The Learning Problem and Regularization

9.520 Class 02, 12 February 2007

Tomaso Poggio

Plan

- Learning as function approximation
- Empirical Risk Minimization
- Generalization and Well-posedness
- Regularization
- Appendix: Sample and Approximation Error

About This Class

Theme We introduce the learning problem as the problem of function approximation from sparse data. We define the key ideas of loss functions, empirical error and generalization error. We then introduce the Empirical Risk Minimization approach and the two key requirements on algorithms using it: well-posedness and consistency. We then describe a key algorithm – Tikhonov regularization – that satisfies these requirements.

Math Required Familiarity with basic ideas in probability theory.

Data Generated By A Probability Distribution

We assume that X and Y are two sets of random variables. We are given a **training set** S consisting n samples drawn i.i.d. from the probability distribution $\mu(z)$ on $Z = X \times Y$:

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

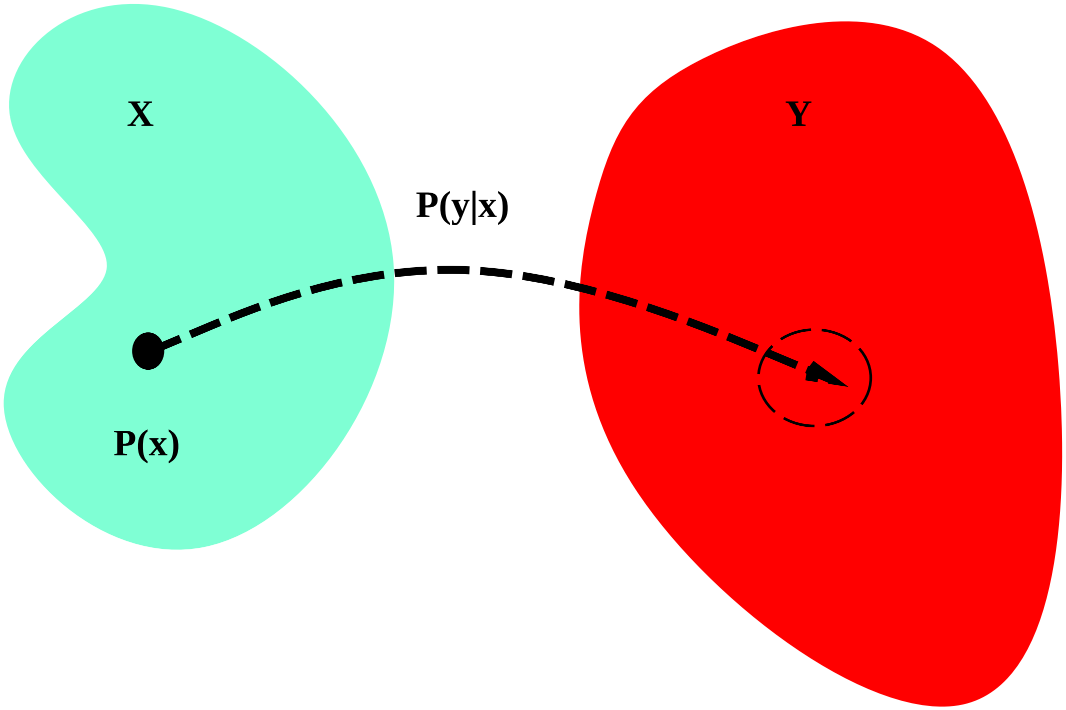
that is $z_1, \ldots, z_n$

We will make frequent use of the **conditional probability of y given x**, written $p(y|x)$:

$$\mu(z) = p(x, y) = p(y|x) \cdot p(x)$$

It is crucial to note that we view $p(x, y)$ as **fixed** but **unknown**.

Probabilistic setting



Hypothesis Space

The **hypothesis space** $\mathcal{H}$ is the space of functions that we allow our algorithm to provide. For many algorithms (such as optimization algorithms) it the space the algorithm is allowed to search. As we will see, it is often important to choose the hypothesis space as a function of the amount of data available.

Learning As Function Approximation From Samples: Regression and Classification

The basic goal of **supervised learning** is to use the training set S to "learn" a function f_S that looks at a new x value x_{new} and predicts the associated value of y:

$$y_{pred} = f_S(x_{new})$$

If y is a real-valued random variable, we have **regression**.

If y takes values from an unordered finite set, we have **pattern classification**. In two-class pattern classification problems, we assign one class a y value of 1, and the other class a y value of -1.

Loss Functions

In order to measure goodness of our function, we need a **loss function** V. In general, we let $V(f, z) = V(f(x), y)$ denote the price we pay when we see x and guess that the associated y value is $f(x)$ when it is actually y.

Common Loss Functions For Regression

For regression, the most common loss function is square loss or L2 loss:

$$V(f(x), y) = (f(x) - y)^2$$

We could also use the absolute value, or L1 loss:

$$V(f(x), y) = |f(x) - y|$$

Vapnik's more general ϵ-insensitive loss function is:

$$V(f(x), y) = (|f(x) - y| - \epsilon)_+$$

Common Loss Functions For Classification

For binary classification, the most intuitive loss is the 0-1 loss:

$$V(f(x), y) = \Theta(-yf(x))$$

where $\Theta(-yf(x))$ is the step function. For tractability and other reasons, we often use the hinge loss (implicitely introduced by Vapnik) in binary classification:

$$V(f(x), y) = (1 - y \cdot f(x))_+$$

The learning problem: summary so far

There is an unknown **probability distribution** on the product space $Z = X \times Y$, written $\mu(z) = \mu(x, y)$. We assume that X is a compact domain in Euclidean space and Y a closed subset of $\mathbb{R}$.

The **training set** $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\} = \{z_1, ...z_n\}$ consists of n samples drawn i.i.d. from μ.

$\mathcal{H}$ is the **hypothesis space**, a space of functions $f : X \rightarrow Y$.

A **learning algorithm** is a map $L : Z^n \rightarrow \mathcal{H}$ that looks at S and selects from $\mathcal{H}$ a function $f_S : \mathbf{x} \rightarrow y$ such that $f_S(\mathbf{x}) \approx y$ *in a predictive way*.

Empirical error, generalization error, generalization

Given a function f, a loss function V, and a probability distribution μ over Z, the **expected or true error** of f is:

$$I[f] = \mathbb{E}_z V[f, z] = \int_Z V(f, z) d\mu(z)$$

which is the **expected loss** on a new example drawn at random from μ.

We would like to make $I[f]$ small, but in general we do not know μ.

Given a function f, a loss function V, and a training set S consisting of n data points, the **empirical error** of f is:

$$I_S[f] = \frac{1}{n} \sum V(f, z_i)$$

Empirical error, generalization error, generalization

A very natural requirement for f_S is distribution independent **generalization**

$$\forall \mu, \lim_{n \to \infty} |I_S[f_S] - I[f_S]| = 0 \textit{ in probability}$$

In other words, the training error for the solution must converge to the expected error and thus be a "proxy" for it. Otherwise the solution would not be "predictive".

A desirable additional requirement is **universal consistency**

$$\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P}_S \left\{ I[f_S] > \inf_{f \in \mathcal{H}} I[f] + \varepsilon \right\} = 0.$$

Remark: For some of the rsults to be mentioned the requirement of uniform convergence must be added in both definitions.

A reminder: convergence in probability

Let $\{X_n\}$ be a sequence of bounded random variables. We say that

$$\lim_{n\to\infty} X_n = X \textit{ in probability}$$

if

$$\forall\varepsilon > 0 \lim_{n\to\infty} \mathbb{P}\{|X_n - X| \geq \varepsilon\} = 0.$$

or

if for each n there exists a ε_n and a δ_n such that

$$\mathbb{P}\left\{|X_n - X| \geq \varepsilon_n\right\} \leq \delta_n,$$

with ε_n and δ_n going to zero for $n \to \infty$.

3. ERM and conditions for generalization (and consistency)

Given a training set S and a function space $\mathcal{H}$, empirical risk minimization (Vapnik introduced the term) is the algorithm that looks at S and selects f_S as

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

The solution to this problem does not in general exhibit generalization and the problem may be **ill-posed**, depending on the choice of $\mathcal{H}$.

If the minimum does not exist we can work with the infimum.

Notice: *For ERM generalization and consistency are equivalent*

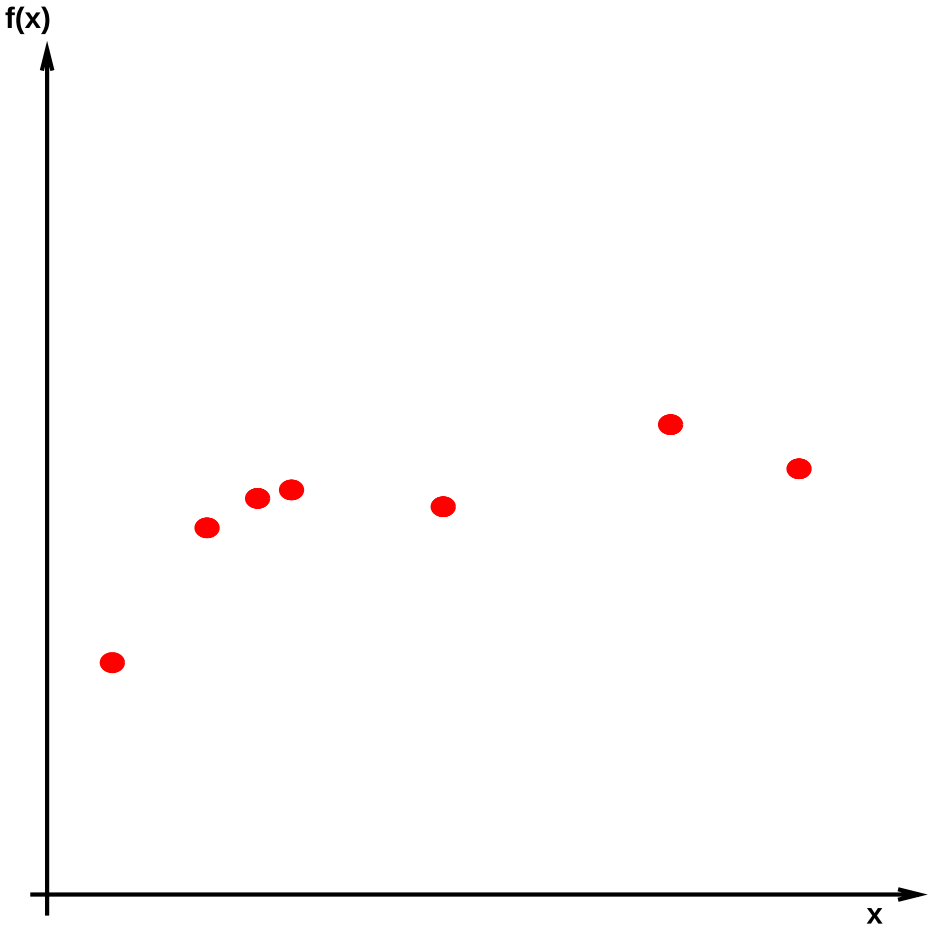
Generalization and Well-posedness of Empirical Risk Minimization

For the solution of ERM to be useful in the context of learning, the solution must

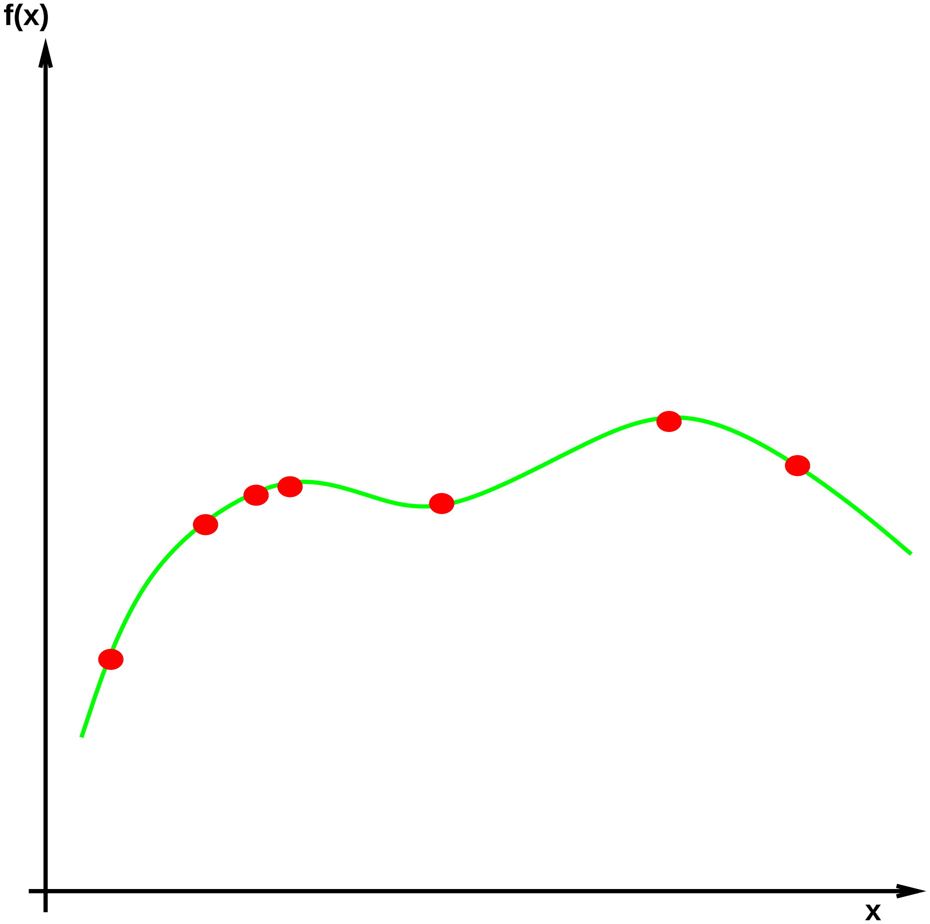
- “generalize”

- exist, be unique and be “stable” (well-posedness).

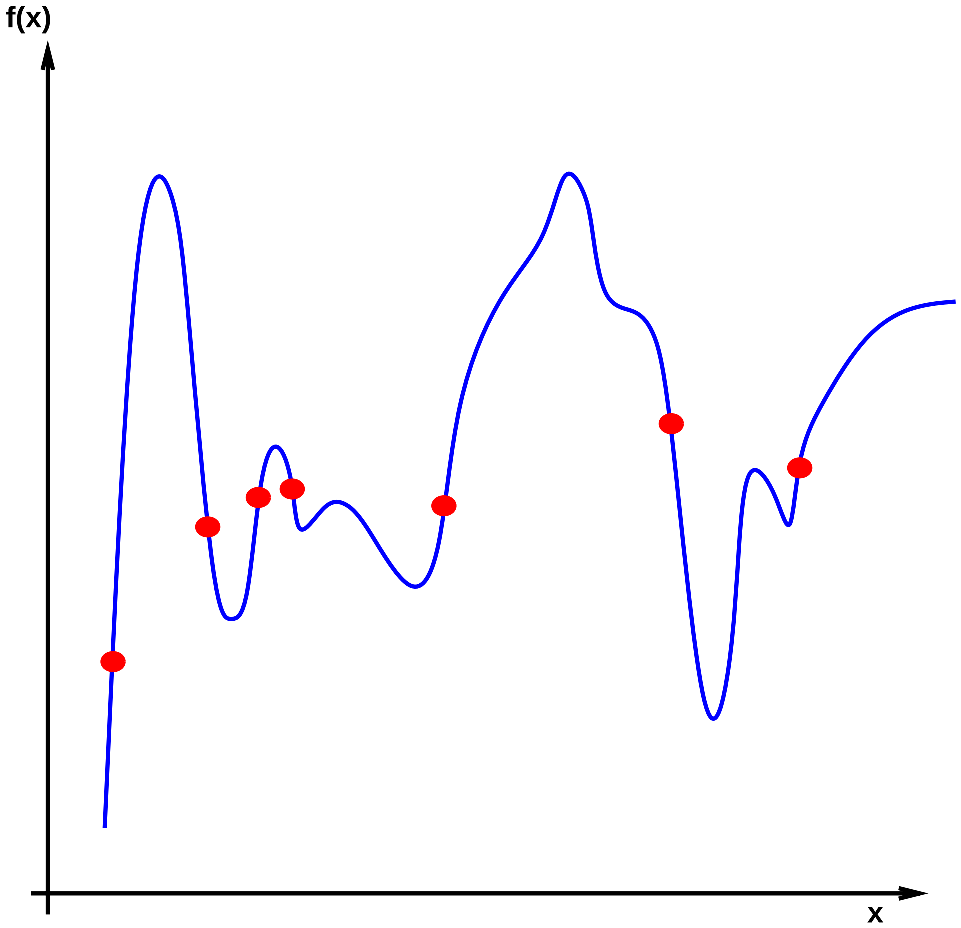
Here is a graphical example for generalization: given a certain number of samples...



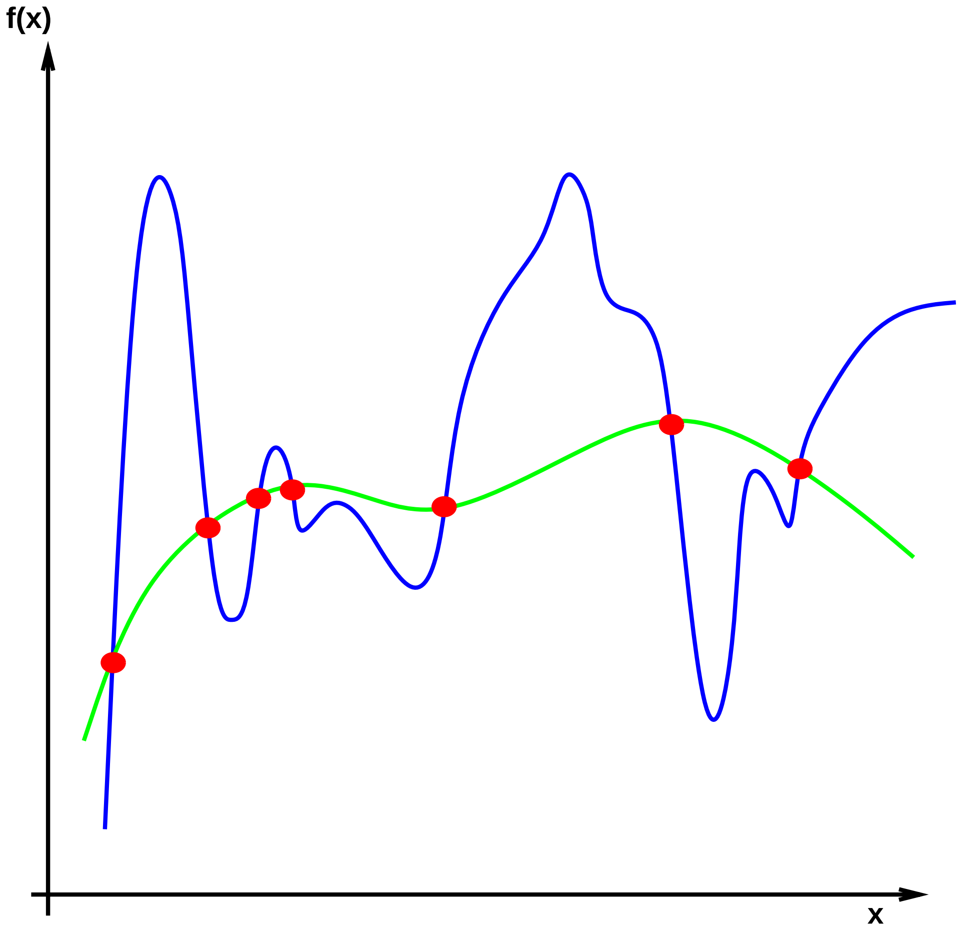
suppose this is the “true” solution...



... but suppose ERM gives this solution!



How can I guarantee that for a sufficient number of examples the ERM solution will converge to the true one?



Classical conditions for consistency of ERM

Uniform Glivenko-Cantelli Classes

$\mathcal{L} = \{\mathcal{H} \circ V\}$ is a (weak) uniform Glivenko-Cantelli (uGC) class

if

$$\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P}_S \left\{ \sup_{\ell \in \mathcal{L}} |I[\ell] - I_S[\ell]| > \varepsilon \right\} = 0.$$

Theorem [Vapnik and Červonenkis (71), Alon et al (97), Dudley, Giné, and Zinn (91)]

A necessary and sufficient condition for consistency of ERM is that $\mathcal{L}$ is uGC.

Thus, as we will see later, a proper choice of the hypothesis space $\mathcal{H}$ ensures generalization of ERM (and consistency since for ERM generalization is necessary and sufficient for consistency and viceversa). We will be exploring the uGC definition (and equivalent definitions) in detail in 9.520.

Well-posedness of ERM

ERM is in general an ill-posed problem. It can be made well-posed by an appropriate choice of $\mathcal{H}$.

As we will see later, well-posedness is mainly used to mean *stability* of the solution: f_S depends 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.

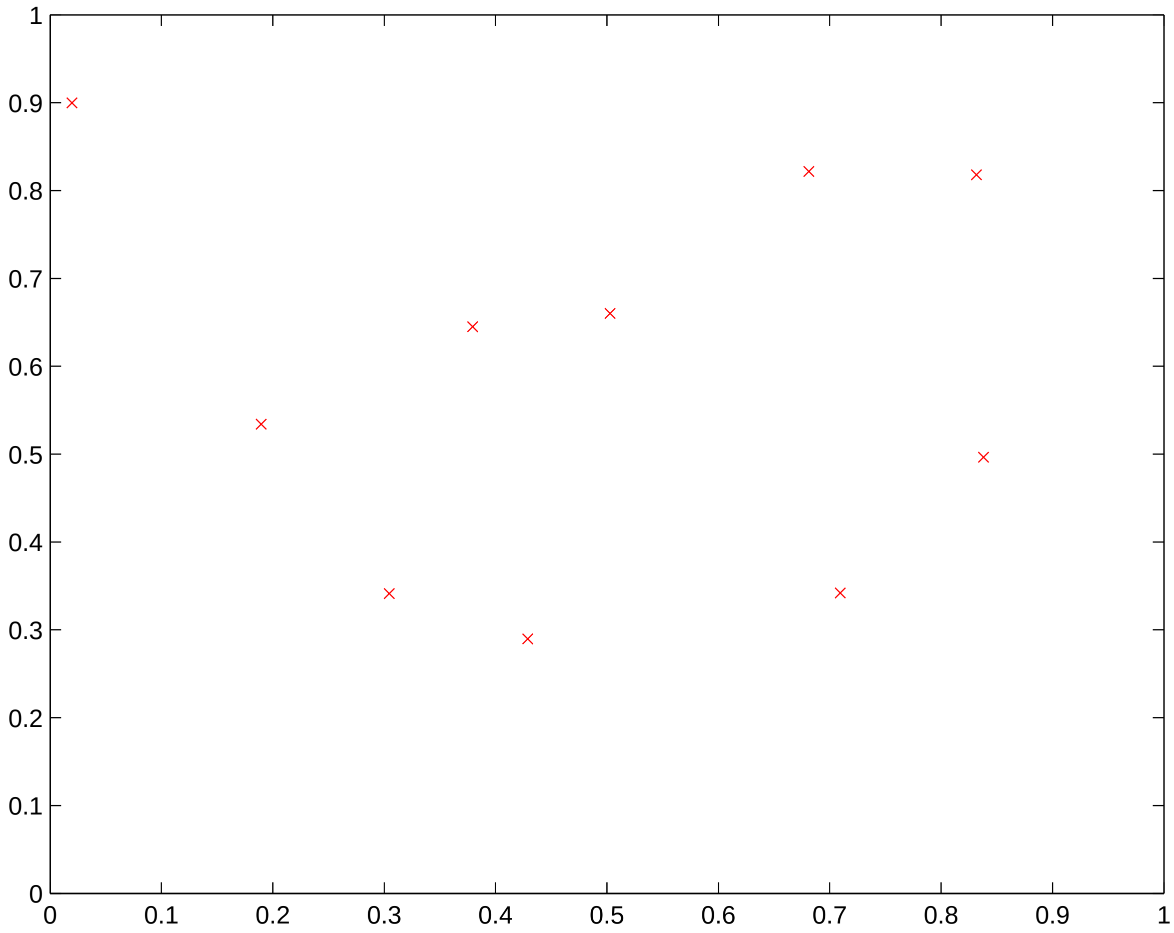
General definition of Well-Posed and Ill-Posed problems

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

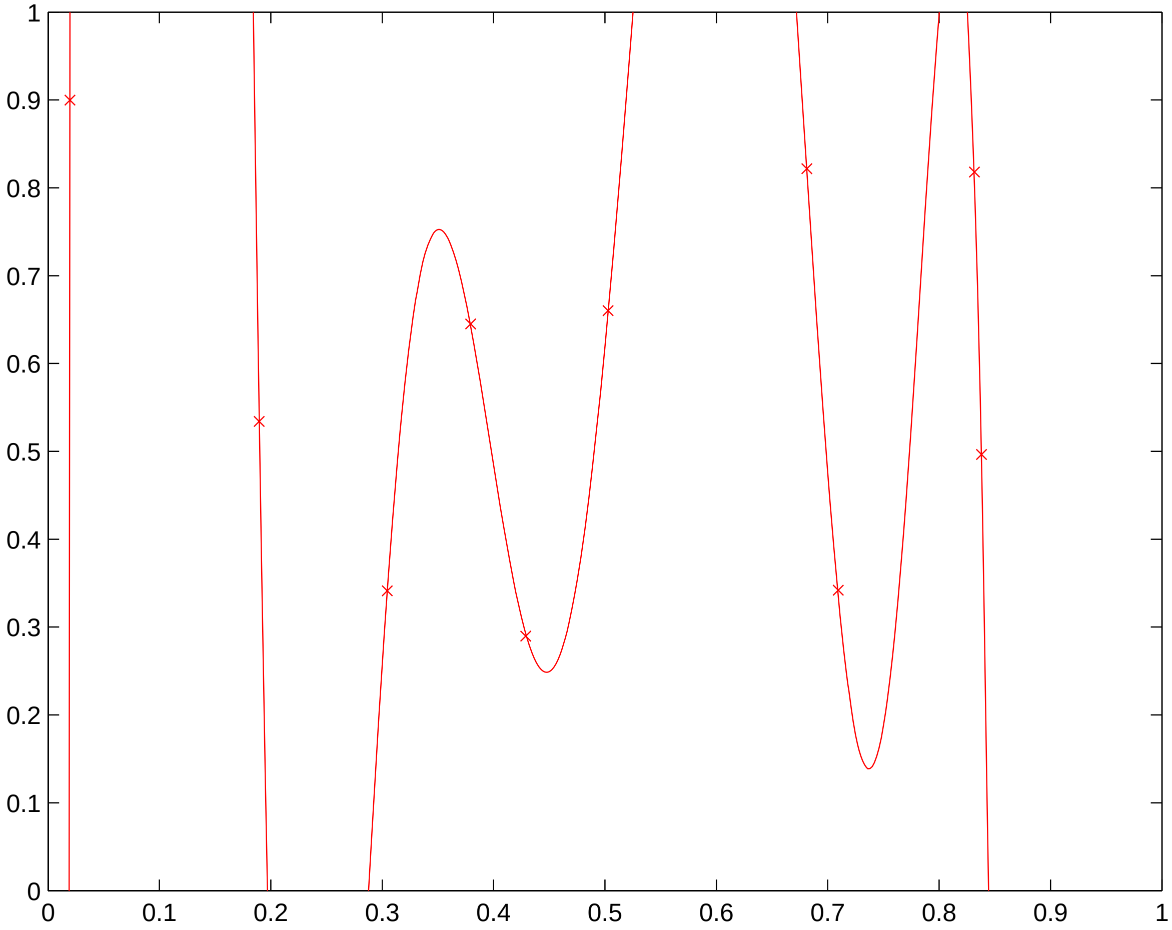
- exists
- is unique
- depends continuously on the data (e.g. it is *stable*)

A problem is **ill-posed** if it is not well-posed.

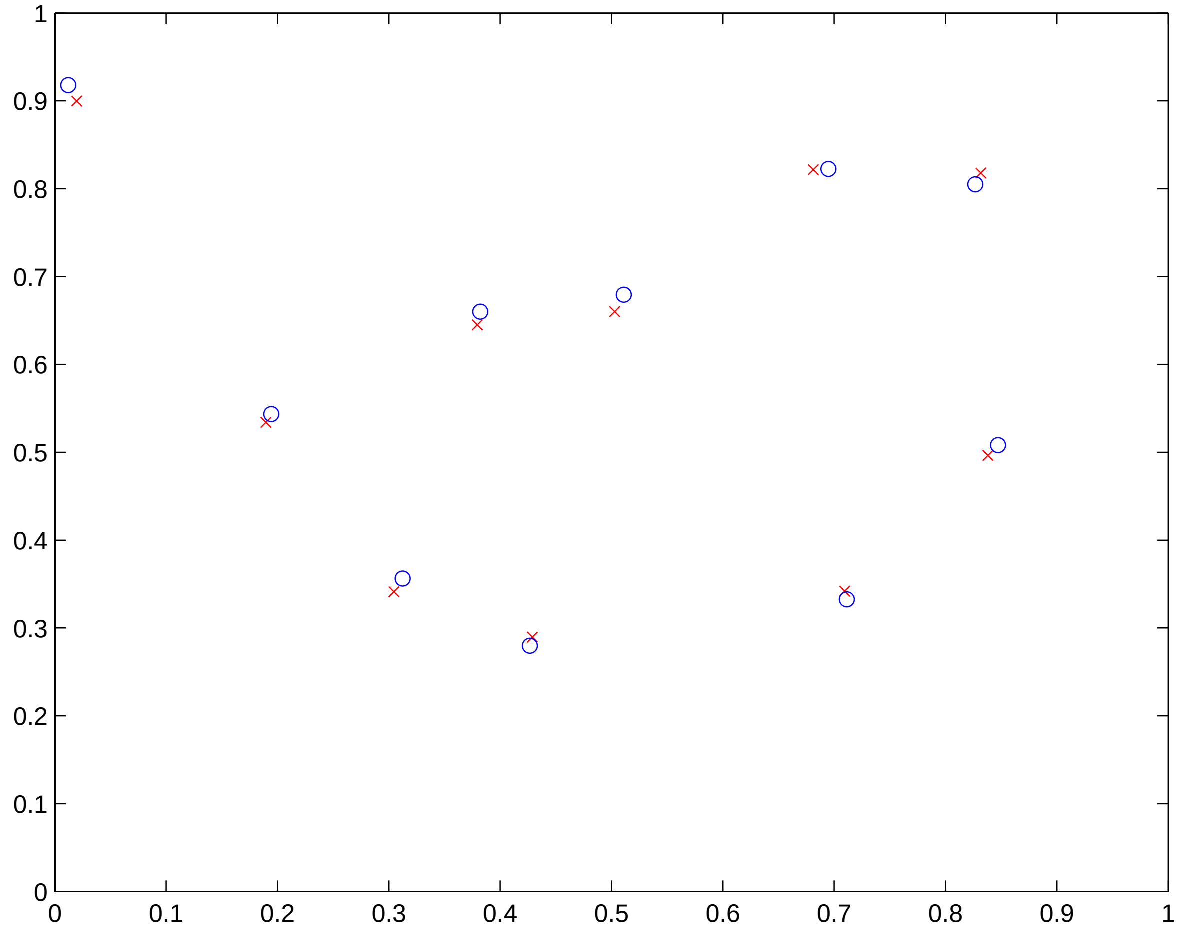
Here is a graphical example for stability: given 10 samples...



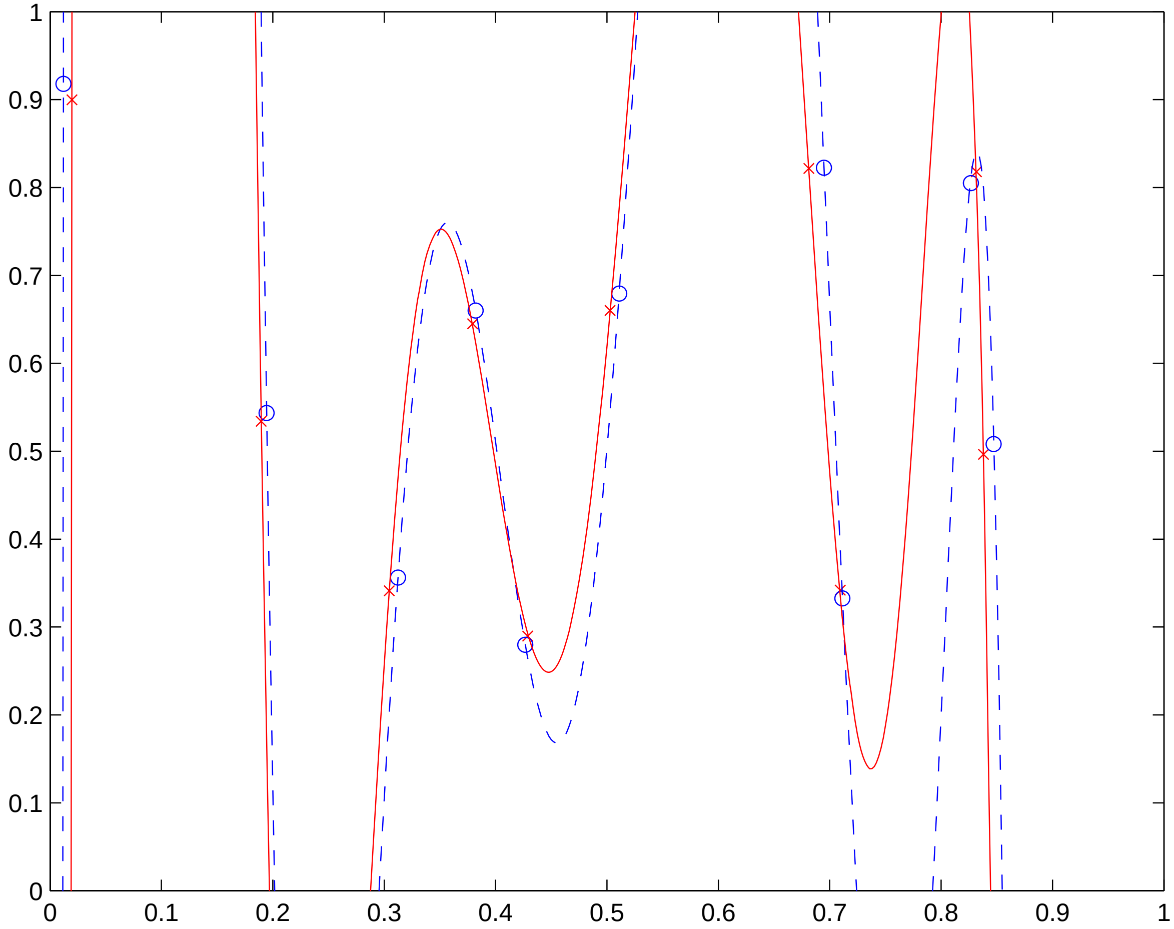
...we can find the smoothest interpolating polynomial (which degree?).



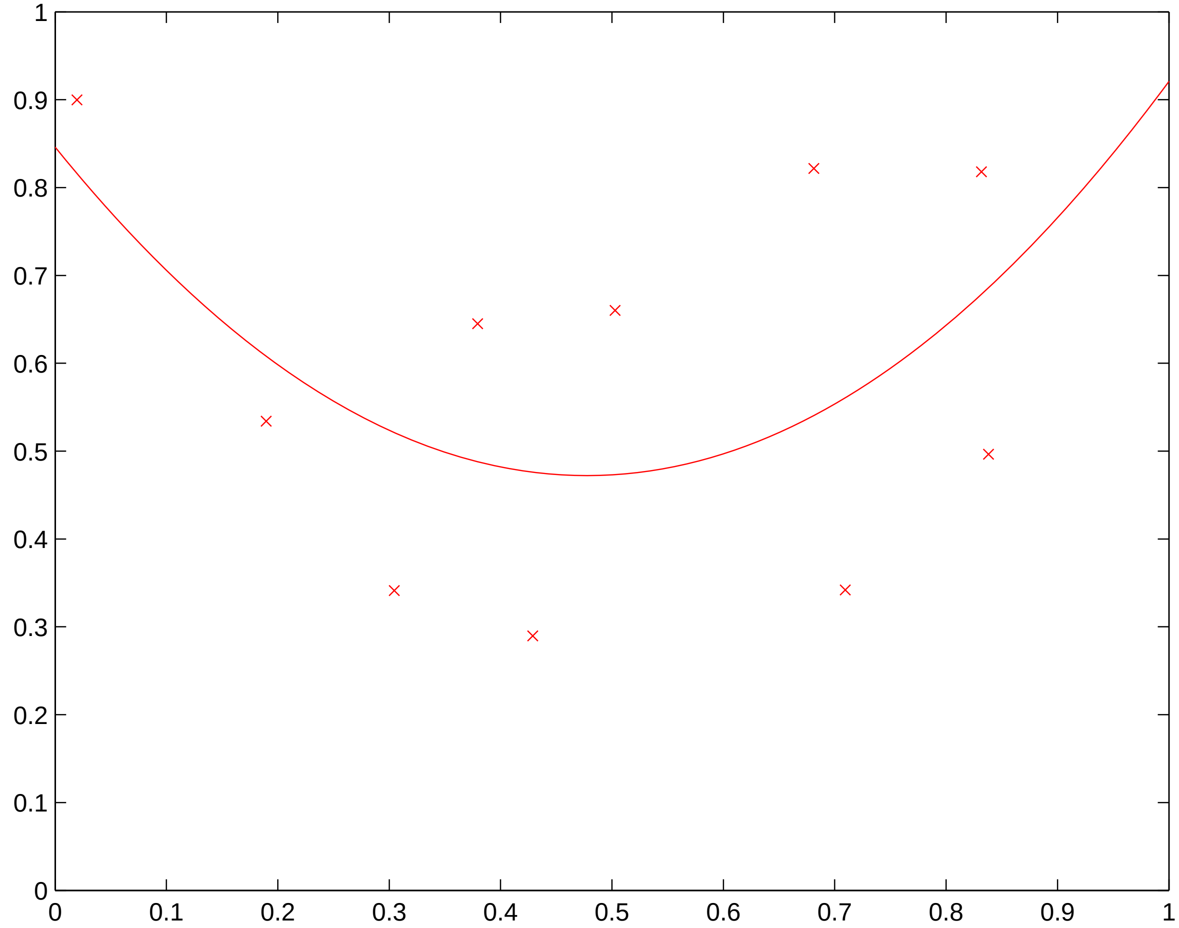
But if we perturb the points slightly...



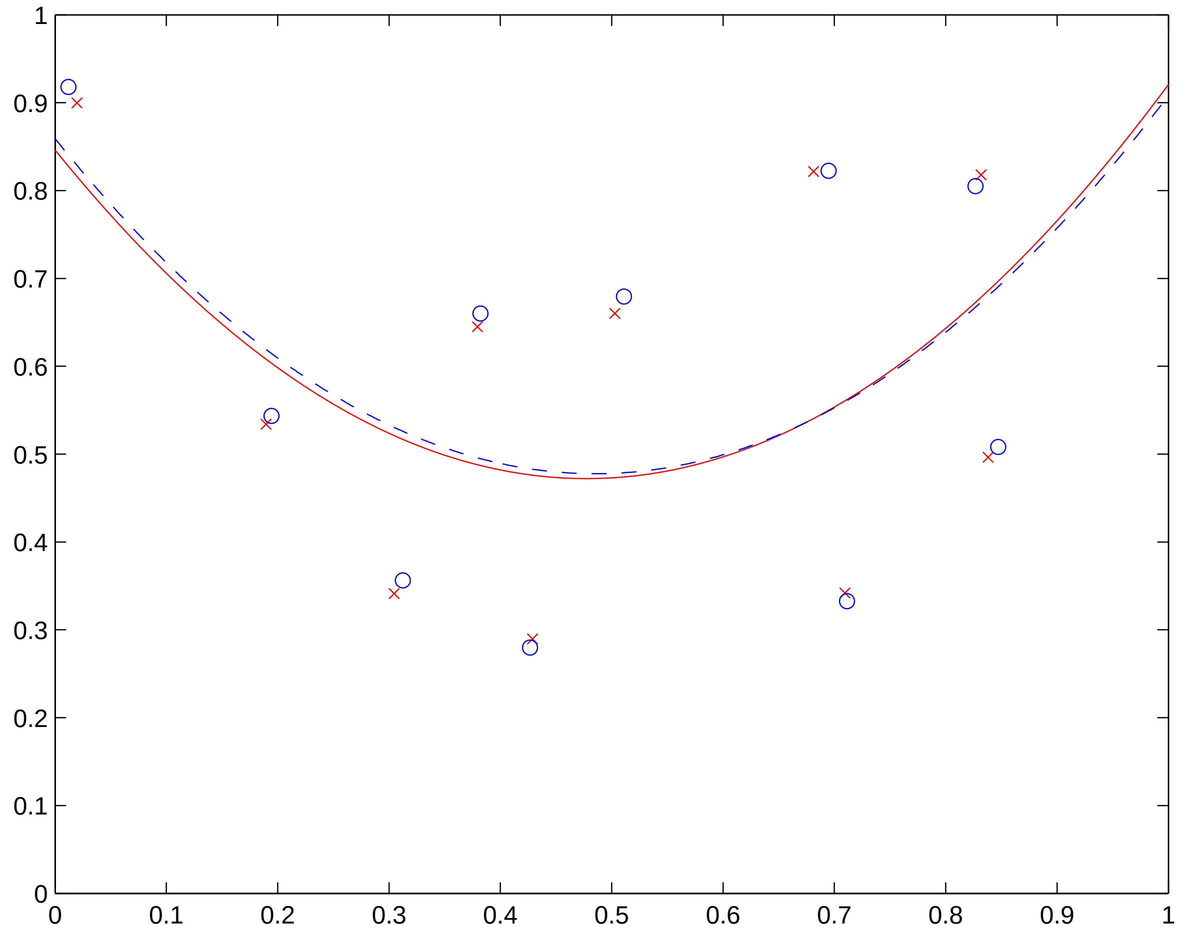
...the solution changes a lot.



If we restrict ourselves to degree two polynomials...



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



Regularization

The basic idea of regularization (originally introduced by Tikhonov independently of the learning problem) is to restore well-posedness of ERM 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 ERM).

Ivanov and Tikhonov Regularization

ERM finds the function in $(\mathcal{H}, \|\cdot\|)$ 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

$$\|f\|^2 \leq A$$

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

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \gamma\|f\|_K^2, \tag{1}$$

where $\|f\|_K$ is the norm in $\mathcal{H}$ – the Reproducing Kernel Hilbert Space (RKHS), defined by the kernel K.

Well-posed and Ill-posed problems

Hadamard introduced the definition of ill-posedness. Ill-posed problems are typically inverse problems.

As an example, assume g is a function in Y and u is a function in X, with Y and X Hilbert spaces. Then given the linear, continuous operator L, consider the equation

$$g = Lu.$$

The direct problem is is to compute g given u; the inverse problem is to compute u given the data g. In the learning case L is somewhat similar to a "sampling" operation and the inverse problem becomes the problem of finding a function that takes the values

$$f(x_i) = y_i, i = 1, ...n$$

The inverse problem of finding u is well-posed when

- the solution exists,
- is unique and
- is *stable*, that is depends continuously on the initial data g.

Ill-posed problems fail to satisfy one or more of these criteria. Often the term ill-posed applies to problems that are **not stable**, which in a sense is the key condition.

Tikhonov Regularization

As we will see in future classes

- Tikhonov regularization ensures well-posedness eg existence, uniqueness and especially *stability* (in a very strong form) of the solution

- Tikhonov regularization ensures generalization

- Tikhonov regularization is closely related to – but different from – Ivanov regularization, eg ERM on a hypothesis space $\mathcal{H}$ which is a ball in a RKHS.

Next Class

- In the next class we will introduce RKHS: they will be the hypothesis spaces we will work with.

- We will also derive the solution of Tikhonov regularization.

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 Errror

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 because $\mathcal{H}$ 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 $n \rightarrow \infty$, 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.

Error Definition: a caveat

We define the **error** to be $I[f_S] - I[f_0]$. In the literature, the true risk of the function we find, $I[f_S]$ is sometimes called the error. In the case where $I[f_0] = 0$, the two definitions are equivalent.

...mapping notation and results in CuckerSmale...

$$\epsilon(f) \longleftrightarrow I(f)$$

$$\epsilon_z(f) \longleftrightarrow I_S(f)$$

Thus

$$L_z \longleftrightarrow I(f) - I_S(f)$$

For ERM

$$f_z \longleftrightarrow f_S$$

Theorem B (for $\mathcal{H}$ compact) $\longleftrightarrow$ *generalization*, see Theorem b (Nature; for general algorithms and general $\mathcal{H}$)

Theorem C (eg $\epsilon_{\mathcal{H}}(f_z) \to 0$) $\longleftrightarrow$ Theorem a (Nature; consistency of ERM) where $\epsilon_{\mathcal{H}}(f) = \epsilon(f) - \epsilon(f_{\mathcal{H}})$,