The Learning Problem and Regularization

9.520 Class 02, 10 February 2003

Tomaso Poggio and Ryan Rifkin

Plan

- Learning as function approximation
- Empirical Risk Minimization
- Well-posedness and consistency
- 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 ℓ samples drawn i.i.d. from the probability distribution $X \times Y$:

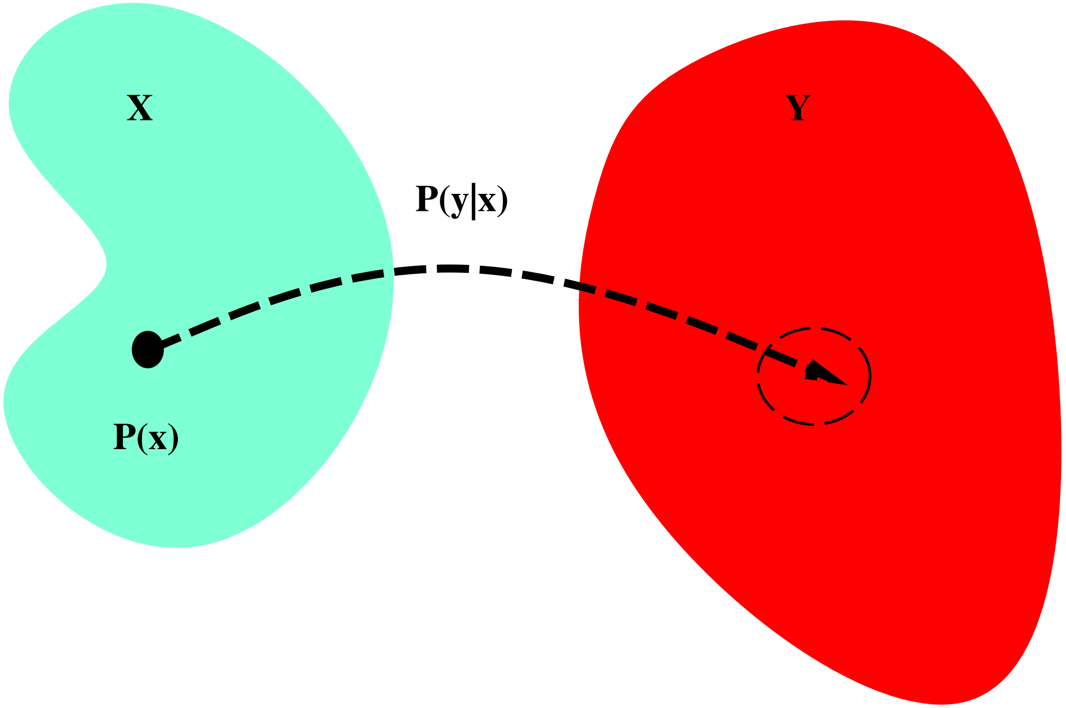
$$(x_1, y_1), \ldots, (x_\ell, y_\ell)$$

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

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

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

Probabilistic setting



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 $V(f(x), y^*)$ 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:

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

For tractability and other reasons, we often use the hinge loss (implicitely introduced by Vapnik) in binary classification:

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

Generalization error and empirical error

Given a function f, a loss function V, and a probability distribution P over X and Y, we can define the **expected error** of f as:

$$I[f] = \int V(f(x), y) d\mu(x, y)$$

which is also the **expected loss** on a new example drawn at random from a distribution (and where we use $d\mu$ to mean dP).

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

Given a function f, a loss function V, and a training set S consisting of ℓ datapoints, we can measure the **empirical error** (or risk) of f as:

$$I_S[f] = \frac{1}{\ell} \sum V(f(x_i), y_i)$$

Hypothesis Space

The **hypothesis space** $\mathcal{H}$ is the space of functions that we allow our algorithm to search. It is often chosen with respect to the amount of data available.

Empirical Risk Minimization

Given a training set S and a function space $\mathcal{H}$, empirical risk minimization (Vapnik) finds a function f_S that minimizes the empirical risk over all functions $f \in \mathcal{H}$:

$$\begin{aligned} f_S &= \arg\min_{f \in \mathcal{H}} I_S[f] \\ &= \arg\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i), y_i) \end{aligned}$$

(For now, we are assuming the existence of such a function.)

Consistency and Well-posedness of Empirical Risk Minimization

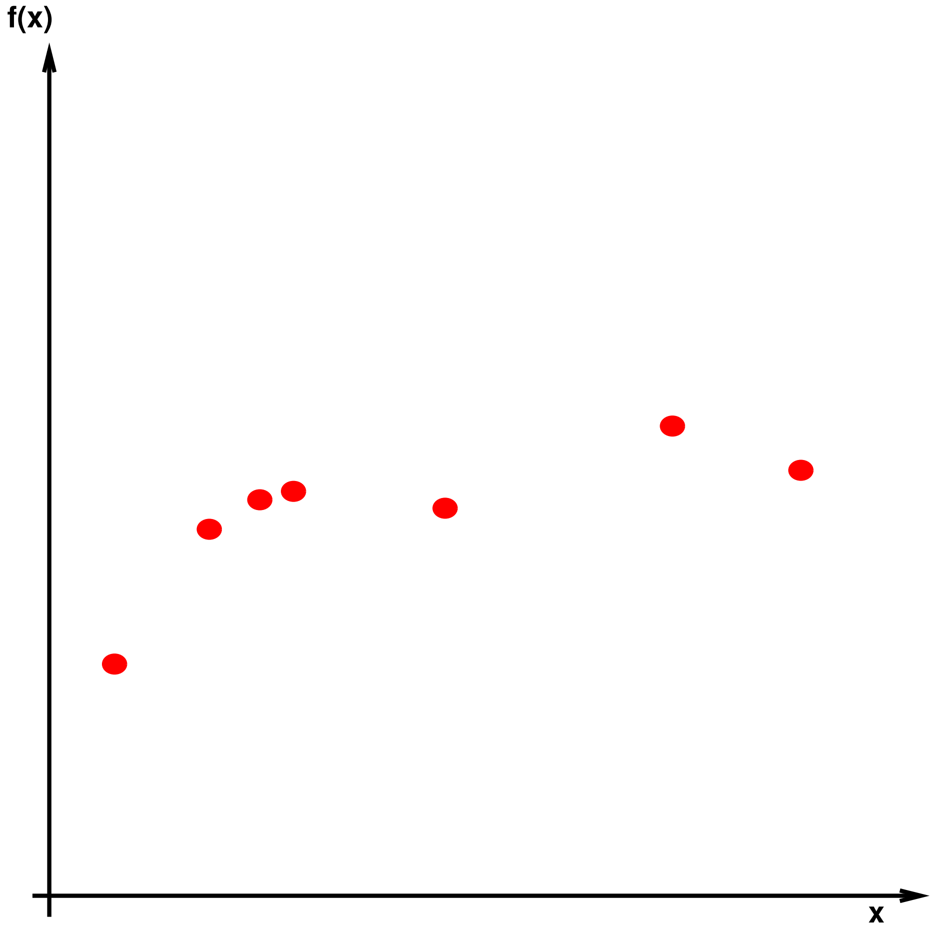
For the solution of ERM to be useful in the context of learning, the solution must be

- “consistent”
- it also must exist, be unique and be “stable” (well-posedness).

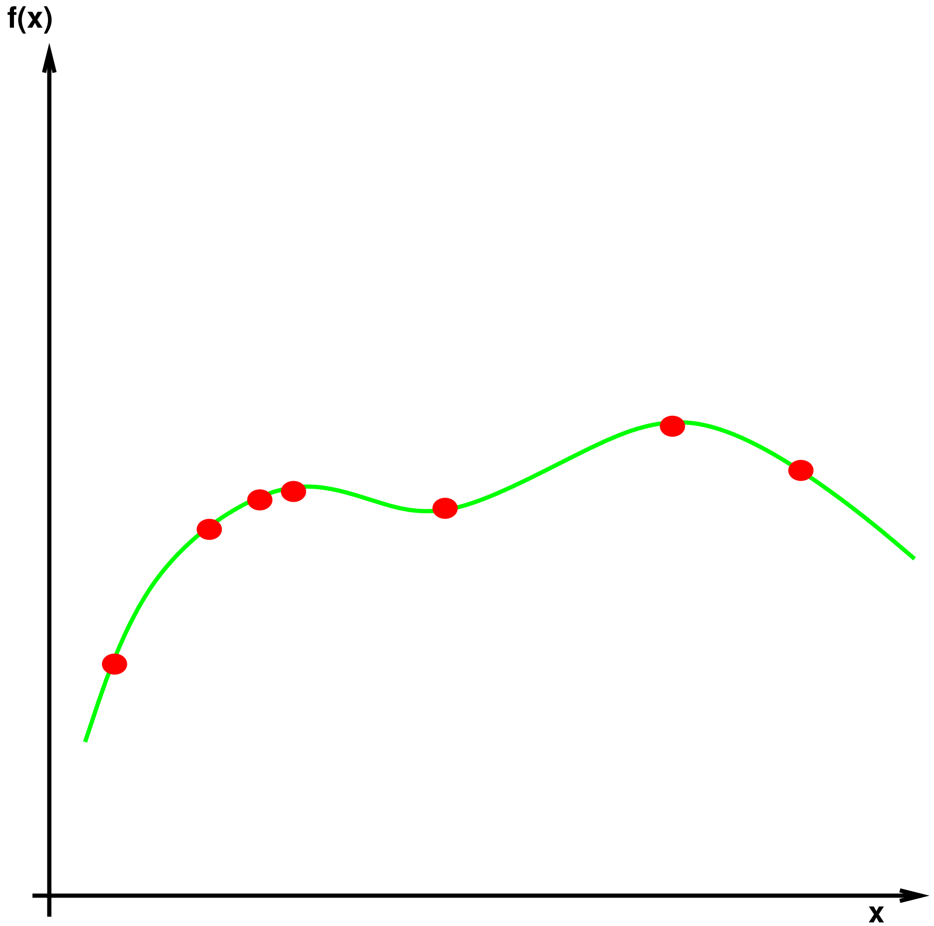
Consistency of ERM

Consistency means that the difference $I_S[f_S] - I[f_S]$ must go to zero as the number of training examples increases, that is $\ell \to \infty$. In other words, the training error for the ERM solution must converge to the expected error and thus be a "proxy" for it. Otherwise the solution would not be "predictive".

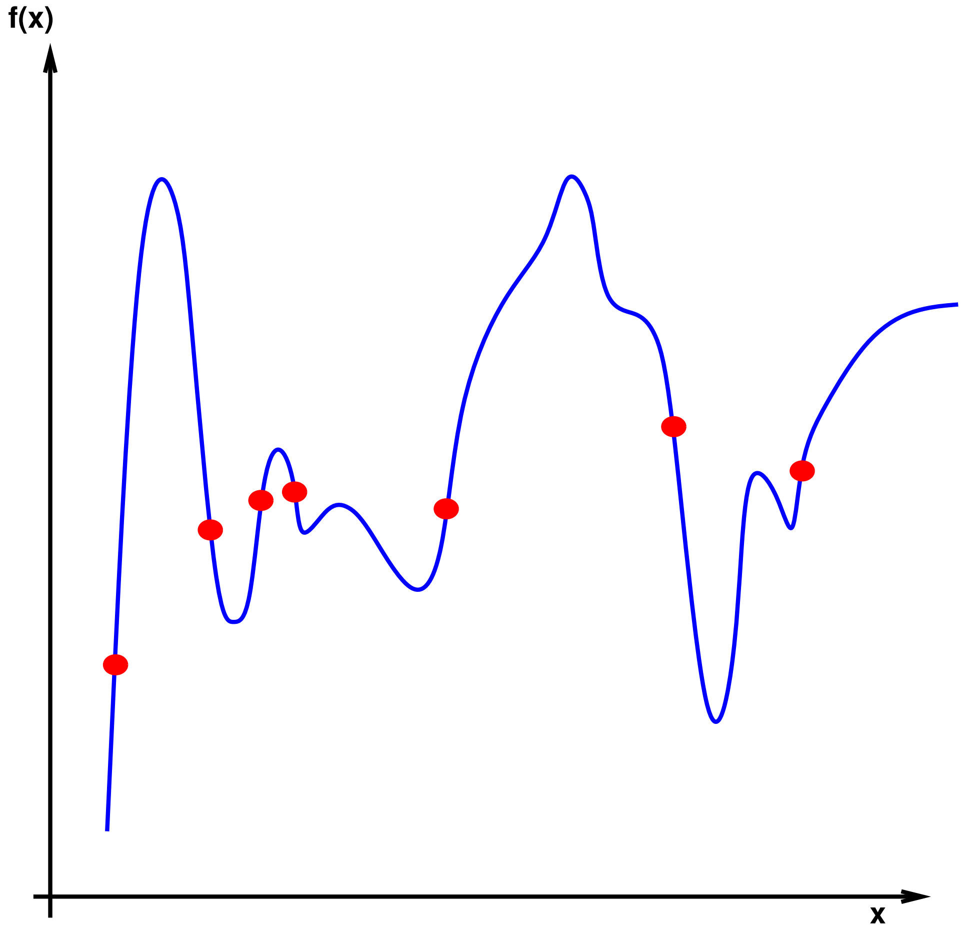
Here is a graphical example: given a certain number of samples...



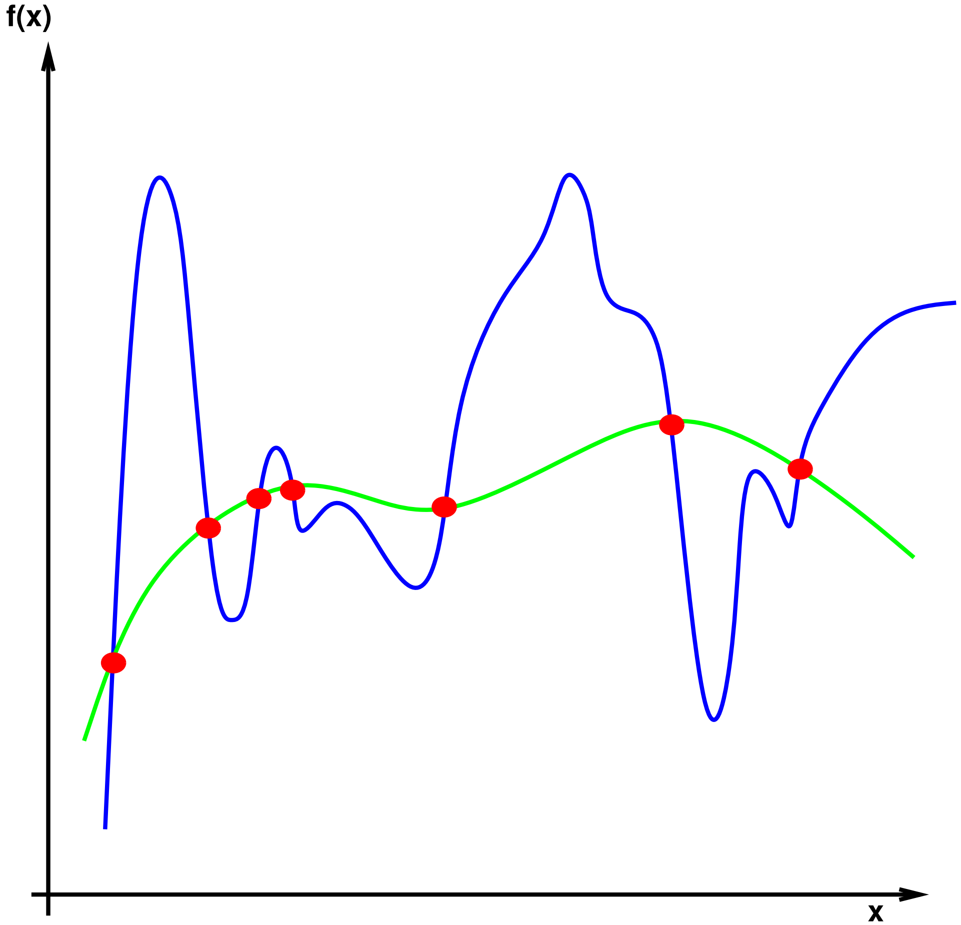
suppose this is the "true" solution...



... but ERM gives this solution!



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



Uniform Glivenko-Cantelli Classes

As we will see later a proper choice of the hypothesis space $\mathcal{H}$ ensures consistency of ERM. The key property is the uGC property. We say that a class of functions $\mathcal{F}$ is a uniform Glivenko-Cantelli (uGC) class *iff*, for all $\epsilon > 0$ and for all μ,

$$\mathbb{P}_\mu\left(\lim_{\ell\to\infty} \sup_{f\in\mathcal{F}} |I[f] - I_S[f]| > \epsilon\right) = 0$$

We will be exploring this definition (and equivalent definitions) in detail in 9.520. If $\mathcal{H}$ is a uGC class, this directly implies that $I_S[f] - I[f]$ for every function f in $\mathcal{H}$ gets small as $\ell \to \infty$.

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 ℓ goes to infinity.

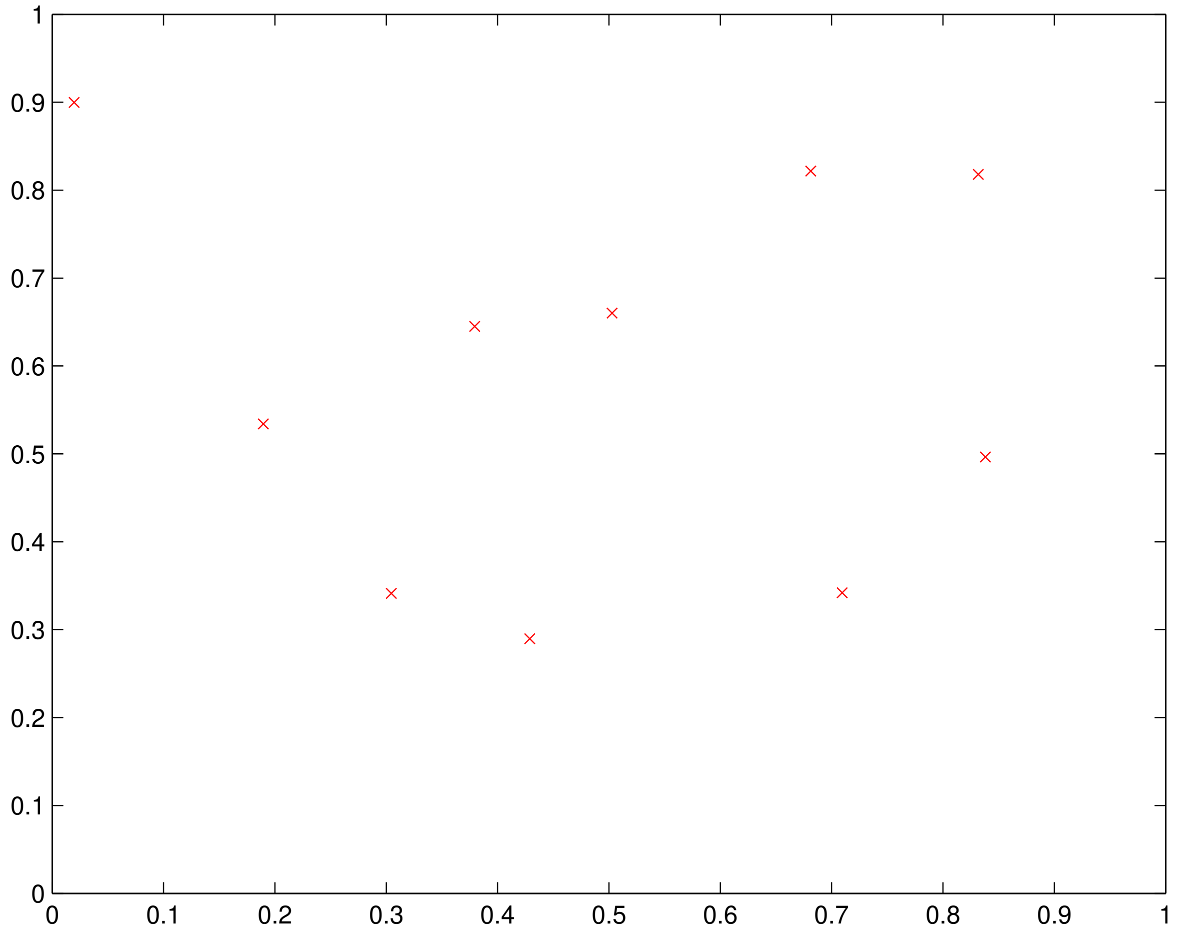
General definition of Well-Posed and Ill-Posed problems

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

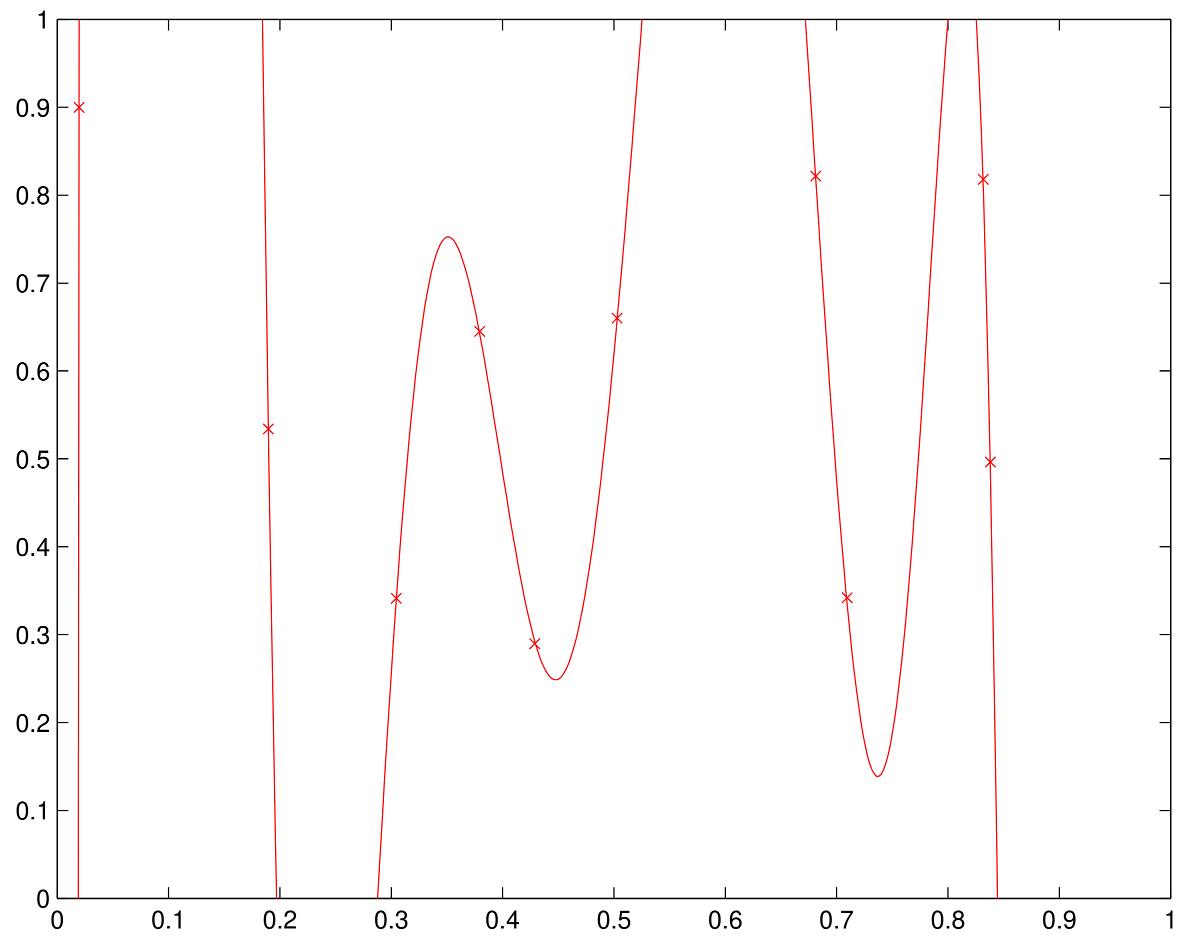
- exists
- is unique
- depends continuously on the data (e.g. it is *stable*)

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

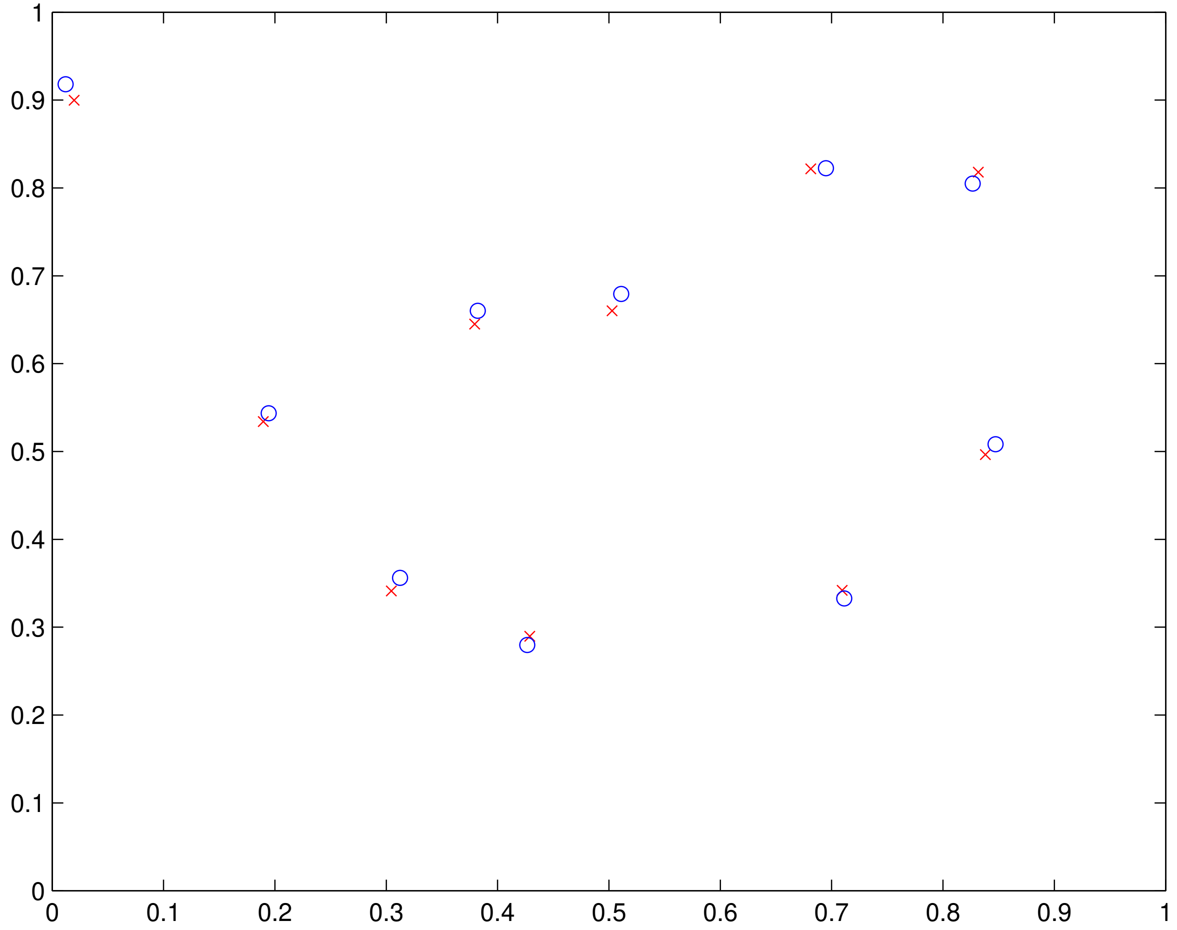
Here is a graphical example: given 10 samples...



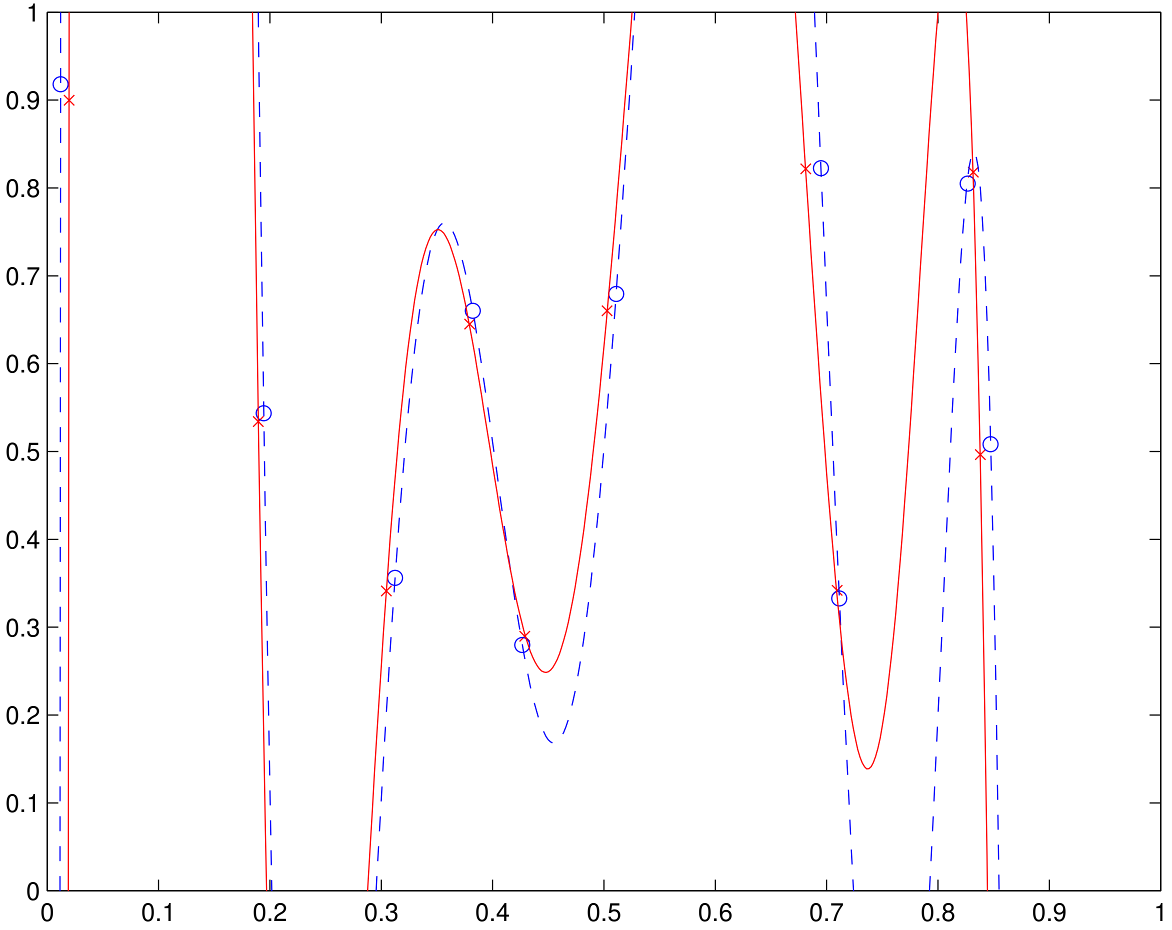
...we can find the smoothest interpolating polynomial.



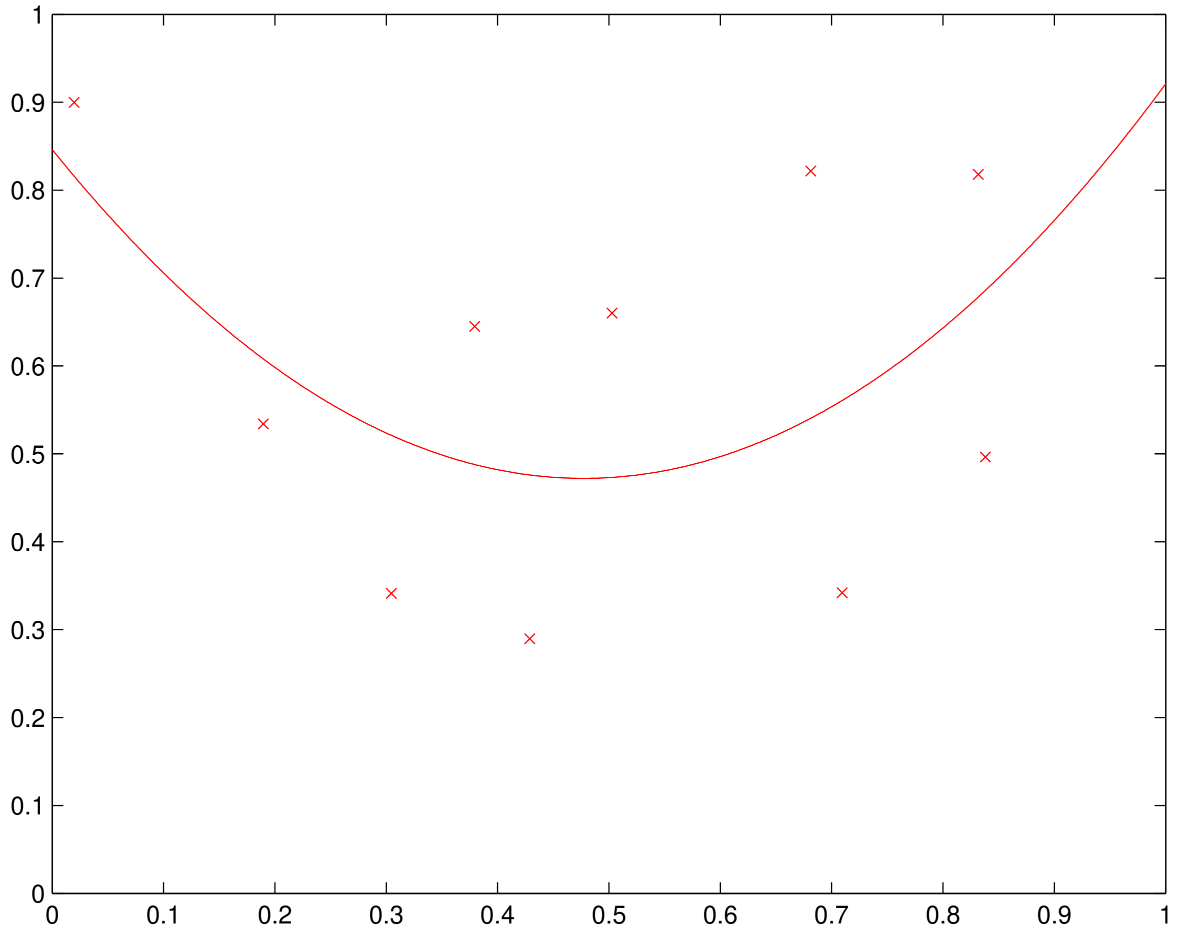
But if we perturb the points slightly...



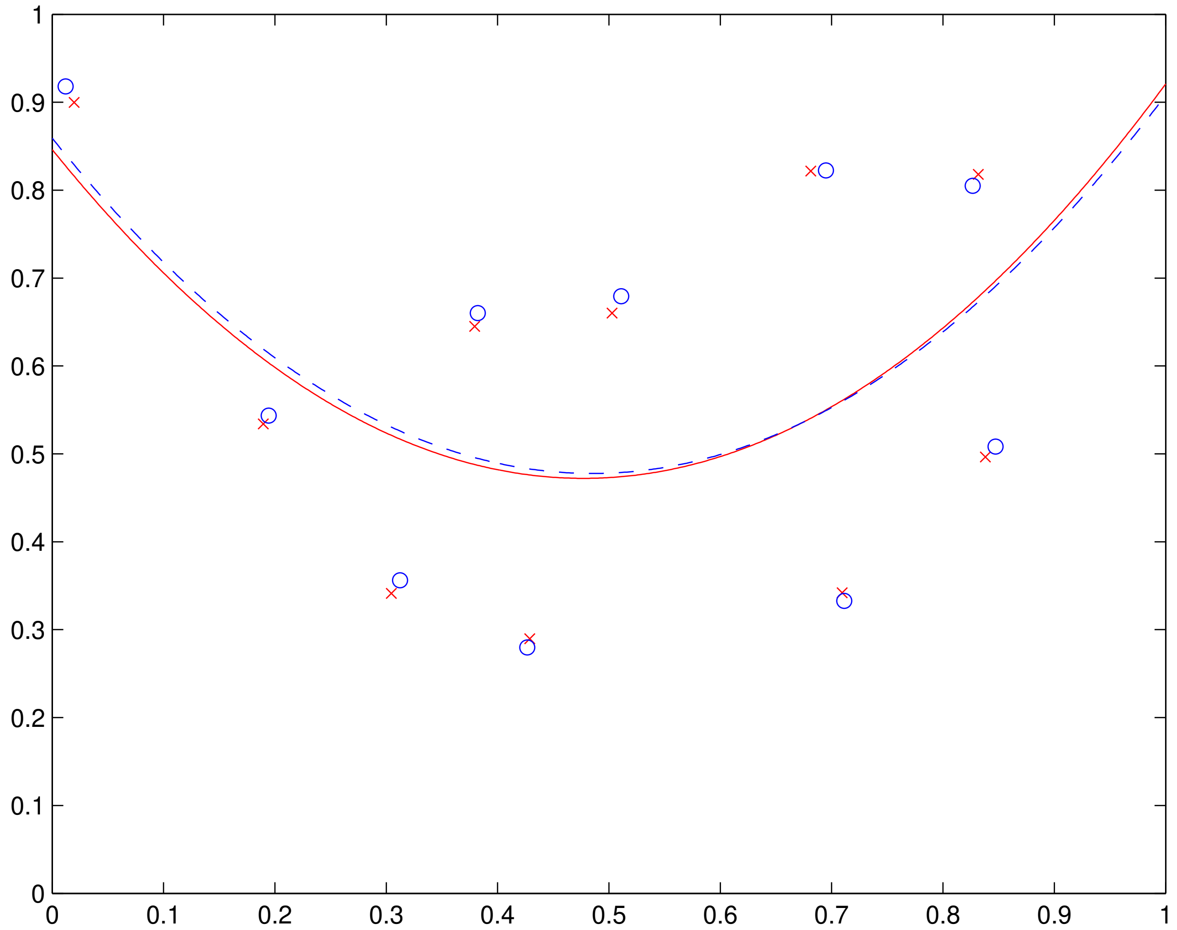
...the solution changes a lot.



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 is to restore well-posedness of ERM by constraining the hypothesis space $\mathcal{H}$. An indirect way to do so is to use Tikhonov regularization (which is not ERM).

Tikhonov Regularization

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

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

which in general – for arbitrary hypothesis space $\mathcal{H}$ – is *ill-posed*. Instead, we minimize over the hypothesis space $\mathcal{H}$, for a fixed positive parameter λ, the regularized functional

$$\frac{1}{\ell}\sum_{i=1}^{\ell} V(f(x_i), y_i) + \lambda \|f\|_K^2, \qquad (1)$$

where $\|f\|_K^2$ is the norm in $\mathcal{H}_K$ – the Reproducing Kernel Hilbert Space (RKHS), defined by the kernel K.

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 consistency

- 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 that minimizes the risk. Often, $\mathcal{T}$ is chosen to be all functions in L_2, or all differentiable functions.

Sample Error (also called Estimation Error)

Let $f_{\mathcal{H}}$ be the function in $\mathcal{H}$ with the smallest true risk.

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.

A main topic of this course is "bounding" the sample error; 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 ℓ 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.

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.

Generalization Error

We define the **generalization 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.

The generalization 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 generalization error will be small. There is a tradeoff between the approximation error and the sample error...

The Approximation/Sample Tradeoff

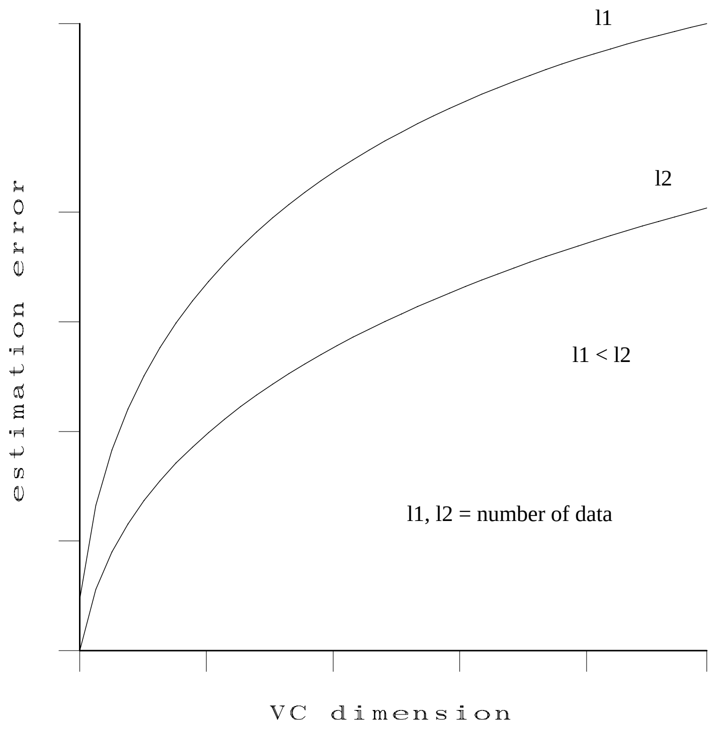
It should already be intuitively clear that making $\mathcal{H}$ big makes the approximation error small. This implies that we can (help) make the generalization error small by making $\mathcal{H}$ big.

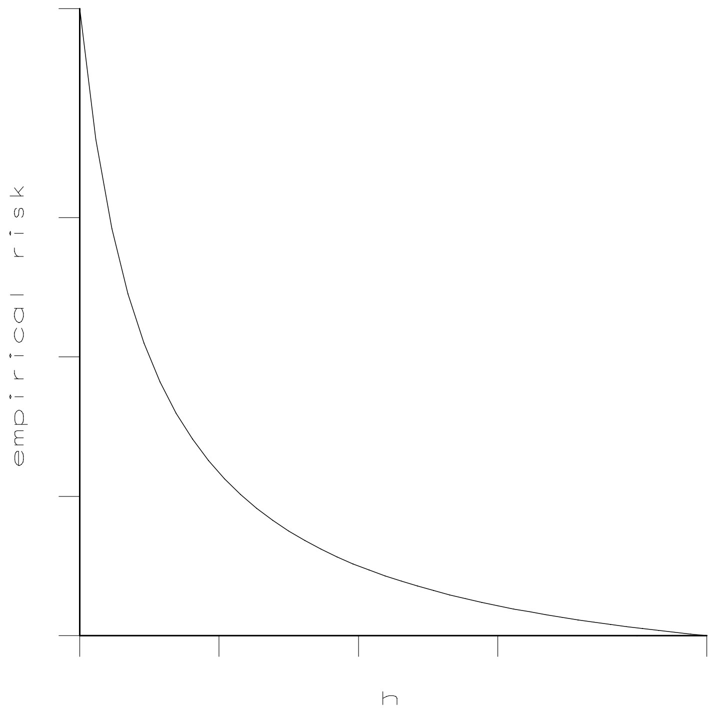
On the other hand, we will show that making $\mathcal{H}$ small will make the sample error small. In particular, if $\mathcal{H}$ is a uGC class, the sample error will go to zero as $\ell \to \infty$, but how quickly it goes to zero depends directly on the "size" of $\mathcal{H}$. This implies that we want to keep $\mathcal{H}$ as small as possible. (Furthermore, $\mathcal{T}$ itself may or may not be a uGC class.)

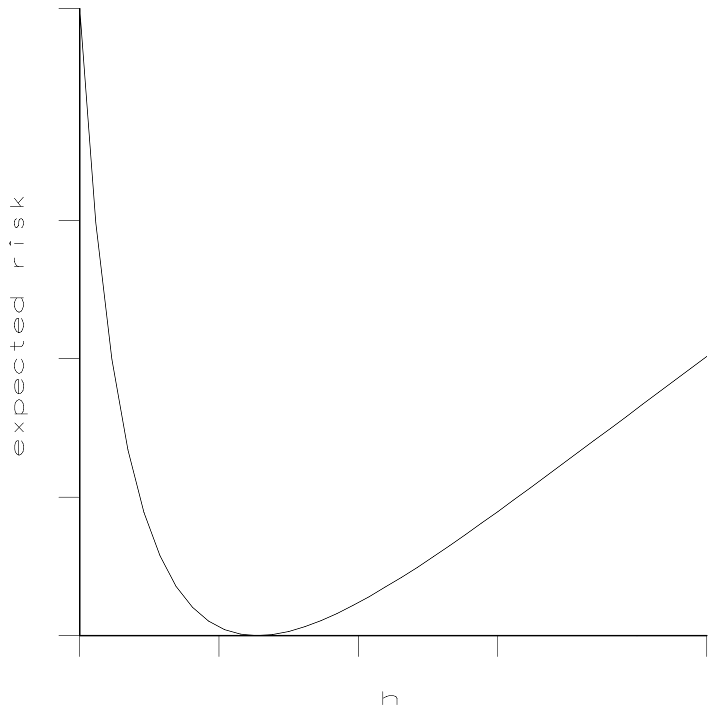
Ideally, we would like to find the optimal tradeoff between these conflicting requirements.

Generalization Error Definition Caveat

We define the generalization 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 generalization error. In the case where $I[f_0] = 0$, the two approaches are equivalent.


l1
l2
l1 < l2
l1, l2 = number of data
estimation error
VC dimension


empirical risk
h


expected risk
h