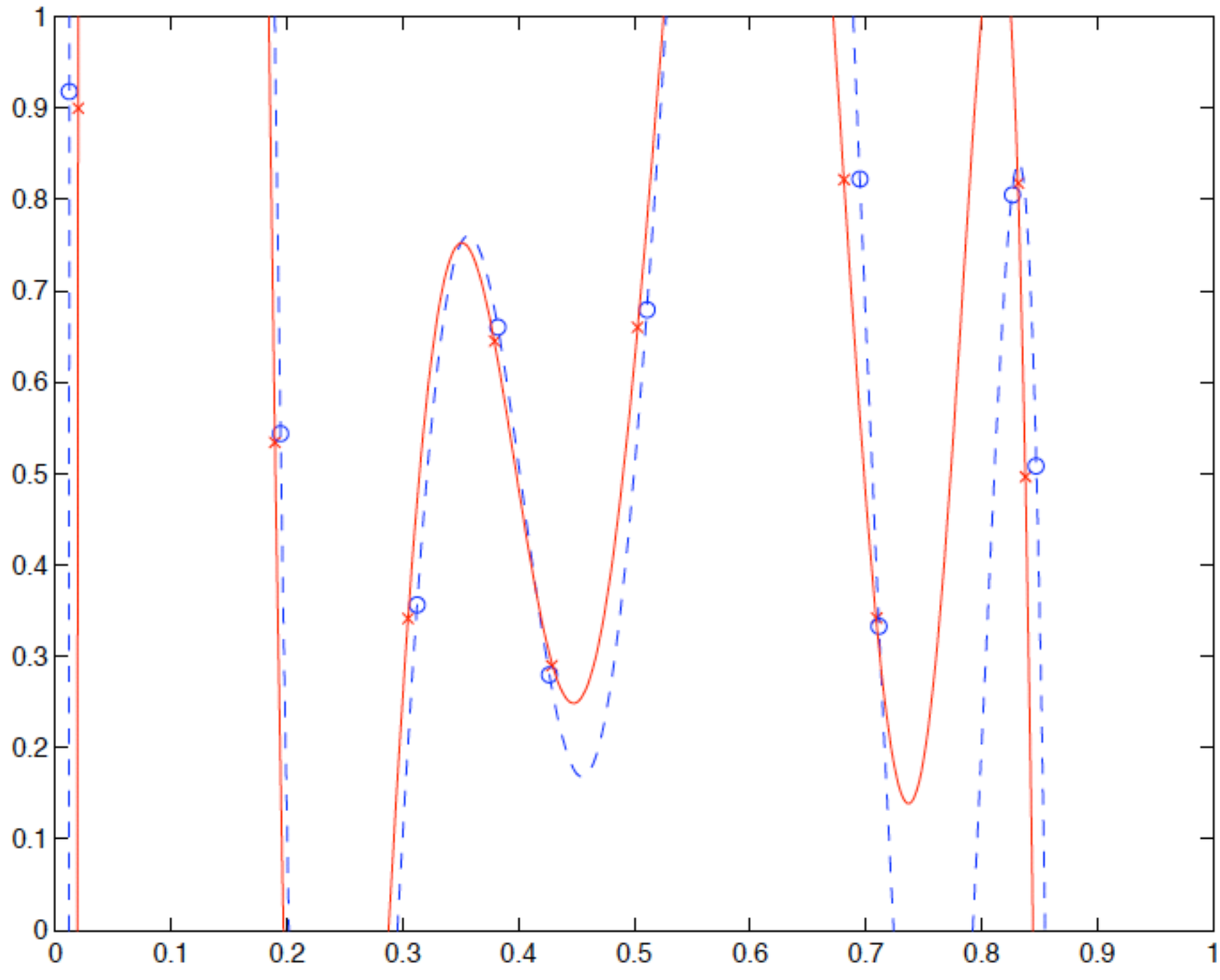
...we can find the smoothest interpolating polynomial (which degree?).



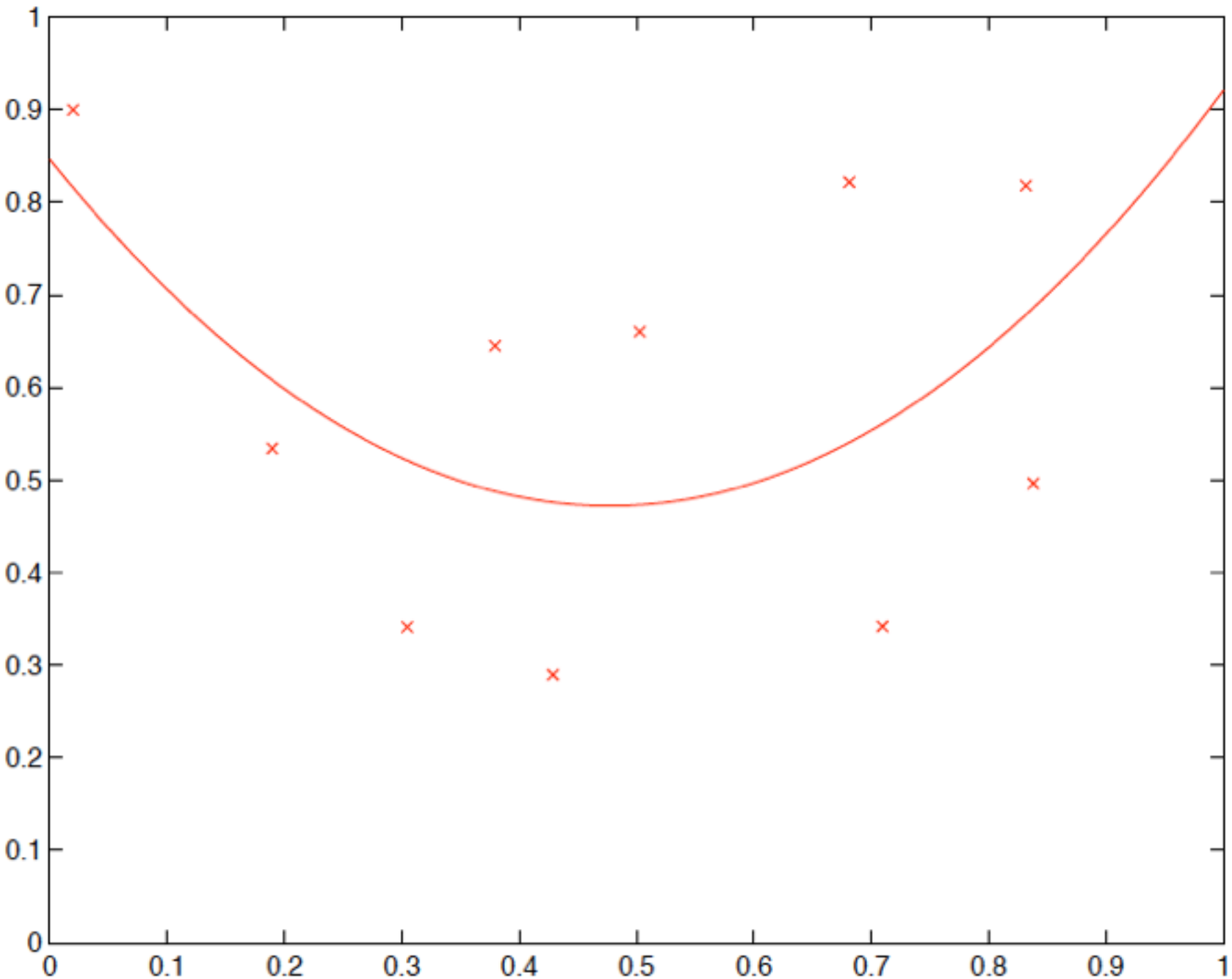
But if we perturb the points slightly...



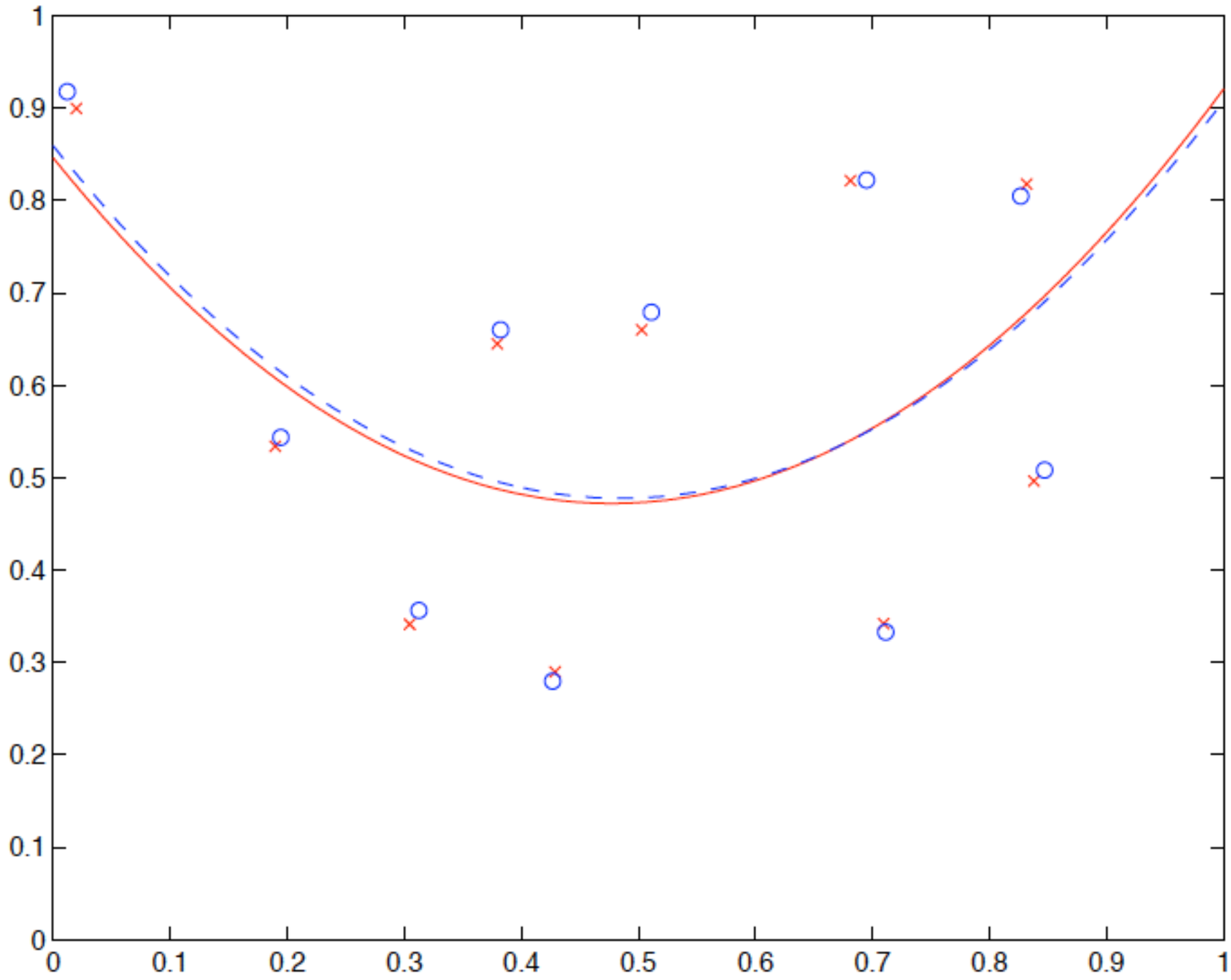
...the solution changes a lot!



If we restrict ourselves to degree two polynomials...



...the solution varies only a small amount under a small perturbation.



ERM: conditions for well-posedness (stability) and predictivity (generalization)

It turns out that with the appropriate definition of stability, *stability and generalization are equivalent for ERM.*

Thus the two desirable conditions for a supervised learning algorithm – generalization and stability – are equivalent (and they correspond to the same constraints on $\mathcal{H}$).

S training set, $S^{i,z}$ training set obtained replacing the i-th example in S with a new point $z = (x, y)$.

Definition

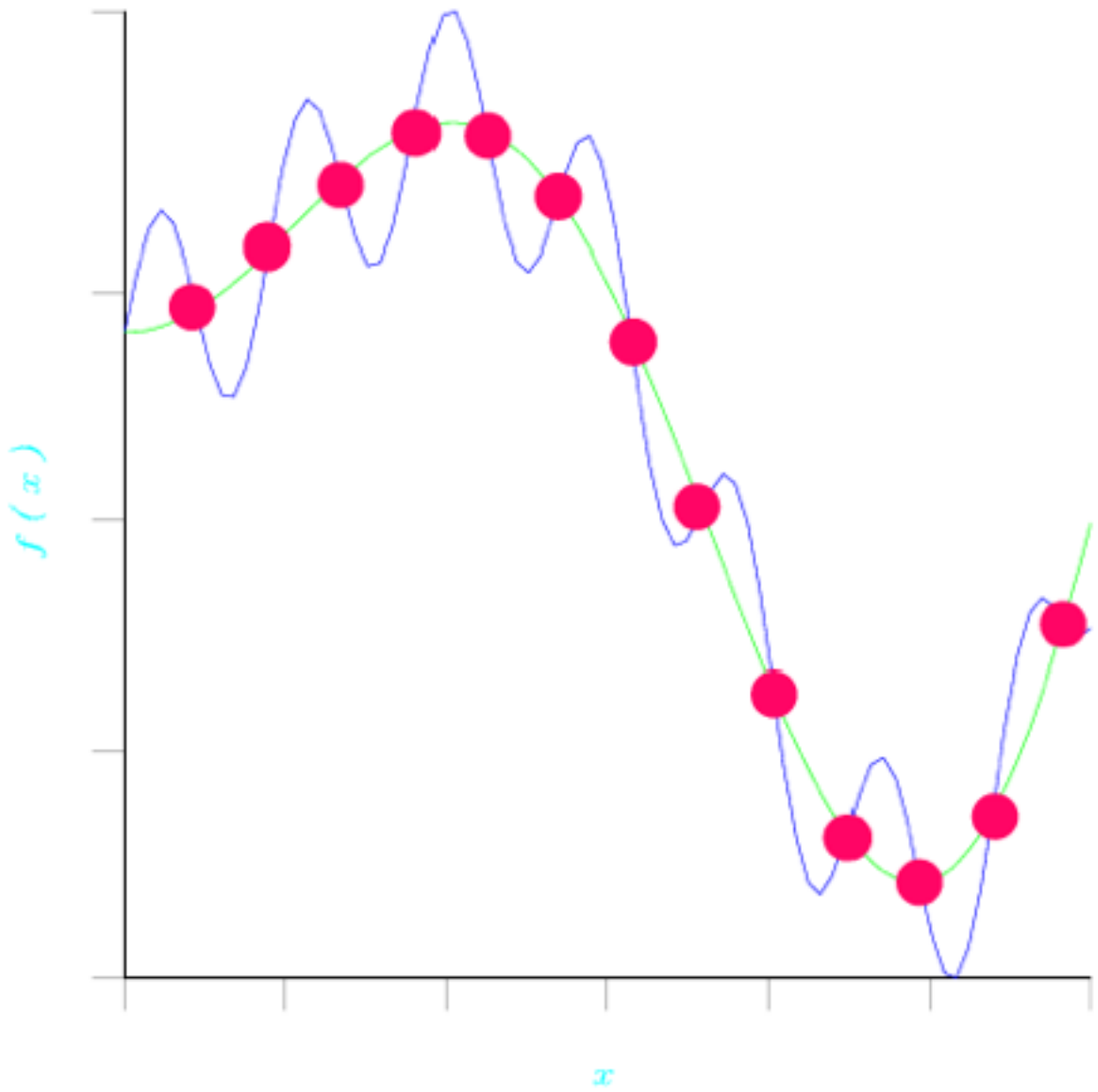
We say that an algorithm $\mathcal{A}$ has **uniform stability** β (is β-stable) if

$$\forall (S, z) \in \mathcal{Z}^{n+1}, \ \forall i, \ \sup_{z' \in Z} |V(f_S, z') - V(f_{S^{i,z}}, z')| \leq \beta.$$

This definition is sufficient for stability and for generalization (but it is strong)

ERM and ill-posedness

There are well-known approaches to re-estabilsh well-posedness.



Regularization

Regularization is the classical way to restore well posedness (and ensure generalization). Regularization in general means retricting H, as we have in fact done for ERM. There are two standard approaches in the field of ill-posed problems that ensure for ERM *well-posedness* (and *generalization*) by constraining the hypothesis space $\mathcal{H}$. The direct way – minimize the empirical error subject to f in a ball in an appropriate $\mathcal{H}$ – is called *Ivanov regularization*. The indirect way is *Tikhonov regularization* (which is not strictly ERM).

Ivanov and Tikhonov Regularization

ERM finds the function in $(\mathcal{H})$ which minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

which in general – for arbitrary hypothesis space $\mathcal{H}$ – is *ill-posed*.

- Ivanov regularizes by finding the function that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

while satisfying $\mathcal{R}(f) \leq A$.

- Tikhonov regularization minimizes over the hypothesis space $\mathcal{H}$, for a fixed positive parameter γ, the regularized functional

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \gamma\mathcal{R}(f). \tag{1}$$

$\mathcal{R}(f)$ is the regulirizer, a penalization on f. In this course we will mainly discuss the case $\mathcal{R}(f) = \|f\|_K^2$ where $\|f\|_K^2$ is the norm in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, defined by the kernel K.

Generalization, Sample Error and Approximation Error

Generalization error is $I_S[f_S] - I[f_S]$.
Sample error is $I[f_S] - I[f_{\mathcal{H}}]$
Approximation error is $I[f_{\mathcal{H}}] - I[f_0]$
Error is $I[f_S] - I[f_0] = (I[f_S] - I[f_{\mathcal{H}}]) + (I[f_{\mathcal{H}}] - I[f_0])$

Appendix: Target Space, Sample and Approximation Error

In addition to the hypothesis space $\mathcal{H}$, the space we allow our algorithms to search, we define...
The **target space** $\mathcal{T}$ is a space of functions, chosen a priori in any given problem, that is assumed to contain the "true" function f_0 that minimizes the risk. Often, $\mathcal{T}$ is chosen to be all functions in L_2, or all differentiable functions. Notice that the "true" function if it exists is defined by $\mu(z)$, which contains all the relevant information.

Sample Error (also called Estimation Error)

Let $f_{\mathcal{H}}$ be the function in $\mathcal{H}$ with the smallest true risk.
We have defined the **generalization error** to be $I_S[f_S] - I[f_S]$.
We define the **sample error** to be $I[f_S] - I[f_{\mathcal{H}}]$, the difference in true risk between the best function in $\mathcal{H}$ and the function in $\mathcal{H}$ we actually find. This is what we pay because our finite sample does not give us enough information to choose to the "best" function in $\mathcal{H}$. We'd like this to be small. *Consistency* – defined earlier – is equivalent to the sample error going to zero for $n \to \infty$.
A main goal in classical learning theory (Vapnik, Smale, ...) is "bounding" the generalization error. Another goal – for learning theory *and* statistics – is bounding the sample error, that is determining conditions under which we can state that $I[f_S] - I[f_{\mathcal{H}}]$ will be small (with high probability).
As a simple rule, we expect that if $\mathcal{H}$ is "well-behaved", then, as n gets large the sample error will become small.

Approximation Error

Let f_0 be the function in $\mathcal{T}$ with the smallest true risk.
We define the **approximation error** to be $I[f_{\mathcal{H}}] - I[f_0]$, the difference in true risk between the best function in $\mathcal{H}$ and the best function in $\mathcal{T}$. This is what we pay when $\mathcal{H}$ is smaller than $\mathcal{T}$. We'd like this error to be small too. In much of the following we can assume that $I[f_0] = 0$.

We will focus less on the approximation error in 9.520, but we will explore it.

As a simple rule, we expect that as $\mathcal{H}$ grows bigger, the approximation error gets smaller. If $\mathcal{T} \subseteq \mathcal{H}$ – which is a situation called *the realizable setting* –the approximation error is zero.

Approximation Error

Let f_0 be the function in $\mathcal{T}$ with the smallest true risk.
We define the **approximation error** to be $I[f_{\mathcal{H}}] - I[f_0]$, the difference in true risk between the best function in $\mathcal{H}$ and the best function in $\mathcal{T}$. This is what we pay when $\mathcal{H}$ is smaller than $\mathcal{T}$. We'd like this error to be small too. In much of the following we can assume that $I[f_0] = 0$.

We will focus less on the approximation error in 9.520, but we will explore it.

As a simple rule, we expect that as $\mathcal{H}$ grows bigger, the approximation error gets smaller. If $\mathcal{T} \subseteq \mathcal{H}$ – which is a situation called *the realizable setting* –the approximation error is zero.

Error

We define the **error** to be $I[f_S] - I[f_0]$, the difference in true risk between the function we actually find and the best function in $\mathcal{T}$. We'd really like this to be small. As we mentioned, often we can assume that the **error** is simply $I[f_S]$.
The error is the sum of the sample error and the approximation error:

$$I[f_S] - I[f_0] = (I[f_S] - I[f_\mathcal{H}]) + (I[f_\mathcal{H}] - I[f_0])$$

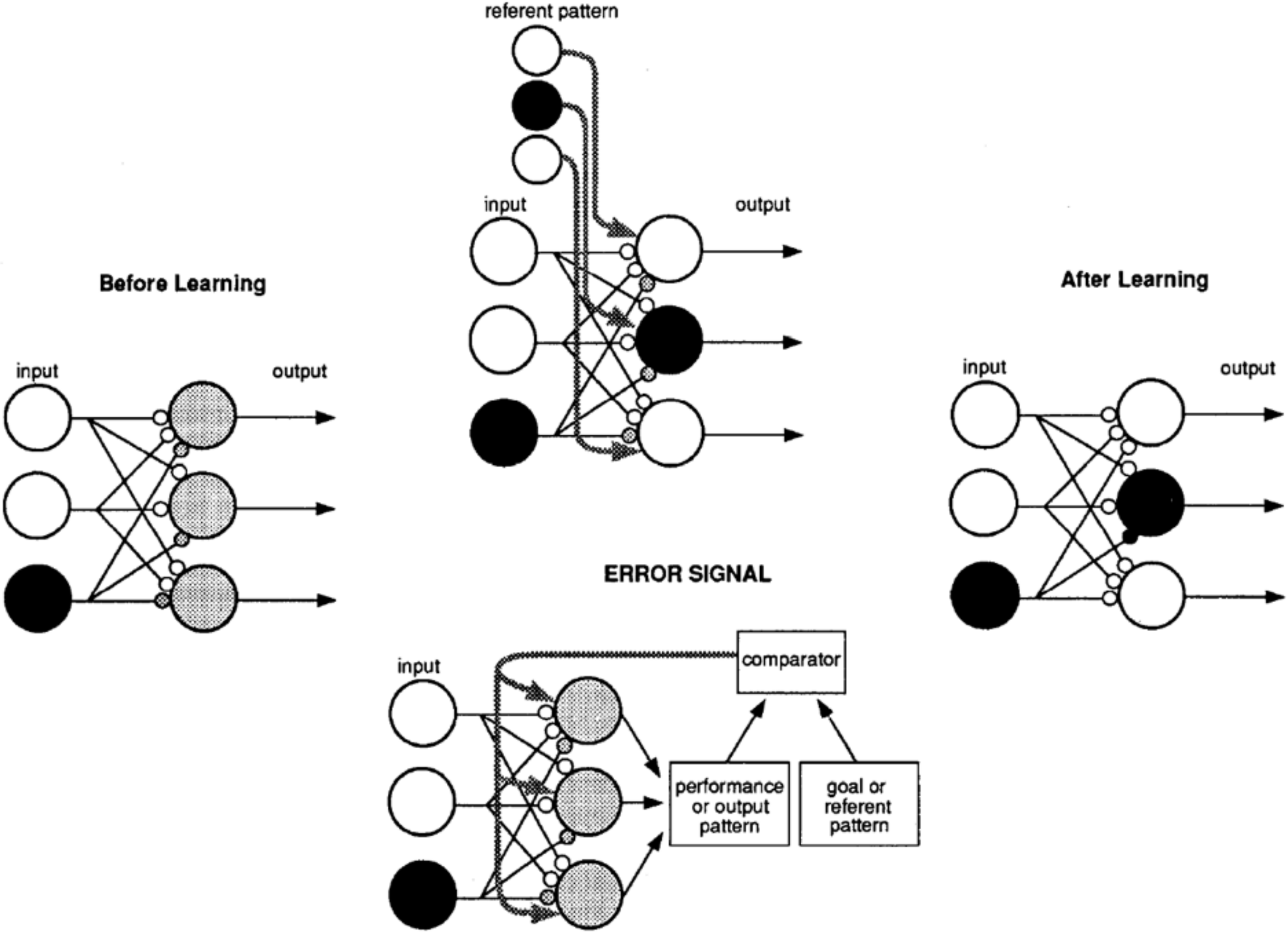
If we can make both the approximation and the sample error small, the error will be small. There is a tradeoff between the approximation error and the sample error...

The Approximation/Sample Tradeoff

It should already be intuitively clear that making $\mathcal{H}$ big makes the approximation error small. This implies that we can (help) make the error small by making $\mathcal{H}$ big.
On the other hand, we will show that making $\mathcal{H}$ small will make the sample error small. In particular for ERM, if $\mathcal{H}$ is a uGC class, the generalization error and the sample error will go to zero as $n \to \infty$, but how quickly depends directly on the "size" of $\mathcal{H}$. This implies that we want to keep $\mathcal{H}$ as small as possible. (Furthermore, $\mathcal{T}$ itself may or may not be a uGC class.)
Ideally, we would like to find the optimal tradeoff between these conflicting requirements.

Generalization, Sample Error and Approximation Error

Generalization error is $I_S[f_S] - I[f_S]$.
Sample error is $I[f_S] - I[f_{\mathcal{H}}]$
Approximation error is $I[f_{\mathcal{H}}] - I[f_0]$
Error is $I[f_S] - I[f_0] = (I[f_S] - I[f_{\mathcal{H}}]) + (I[f_{\mathcal{H}}] - I[f_0])$



Hebbian mechanisms can be used for biological supervised learning (Knudsen, 1990)