9.520: Statistical Learning Theory and Applications February 8th, 2010

The Learning Problem and Regularization

Lecturer: Tomaso Poggio *Scribe: Sara Abbasabadi*

1 Introduction

Today we will introduce a few basic mathematical concepts underlying conditions of the learning theory and generalization property of learning algorithms. We will define empirical risk, expected risk and empirical risk minimization. Then regularization algorithms will be introduced which forms the bases for the next few classes.

2 The Learning Problem

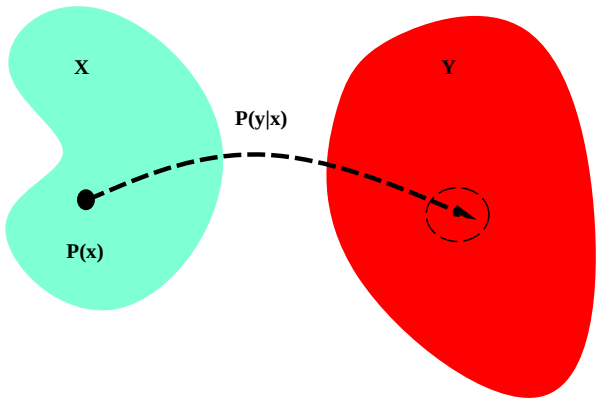
Consider the problem of supervised learning where for a number of input data X, the output labels Y are also known. Data samples are taken independently from an underlying distribution $\mu(z)$ on $Z = X \times Y$ and form the training set S:

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

that is $z_1, \ldots, z_n$. We assume independent and identically distributed (i.i.d.) random variables which means the order does not matter (this is the exchangeability property). Our goal is to find the conditional probability of y given a data x.

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

$p(x, y)$ is fixed but unknown. Finding this probabilistic relation between X and Y solves the learning problem.



For example suppose we have the data x_i, F_i measured from a spring. Hooke's law $F = ax$ gives $p(F/x) = \delta(F - ax)$. For additive Gaussian noise Hooke's law $F = ax$ gives $p(F/x) = e^{-(F-ax)^2/2\sigma^2}$.

2.1 Hypothesis Space

Hypothesis space $\mathcal{H}$ should be defined in every learning algorithm as the space of functions that can be explored. This space can be infinite or finite. The learning algorithm will look into this space of possible functions and depending on the training data S, finds the function that maps the input into the output. This function will be used for future predictions for a new value of X:

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

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. Some exapmles of the possible hypothesis spaces include:

- Example 1: H consists of continuous functions on $[0, 1]$. Continuity: For each ϵ there exist δ s.t. for all $x' : |x - x'| < \delta$ the following holds $|f(x) - f(x')| < \epsilon$
- Example 2: H consists of polynomials of degree n
- Example 3: H consists of all linear functions
- Example 4: H consists of equicontinuous functions on $[0, 1]$. H is bounded uniformly if there exists M s.t. $\forall f, x \ |f(x)| \leq M$. H is equicontinuous if: For each ϵ there exist δ s.t. for all $x', x"$ s.t. $|x' - x''| < \delta$ and for all $f \in H$ the following holds $|f(x') - f(x")| < \epsilon$. D'Arzela' theorem states that a necessary and sufficient condition for a class of continuous functions H to be compact is to be uniformly bounded and equicontinuous.

2.2 Loss Functions

Loss function V is defined in order to measure the goodness of our prediction. $V(f, z) = V(f(x), y)$ denotes the price we pay when we see sample x and guess that the associated y value is $f(x)$ when it is actually y. 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:

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

y is the correct label and $f(x)$ being the predicted label of sample x. 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 and y is binary, eg $y = +1$ or $y = -1$. This means that when y and f(x) have the same sign a correct prediction with zero loss is achieved. For tractability and other reasons, we often use the hinge loss (implicitely introduced by Vapnik) in binary classification:

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

2.3 Expected Error, Empirical Error

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. Expected loss indicates performance of the algorithm on the future samples. The goal is to choose f that makes the expected error $I[f]$ as small as possible but in general we do not know μ. Instead we compute the empirical approximation of the expected error called the empirical error which is the average error on the training set. Given a function f, a loss function V, and a training set S consisting of n data points, the **empirical error** of f is:

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

3 Generalization and Stability

Let's first defince convergence in probability. For a sequence of random variables, $\{X_n\}$, convergence is defined as follows:

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

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

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

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

3.1 Generalization

A natural requirement for f_S is distribution independent **generalization**. Generalization means that the difference between the empirical error and the expected error goes to zero as the number of training set samples increase:

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

This is equivalent to saying that for each n there exists a ε_n and a δ_n such that $\forall \mu$

$$\mathbb{P}\{|I_{S_n}[f_{S_n}] - I[f_{S_n}]| \geq \varepsilon_n\} \leq \delta_n,$$

with ε_n and δ_n going to zero for $n \to \infty$. 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". This is similar to the law of large numbers but there are diffrences between these two. A desirable additional requirement is **universal consistency** defined as:

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

The consistency criteria indicates that if the size of training set goes to infinity and over all probability distributions, the expected error is very close to the f with minimum error. We are interested to

have error bounds for any number of samples. For that we define convergence rate. Suppose we can prove that with probability at least $1 - e^{-\tau 2}$ we have

$$|I_S[f_S] - I[f_S]| \leq \frac{C}{\sqrt{n}}\tau$$

for some (problem dependent) constant C. The above result gives a convergence rate. If we fix ϵ, τ and solve for n the eq. $\epsilon = \frac{C}{\sqrt{n}}\tau$ we obtain the sample complexity:

$$n(\epsilon, \tau) = \frac{C^2\tau^2}{\epsilon^2}$$

This setup allows to find the number of samples needed for obtaining an error of ϵ, with confidence $1 - e^{-\tau 2}$. In addition to the key property of generalization, a "good" learning algorithm should also be *stable*. This means that we don't want the function found by the learning algorithm to depend critically on small changes in the training points. The stability problem is described in the next section.

3.2 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. 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. The problem of learning is the inverse problem. 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:

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

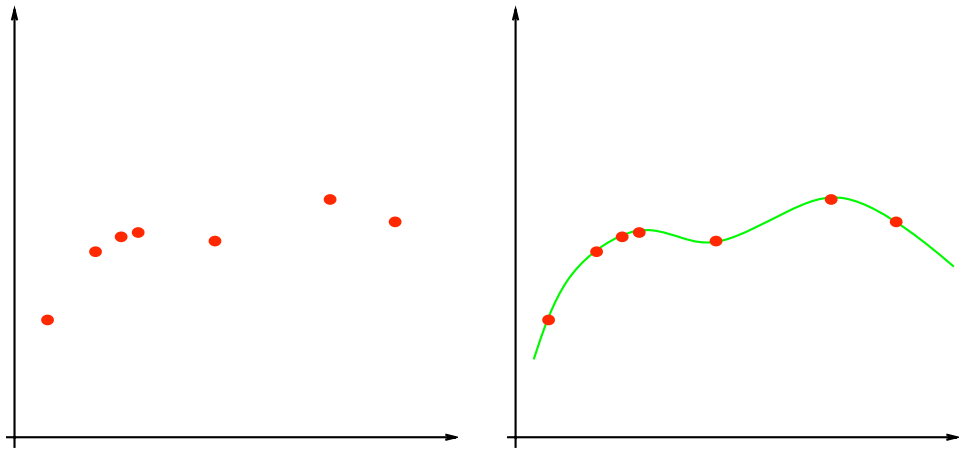
The inverse problem of finding u is well-posed when the solution exists, is unique and is *stable*, that is depends continuously on the initial data g. Ill-posed problems fail to satisfy one or more of these criteria. The learning problem like many inverse problems is often not stable and thus ill-posed. We would like our class of learning algorithms to be stable.

4 Empirical Risk Minimization

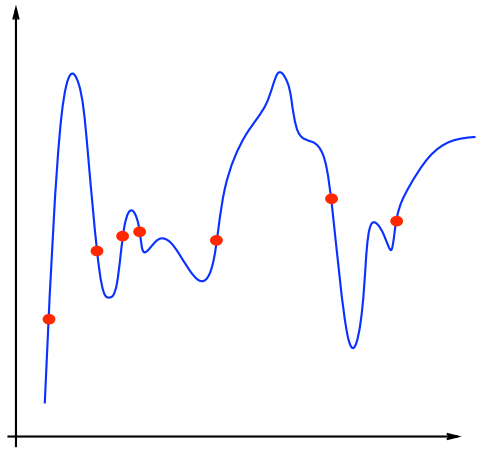
One classical and simple class of learning algorithms is the empirical risk minimization algorithm. Given a training set S and a function space $\mathcal{H}$, empirical risk minimization (Vapnik introduced the term) is the class of algorithms that look at S and select f_S in the hypothesis space that minimizes the empirical error:

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

For example linear regression is ERM when $V(z) = (f(x) - y)^2$ and H is the space of linear functions $f = ax$. For ERM to represent a "good" class of learning algorithms, the solution should be general and stable. Here is an example of generalization given a certain number of samples and the true solution for the samples:

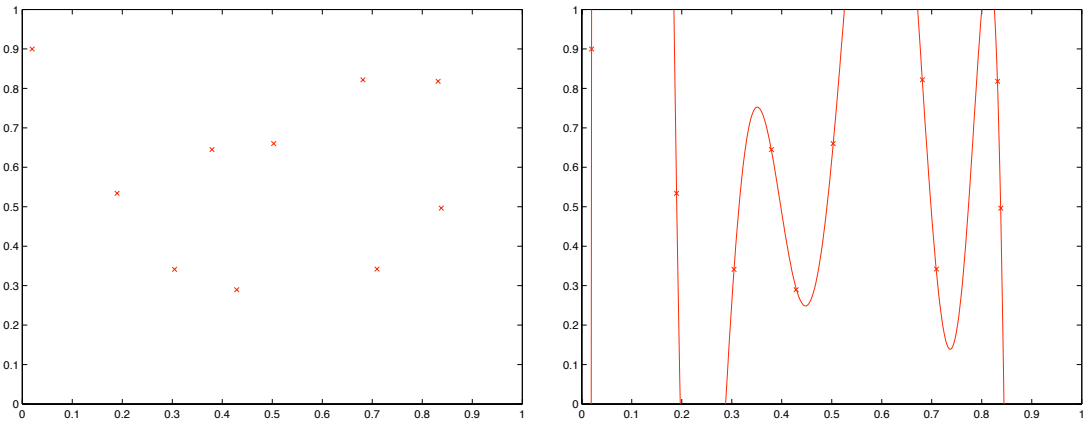


Suppose that ERM algorithm selects this function (with zero empirical error):

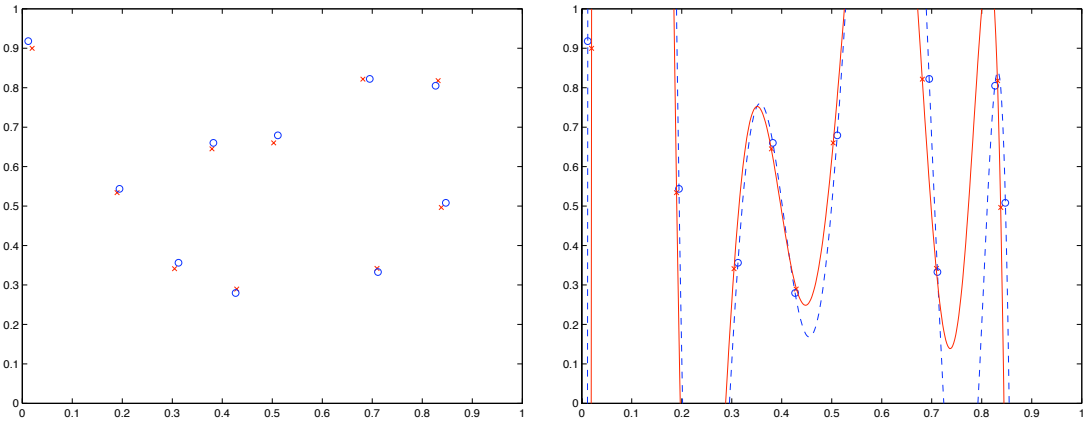


This is not the true function that we want to know. Thus, we need to set generalization conditions so that the ERM solution converges with increasing number of examples to the true solution.

Now let's consider a stability example using 10 training points and the smoothest interpolating polynomial fit with degree of 9 that gives a zero empirical error:

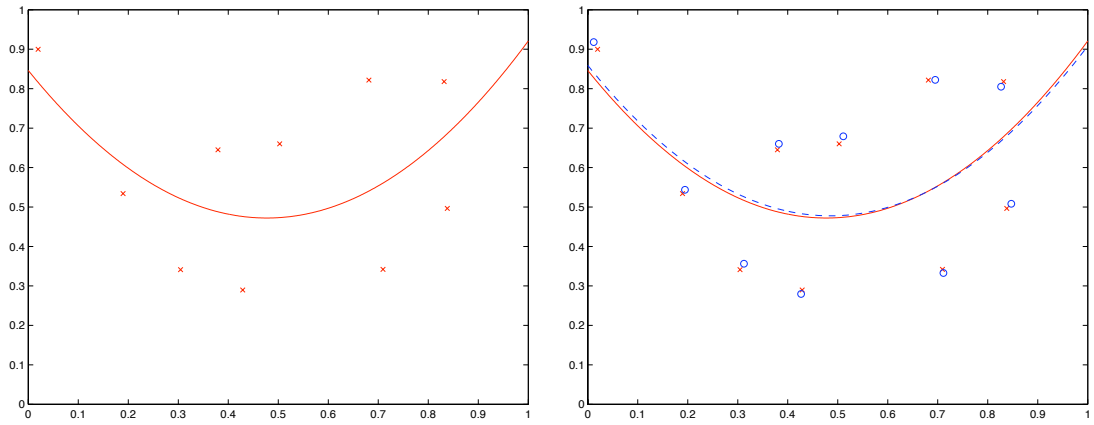


Perturbing the points slightly gives the following solution:



The results is a very different solution. This means lack of stability in this example.

Using a lower degree polynomial e.g. 2, gives the following fits before and after perturbation:



Although the fit of data is not as good as before (empirical error > 0) but the solution does not change much with perturbing the data, a fact that indicates stability.

4.1 Conditions for Well-posedness(stability) and Predictivity (generalization)

Since Tikhonov, it is well-known that a generally ill-posed problem such as ERM, can be guaranteed to be well-posed and therefore *stable* by an appropriate choice of $\mathcal{H}$. For example, compactness of $\mathcal{H}$ guarantees stability. We would like to have a hypothesis space that yields generalization too. Loosely speaking this would be a $\mathcal{H}$ for which the solution of ERM, say f_S is such that $|I_S[f_S] - I[f_S]|$ converges to zero in probability for n increasing. Note that the above requirement is NOT the law of large numbers since f depends on the training set S and is not fixed. Is there any possible relationship between stability and generalization?
According to (Vapnik and Červonenkis (71), Alon et al (97), Dudley, Giné, and Zinn (91)), a necessary and sufficient condition for generalization (and consistency) of ERM is that $\mathcal{H}$ is is a class of functions called uniform Glivenko-Cantelli (uGC). $\mathcal{H}$ is a uGC class if

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

For example class of continious functions are not uGC and thus don't have the generalization condition. This theorem (Vapnik et al.) indicates that 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). A separate theorem (Niyogi, Poggio et al.) guarantees stability of ERM. Thus with the appropriate definition of stability, stability and generalization are equivalent for ERM and correspond to the same constraints on $\mathcal{H}$.

5 Regularization

Regularization (originally introduced by Tikhonov independently of the learning problem) ensures *well-posedness* and thus *generalization* of ERM by constraining the hypothesis space $\mathcal{H}$ for instance

not allowing polynomials with too high degree with respect to the number of data. The direct way is called Ivanov regularization. The indirect way is Tikhonov regularization (which is not strictly ERM). 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

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

with functions that are constrained in their norm 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. Tikhonov regularization ensures well-posedness eg existence, uniqueness and especially *stability* (in a very strong form). It also provides generalization.

6 Target space, sample and approximation error

In addition to the hypothesis space $\mathcal{H}$, the space we allow our algorithms to search, we need to define the **target space** $\mathcal{T}$. This space is in general larger than the hypothesis space and contains 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.

6.1 Sample 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.

6.2 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.

6.3 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. Thus, by decreasing the approximation error, the sample error increases and vice versa. The goal is to have a hypothesis space which is large enough to give a small approximation error and small enough to give a small sample error.