How to control the generalization ability of learning processes?

Target: An inductive principle for minimizing the risk functional using a small sample of training samples.

The sample size $d$ is considered to be small if the ratio $\frac{d}{h}$ (where $h = \text{VC dimension}$) is small, say $\frac{d}{h} < 20$

Recall our bounds for the generalization ability of learning machines with sets of totally bounded non-negative functions:

$$R(f_{e}) \leq \text{Remp}(f_{e}) + \frac{B \epsilon}{2} \left(1 + \sqrt{1 + \frac{4 \text{Remp}(f_{e})}{B \epsilon}}\right)$$

(1)
bounds for the generalization ability of learning machines with sets of unbounded functions

\[ R(\varepsilon) \leq \frac{R_{\text{emp}}(\varepsilon)}{(1 - a(p) \varepsilon)^+} \]

(2)

where \( a(p) = \sqrt{\frac{1}{2} \left( \frac{p-1}{p-2} \right)^{p-1}} \)

Here \( \varepsilon = 2 \frac{\ln N - \ln \eta}{\varepsilon} \) if the set of functions \( \Omega(z, x_i) \) \( i = 1 \ldots N \) contains \( N \) elements and \( \varepsilon = 4 \frac{h \left( \ln \frac{2h}{n} + 1 \right) - \ln \frac{n}{4}}{\varepsilon} \) if the set of functions contains an infinite number of elements with finite VC dimension \( h \)

Each bound is valid with probability \( \geq 1 - \eta \).

We've been working under ERM principle: if \( \frac{h}{n} \to \infty \) then \( \varepsilon \) is small \( \Rightarrow \) consistency.
But in algorithmic practice we may face \( \frac{d}{n} \) small
Structural Risk Minimization Principle (SRM principle)
(Vapnik–Chervonenkis 1974) "admissible structure"
Let the set \( S \) of functions \( \Omega(z, x) \), \( x \in \Lambda \) be
provided with a structure consisting of nested
subsets of functions \( S_k = \{ \Omega(z, x), x \in \Lambda_k \} \) such
that \( S_1 \subset S_2 \subset \ldots \subset S_n \subset \ldots \)
where the elements of the structure satisfy the
following two properties:
(i). The VC dimension \( h_k \) of each set \( S_k \) of functions
is finite, and \( h_1 \leq h_2 \leq \ldots \leq h_n \leq \ldots \)
(ii). Any element \( S_k \) of the structure contains
either a set of totally bounded functions
\( 0 \leq \Omega(z, x) \leq B_k \), \( x \in \Lambda_k \)
or a set of functions satisfying the inequality
\[
\sup_{x \in \Lambda_k} \left( \int \Omega^p(z, x) dF(z) \right)^{\frac{1}{p}} \leq T_k, \quad p > 2
\]
for some pair \( (p, T_k) \)
Given a set of observations \( z_1, \ldots, z_N \), the SRM principle chooses \( \hat{z}(z, x^k) \) that minimizes the empirical risk in the subset \( S_k \) for which the guaranteed risk is minimal.

"trade-off between the quality of the approximator of the given data and the complexity of the approximating function".

As subset index \( n \uparrow \) the minima of \( \text{Remp}(x_n) \)

But the confidence interval (second summand in (1)) or multiplier in (2) increase \( \uparrow \)
Analysis of convergence based on SRM principle

\[ S^* = \bigcup_{k=1}^{\infty} S_k \]

\( S^* \) is everywhere dense in \( S = \{ Q(z, \alpha), \alpha \in \Lambda \} \) w.r.t. metric

\[ p(Q(z, \alpha_1), Q(z, \alpha_2)) = \int |Q(z, \alpha_1) - Q(z, \alpha_2)| dF(z) \]

Given any \( \ell \), we consider \( n = n(\ell) \) corresponding (determine) to \( S_n \) in which we minimize the empirical risk.

What is the asymptotic law of \( n(\ell) \)?

**Theorem 1.** The SRM method provides approximations \( Q(z, x_{n(\ell)}) \) for which the sequence of risks \( R(x_{n(\ell)}) \xrightarrow{\ell \to \infty} R(x_0) = \inf_{\alpha \in \Lambda} \int Q(z, \alpha) dF(z) \) s.t. the rate of convergence is \( V(\ell) \)

\[ V^{-1}(\ell) \frac{\left| R(x_{n(\ell)}) - R(x_0) \right|}{P} \xrightarrow{\ell \to \infty} C \]

\[ V(\ell) = R_{n(\ell)} + T_{n(\ell)} \sqrt{\frac{h_{n(\ell)} \ln \ell}{\ell}} \]
if the law \( n = n(x) \) is such that
\[
\lim_{x \to \infty} \frac{T_n(x) h_n(x) \ln x}{x} = 0
\]
where

(i). \( T_n = B_n \) if one considers a structure with totally bounded functions \( A(2, x) \leq B_n \) in \( S_n \)

(ii). \( T_n = T_n \) if one considers a structure with elements satisfying (3)

and \( r_n(x) \) is the rate of function approximation

\[
T_n = \inf_{x \in \Lambda_n} \int A(2, x) dF(x) - \inf_{x \in \Lambda} \int A(2, x) dF(x)
\]

The above theorem reduces the problem of best rate of convergence to the rate \( T_n \) of approximation for chosen structure.

\( T_n \) known \( \Rightarrow \) find \( n = n(x) \) a-priori

Estimate \( T_n \) for different structures on sets of functions

Classical function approximation theory
Example of applying Theorem 1
\[
V(t) = r_n(t) + T_n(t) \sqrt{\frac{\ln(t/e)}{t}}
\]

Deterministic Component of the learning process usually only has an asymptotic bound

Stochastic behavior of the learning process usually determined by non-asymptotic bounds of the risk

**Theorem 2** Suppose the set of functions \( f(x) \) is such that for some \( d > 0 \) and \( \hat{f}(w) \) Fourier transform

\[
\int |w|^d |\hat{f}(w)| \, dw = C_d(f) < \infty \quad d > 0
\]

Then the rate of approximation of the desired function by the best function of the elements of the structure is bounded by \( O\left(\frac{1}{\sqrt{n}}\right) \) if one of the following holds

1. \( d = 0 \) Elements \( S_n \) of the structure contains the functions

\[
f(x, \alpha, w, v) = \sum_{i=1}^{n} \alpha_i \sin \left[ (x \cdot w_i) + v_i \right]
\]

Jones, 1992
(2) \( d=1 \). Elements \( S_n \) of the structure contains the functions

\[
f(x, \omega, w, v) = \sum_{i=1}^{n} \alpha_i \left( s((x, \omega_i) + v_i) \right)
\]

where \( s(u) = \text{sigmoid function} \)

(Breiman, 1993) e.g \( s(u) = \tanh u \)

\[
s(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}
\]

(3) \( d=2 \). Elements \( S_n \) of the structure contains the functions

\[
f(x, \omega, w, v) = \sum_{i=1}^{n} \alpha_i \left| (x, \omega_i) + v_i \right|_+, \quad \left| w_i \right| = \text{max}\{w_i\}
\]

(Breiman, 1993)

In general, the rate of convergence of approximation of functions with \( S \) continuous derivatives is given by \( O(n^{-\frac{S}{N}}) \).

\( S_n \) — follow (2) in Theorem 2

\( N \) — dimensionality of input space
That is why "regularization" is called "smoothing".

Examples of structures of Neural Nets

1. Structure given by the architecture of the neural network

2. Structure given by learning procedure
   Assign "weights" to a given structured NN

\[ S_p = \{ f(x, w) : \| w \| < C_p \} , \quad C_1 < C_2 < \ldots < C_n \]

minimize \[ \mathcal{E}(w, V_p) = \frac{1}{e} \sum_{i=1}^{e} L(y_i, f(x_i, w)) + \gamma \| w \| \]

"weight - decay" procedure
3. Structure given by preprocessing
NN with fixed architecture

Input representation \( \mathbf{z} \rightarrow z = K(x, \beta) \)

\( \beta \) controls the degree of degeneracy introduced by this transformation (e.g., width of a smoothing kernel)

Structure \( S = \{ f(K(x, \beta), w), w \in W \} \).

\( \beta \geq \beta_p \quad c_1 > c_2 > \ldots > c_p \)

Implement SRM principle:

1. VC dim of \( S_k \)
2. Minimizer of empirical risk within \( S_k \)

Instead of global approximation one can also consider local approximation: Kernel method

"Local risk functional" \( R(d, \beta; x_0) = \int L(y, f(x_0)) \frac{K(k, x_0, \beta)}{k(x_0, \beta)} \) \( dF(x, y) \)

\( R(d, \beta; x_0) \leq \frac{1}{d} \sum_{i=1}^{d} L(y_i, f(x_i, x_0)) K(x_i, x_0, \beta) + (\beta - A) \mathbb{E}(l, h) \)
\[
E(l, h) = \sqrt{\frac{2h}{n} \ln(\frac{2e}{\delta})} - \ln\left(\frac{n}{e}\right)
\]

\(h_{\Sigma}\) : VC dimension of \(L(y, f(x, \alpha))K(x, x_0, \beta)\)

\(h_\beta\) : VC dimension of \(K(x, x_0, \beta)\)

(Vapnik - Botton 1993)

---

Minimum Description Length (MDL) Principle

Algorithm Complexity (Kolmogorov 1965) of an object

= Length of the shortest binary computer program that describes this object.

Kolmogorov proved that the value of the algorithmic complexity, up to an additive constant, does not depend on the type of computer.

What is a random object?

Randomness = High algorithm complexity

i.e. the string which describes the object cannot be compressed significantly
MDL Principle

Training set \((w_1, x_1), \ldots, (w_N, x_N)\)

Code book \(C_k\): \(N < 2^d\) different tables

\[ T_i = i = 1, 2, \ldots, N \]

Table \(T_i\) : \(\star \rightarrow w\) function

From table \(T_i\) we get

\[ x_1 \ldots x_N \xrightarrow{T_i} w_1^* \ldots w_N^* \]

Decoding: Perfect table \(T_0\) that minimize Hamming distance \(w_1^* \ldots w_N^*\) and \(w_1 \ldots w_N\)

Shannon: Minimal number of bits that describe any one of the \(N\) tables \(= \lceil \log_2 N \rceil\)

Coefficient of Compression

\[ K(T_0) = \frac{\lceil \log_2 N \rceil + \lceil \log_2 C_k \rceil + \lceil \log_2 d \rceil + \Delta d}{l} \]

\(d\) — minimal errors in Hamming distance.

\(K(T)\) small \(\longrightarrow\) String is not random and it depends on input \(x\)
The compression coefficient \( K(T) \) determines the probability of test error in classification (decoding) vectors \( x \) by the table \( T \).

**Theorem** If, on a given structure of code-books one compresses by a factor \( K(T) \) the description of string \( (w_i, x_i) \ldots (w_e, x_e) \) using a table \( T \), then with probability \( 1 - \eta \) one can assert that the probability of committing an error by the table \( T \) is bounded by

\[
R(T) < 2(K(T) \ln 2 - \frac{\ln \eta}{e})
\]

**Advantage** Only need to know \( K(T) \)

Do not need (i'). # of examples

(ii). Structure of Code book

(iii). Which code book

(iv). # of Table in codebook

(v). Training Error made using this table

**Disadvantage** "Smart" code-book.