Using Statistical Learning Theory to Rationalize System Model Identification and Validation Part I: Mathematical Foundations

A. A. Guergachi[*]
School of Information Technology Management,
Ryerson University,
Toronto, Ontario, Canada, M5B 2K3

G. G. Patry[†]
Department of Civil Engineering,
University of Ottawa,
Ottawa, Ontario, Canada, K1N 6N5

Existing procedures for model validation have been deemed inadequate for many engineering systems. The reason of this inadequacy is due to the high degree of complexity of the physical mechanisms that govern these systems. It is proposed in this paper to shift the attention from modeling the engineering system itself to modeling the uncertainty that underlies its behavior. A mathematical framework for modeling the uncertainty in complex engineering systems is developed. This framework uses the results of computational learning theory. It is based on the premise that a system model is a learning machine.

1. Introduction

Modeling of engineering systems is traditionally carried out in three sequential steps.

1. *Model development*. The modeler collects available knowledge about the studied system S in the form of first principles, empirical laws, and/or heuristic hypotheses. Based on this knowledge, the modeler develops a set of mathematical relationships (i.e., the system model $\mathcal{M}$) among the system state variables, which can generally be written in the form of a differential equation:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \tag{1}$$

where t is the time, $\mathbf{x}$ is the system state vector, $\mathbf{p}$ is the model parameter vector, and $\mathbf{f}$ is a mathematical function which is generally nonlinear.

[*]Electronic mail address: a2guerga@ryerson.ca.
[†]Electronic mail address: patry@uottawa.ca.

Complex Systems, **14** (2003) 63–90;

https://doi.org/10.25088/ComplexSystems.14.1.63

2. *Model identification*. After the model is developed, the modeler uses a set Υ_N (N being a natural number greater than 0) of empirical data:

$$\Upsilon_N : \mathbf{x}^{\text{data}}(t_1), \mathbf{x}^{\text{data}}(t_2), \ldots, \mathbf{x}^{\text{data}}(t_N) \tag{2}$$

collected from the real operation of the system to identify the model parameters. This step usually requires the minimization of an objective function $J(\mathbf{p})$ of the form:

$$J(\mathbf{p}) = \sum_{k=1}^{N} \left\| \mathbf{x}(\mathbf{p}, t_k) - \mathbf{x}^{\text{data}}(t_k) \right\|^2 \tag{3}$$

where $\mathbf{x}(\mathbf{p}, t)$ represents the solution to the model equation (1). In most cases, the data set Υ_N would actually be divided into two subsets Υ_{N_1} and Υ_{N_2} ($N = N_1 + N_2$). The first subset (called *identification sample*) is used for the model parameter vector identification and the second (called *validation sample*) for model validation (step 3).

3. *Model validation*. In this step, the identified system model is tested on the validation subset Υ_{N_2} that it has never "seen." If the model performs well on this sample, then it is retained. Otherwise, the model structure is adjusted and the validation procedure repeated.

The foregoing model validation procedure (called *cross validation*) has been criticized in many areas of engineering. In wastewater engineering, for example, in [1] Jeppsson pointed out that, "in a strict sense, model validation is impossible" with the existing validation techniques. Similarly, Zheng and Bennett in [2] noted that, in groundwater engineering, "models, like any scientific hypothesis, cannot be validated in the absolute sense . . . They can only be invalidated." Konikow and Bredehoeft suggested in [3] that terms like "model verification" and "model validation" convey a false sense of truth and accuracy and thus should be abandoned in favor of more realistic assessment descriptors such as history-matching and benchmarking.

The engineering systems for which the cross validation procedure is deemed inadequate all share one similar feature: the mechanisms that govern each one of them are so complex that no single model can be considered to describe these mechanisms in their entirety. The predictions of a model, no matter how sophisticated it is, are not guaranteed to match the reality. In this paper, it is proposed to shift the attention from modeling the system itself to modeling the uncertainty that underlies its behavior. The aim is to answer questions such as: What makes uncertainty high or low? How can it be controlled and to what extent can it be reduced?

A mathematical framework for modeling the uncertainty in complex engineering systems is developed in this paper. This framework is based

on the premise that a system model is a learning machine. The model identification procedure is viewed as a learning problem or, equivalently, an information transfer from a finite set of real data Y_N into the system model. Uncertainty is measured by the deviation $\mathcal{D}$ between the system's actual response function (see below for the definition of this function) and the approximation delivered by the system model $\mathcal{M}$ (after identification) for this function. This deviation is also a measure of the performance of the system model: the smaller the value of $\mathcal{D}$, the higher the performance of the model.

The framework leads to a set of inequalities (called *uncertainty models*, see equation (4)) that define bound functions φ on the deviation $\mathcal{D}$. These inequalities are of the general form $\mathcal{D} \leq \varphi$. The bound functions φ are dependent on (among others): (1) the amount N of data used to carry out the system model identification, and (2) the complexity of the system model structure q. The inequalities $\mathcal{D} \leq \varphi$ can be utilized to evaluate the quality of a system model after it has been identified and are useful for system modelers in many respects.

- When a set of values of N and q is fixed (e.g., $N = N_0$ and $q = q_0$), the inequalities allow the modeler to compute a bound φ_0 (φ_0 being the value of φ for $N = N_0$ and $q = q_0$) on the deviation $\mathcal{D}$ which, as indicated above, represents a measure of the model performance. The modeler then obtains a guarantee on the model quality.

- When the model structure complexity (i.e., q) is fixed (e.g., $q = q_0$), the inequalities allow the modeler to assess the rate of model performance improvement as the value of N increases. This assessment can be done by computing the partial derivative:

  $$\left(\frac{\partial \varphi}{\partial N}\right)_{q_0}$$

  of the bound φ.

- When the amount of data N is fixed (e.g., $N = N_0$), the inequalities allow the modeler to select the optimal model structure complexity q_{opt} that minimizes the bound function φ, N being set equal to N_0.

Consequently, the inequalities $\mathcal{D} \leq \varphi$ can potentially be used as replacements for the traditional system model validation procedures, since they provide the system modeler with a method of computing a guarantee on the model performance.

The development of the framework is based on the extensive research work by Vapnik in [4, 5, 6] and that of Vapnik and Chervonenkis in [7, 8, 9] in the area of mathematical statistics and its applications to computational machine learning theory. Section 2 shows why and how a system model can be considered as a learning machine. The remainder of the paper is devoted to the framework development.

https://doi.org/10.25088/ComplexSystems.14.1.63

2. A system model is a learning machine

Assume that we are interested in the variations of one state variable x_{i_0} of the system S and consider the model differential equation that governs the dynamics of this variable:

$$\dot{x}_{i_0} = f(t, \mathbf{x}, \mathbf{p})$$

or

$$\frac{dx_{i_0}}{dt} = f(t, \mathbf{x}, \mathbf{p}) \tag{4}$$

where t is the time, $\mathbf{x}$ is the process state vector, $\mathbf{p}$ is the parameter vector, and f is a real-valued function. This equation represents one component of the vector differential equation:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p})$$

of the system model $\mathcal{M}$. However, the vectors $\mathbf{x}$ and $\mathbf{p}$ in equation (4) do not necessarily contain all of their components. Normally, they should be denoted as $\mathbf{x}_{x_{i_0}}$ and $\mathbf{p}_{x_{i_0}}$ and equation (4) should become:

$$\frac{dx_{i_0}}{dt} = f(t, \mathbf{x}_{x_{i_0}}, \mathbf{p}_{x_{i_0}}) \tag{5}$$

in order to highlight the fact that $\mathbf{x}$ and $\mathbf{p}$ contain only those state variables and parameters, respectively, that influence the dynamics of x_{i_0}.

This study will be limited to the case of *autonomous systems*, that is, systems whose models do not depend explicitly on time. In other words, the general model equation that governs x_{i_0} can be written as:

$$\frac{dx_{i_0}}{dt} = f(\mathbf{x}_{x_{i_0}}, \mathbf{p}_{x_{i_0}}). \tag{6}$$

In addition to x_{i_0}, all state variables; that is, components of $\mathbf{x}_{x_{i_0}}$, are assumed to be directly and separately measurable.

Using the Euler method to numerically integrate equation (6), the time is discretized with a time step of Δt and then x_{i_0} is computed at times

$$t_1 = \Delta t\,, \quad t_2 = 2\,\Delta t\,, \quad \ldots\,, \quad t_n = n\,\Delta t\,, \quad \ldots$$

using the following equation:

$$x_{i_0}(t_n) = x_{i_0}(t_{n-1}) + \Delta t\, f(\mathbf{x}_{x_{i_0}}(t_{n-1}), \mathbf{p}_{x_{i_0}}). \tag{7}$$

Define $w_{t_n}^{\mathcal{M}}$ as the value of x_{i_0} to be predicted by the model $\mathcal{M}$, that is:

$$w_{t_n}^{\mathcal{M}} = x_{i_0}(t_n).$$

Similarly, define the vector $\mathbf{v}_{t_n}$ as:

$$\mathbf{v}_{t_n} = [x_{i_0}(t_{n-1}), \mathbf{x}_{x_{i_0}}(t_{n-1})^T]^T, \tag{8}$$

where the superscript T denotes a transposed vector. The number $w_{t_n}^{\mathcal{M}}$ takes values from a subset W of the real line $\Re$, and vector $\mathbf{v}_{t_n}$ from a multidimensional space V.

Now introduce the real-valued function H defined as:

$$H(\mathbf{v}_{t_n}, \mathbf{p}_{x_{i_0}}) = x_{i_0}(t_{n-1}) + \Delta t f(\mathbf{x}_{x_{i_0}}(t_{n-1}), \mathbf{p}_{x_{i_0}}). \tag{9}$$

The expression of this function corresponds to that of the right-hand side of equation (7). The latter equation then becomes:

$$w_{t_n}^{\mathcal{M}} = H(\mathbf{v}_{t_n}, \mathbf{p}_{x_{i_0}}). \tag{10}$$

For a fixed parameter vector $\mathbf{p}_{x_{i_0}}$, $H(\,.\,, \mathbf{p}_{x_{i_0}})$ represents a mapping function from V to W:

$$\begin{aligned} H(\,.\,, \mathbf{p}_{x_{i_0}}) : \quad V &\rightarrow W \\ \mathbf{v}_{t_n} &\mapsto w_{t_n}^{\mathcal{M}} = H(\mathbf{v}_{t_n}, \mathbf{p}_{x_{i_0}}). \end{aligned} \tag{11}$$

The parameter vector $\mathbf{p}_{x_{i_0}}$ takes values from a multidimensional space denoted here as Γ. Define the functional set $\mathcal{H}_{\mathcal{M}}$ of all mappings $H(\,.\,, \mathbf{p}_{x_{i_0}})$ with $\mathbf{p}_{x_{i_0}} \in \Gamma$:

$$\mathcal{H}_{\mathcal{M}} = \{H(\,.\,, \mathbf{p}_{x_{i_0}}) \mid \mathbf{p}_{x_{i_0}} \in \Gamma\}. \tag{12}$$

Now assume that a sequence of actual measurements of the couple $(\mathbf{v}_t, w_t)$:

$$\Upsilon_N : (\mathbf{v}_1, w_1), (\mathbf{v}_2, w_2), \ldots, (\mathbf{v}_N, w_N)$$

can be obtained from the real process operation, and consider an algorithm $\mathcal{A}$ that receives the sequence Υ_N as input and produces a parameter vector $(\mathbf{p}_{x_{i_0}})_{\text{emp}}$ corresponding to the function $H(\,.\,, (\mathbf{p}_{x_{i_0}})_{\text{emp}}) \in \mathcal{H}_{\mathcal{M}}$ that best approximates the real process response. In practice, this algorithm corresponds to the system model identification procedure which consists of minimizing an objective function of the form:

$$J(\mathbf{p}) = \sum_{k=1}^{N} \left| w_k - H(\mathbf{v}_k, \mathbf{p}) \right|^2 \tag{13}$$

or, equivalently:

$$R_{\text{emp}}(\mathbf{p}) = \frac{1}{N} \sum_{k=1}^{N} \left| w_k - H(\mathbf{v}_k, \mathbf{p}) \right|^2. \tag{14}$$

https://doi.org/10.25088/ComplexSystems.14.1.63

The subscript emp means "empirical" and the number $|w_k - H(\mathbf{v}_k, \mathbf{p})|^2$ represents a measure of the loss between the desired response w_k corresponding to the vector $\mathbf{v}_k$ and the model prediction represented by $H(\mathbf{v}_k, \mathbf{p})$.

A set of mapping functions equipped with an algorithm such as $\mathcal{A}$ is called a *learning machine* in the area of artificial intelligence and computational learning theory. We have shown that the couple $\mathcal{LM}_S = (\mathcal{H}_\mathcal{M}, \mathcal{A})$, composed of a system model and an identification procedure, can be viewed as a learning machine. On the basis of this result, it is possible to develop a mathematical framework that will allow us to model the uncertainty that underlies the behavior of the engineering system S, and rationalize the procedures of system model identification and validation. The next sections of this paper are about the development of this framework.

It should be noted that one of the objectives of the framework is to abstract the basic notions (system, model, model parameter, and objective function) of the traditional system modeling approach (introduced previously), in order to enhance the system model identification and validation procedures. As a result of this abstraction work, several new concepts are introduced in the following sections of the paper. For a full and detailed explanation of the implementation of these concepts in the case of modeling a concrete engineering system, the reader is referred to Part II of this paper [10]. As a transition to section 3, however, the following table presents some preliminary indications regarding the correspondence between the traditional approach's basic notions and the framework's new concepts.

Traditional approach basic notions	**Corresponding framework concepts**
The system S itself.	The transformer $\mathcal{T}$.
The system surrounding S.	The environment $\mathcal{E}$.
The space $\mathcal{H}_\mathcal{M}$ ($\mathcal{M}$ being the system model).	The decision rule space $\mathcal{H}$.
The model parameters $\mathbf{p}$.	The parameters that characterize the elements of $\mathcal{H}$.
The objective function $J(\mathbf{p})$.	The empirical risk.

Note also that several concepts will be introduced in the framework, with no corresponding notions in the traditional system modeling approach (e.g., the expected risk, the VC dimension).

Remark 1. Note that training of the machine

$$\mathcal{LM}_S = (\mathcal{H}_\mathcal{M}, \mathcal{A})$$

associated with the system S is carried out for a specific time t_n. This time is arbitrary, but fixed. The examples $(\mathbf{v}_1, w_1), (\mathbf{v}_2, w_2), \ldots, (\mathbf{v}_N, w_N)$ to

be used for machine training should therefore correspond to a realization of the system at time t_n, for a fixed n (i.e., a realization in the ensemble of the stochastic process $(\mathbf{v}_t, w_t)$, and not in time; see pages 372 and 442 of [11]). In practice, this is not possible, because the instance vector $\mathbf{v}_t$ and the outcome w_t are measured only once at any time instant t. And what is obtained from these measurements is actually a time series:

$$(\mathbf{v}_{t_1}, w_{t_1}), (\mathbf{v}_{t_2}, w_{t_2}), \ldots, (\mathbf{v}_{t_n}, w_{t_n}), \ldots$$

whose terms represent the couples instance/outcome at successive time instants $t_1, t_2, \ldots, t_n, \ldots$. It corresponds to one realization of the system S in time. This realization would usually—if not always—be the only one that is available for investigating the system's behavior. The property that allows us to use the series $(\mathbf{v}_{t_i}, w_{t_i})$ instead of $(\mathbf{v}_i, w_i)$ is called *ergodicity*. This condition is quite weak and will be assumed to hold true for the studied system S. A thorough discussion of this condition and how it is utilized to implement this framework in the case of a concrete engineering system is presented in Part II of this paper [10] as well as in [12].

3. General description of the framework

In a certain environment $\mathcal{E}$, at a time instant t, a situation $\mathbf{v}_t$ arises randomly and a transformer $\mathcal{T}$ acts and assigns to this situation $\mathbf{v}_t$ a number w_t obtained as a result of the realization of a random trial. Formally, situation $\mathbf{v}_t$ represents a random vector that takes values from an abstract space V called the *instance space*. It is generated according to a fixed but unknown probability density function (PDF) $P_{\mathbf{v}_t}$ defined on V. The number w_t, which is dependent on $\mathbf{v}_t$, represents a random variable that takes values from another space $W \subseteq \Re$ called the *outcome space*. It is generated according to a conditional PDF $P_{w_t|\mathbf{v}_t}$ defined on W, also fixed but unknown. The mathematical object $(\mathbf{v}_t, w_t)$ arises then in the product space $Z = V \times W$ (called the *sample space*) according to the joint PDF $P_{(\mathbf{v}_t, w_t)} = P_{\mathbf{v}_t} P_{w_t|\mathbf{v}_t}$, which characterizes the probabilistic environment $\mathcal{E}$. In what follows, the couple $(\mathbf{v}_t, w_t)$ is denoted as z_t (meaning that it takes values from the sample space Z). Using this notation, the joint PDF $P_{(\mathbf{v}_t, w_t)}$ is then denoted as P_{z_t}. The vector $\mathbf{v}_t$ will be indifferently called *situation* or *instance* and the number w_t *outcome* or *transformer's response*.

In the context of this paper, the parameter t represents the time; but, *a priori*, it could refer to some other continuous parameter such as distance or angle [11]. It takes values in the set of real numbers $\Re$. The family $\{z_t, t \in \Re\}$ of the random variables z_t is a stochastic process in the environment $\mathcal{E}$. The set $OM = \{\mathbf{E}(z_t), t \in \Re\}$ of all possible values of $\mathbf{E}(z_t)$ can, in theory, cover the entire sample space Z. In

https://doi.org/10.25088/ComplexSystems.14.1.63

practice, however, OM is a subset of Z that covers a specific region of Z. This subset OM will be designated here as the *operating mode* of the transformer $\mathcal{T}$.

To illustrate what is meant by "operating mode" consider, for instance, the behavior of an automotive engine. The operating conditions of such an engine are not the same when the car is climbing a hill as when it is taking a highway. In the first case, the engine develops a very high torque and the speed is low, while in the second case, the same engine operates under opposite conditions: the speed is high but the torque is low. Another example that illustrates this concept of operating mode is a wastewater treatment plant using the activated sludge process. The operation of this plant can use a little return sludge and low solids in the aeration tank in order to achieve the objective of removing soluble substrate with relatively low oxygen supply. But this plant could also be operated with the purpose of aerobically destroying all of the organic solids in the waste, which can be done by returning all the sludge to the aeration tank. Thus, the same plant could operate under different operating conditions.

Associated with the environment $\mathcal{E} = (\mathcal{T}, OM, z_t, P_{z_t})$ is a learning machine $\mathcal{LM}$ whose objective is to understand the behavior of the transformer $\mathcal{T}$. It receives a finite sequence $\boldsymbol{\Upsilon}_N$ of N training examples:

$$\boldsymbol{\Upsilon}_N : ((\mathbf{v}_t)_1, (w_t)_1), ((\mathbf{v}_t)_2, (w_t)_2), \ldots, ((\mathbf{v}_t)_N, (w_t)_N)$$

or, using z-notation:

$$\boldsymbol{\Upsilon}_N : (z_t)_1, (z_t)_2, \ldots, (z_t)_N$$

generated in the environment $\mathcal{E}$, as a result of N distinct random trials in the space Z.

Note. For the sake of simplifying the notations, we shall, in what follows, denote the variables:

$$((\mathbf{v}_t)_i, (w_t)_i) \quad \text{and} \quad (z_t)_i$$

as simply:

$$(\mathbf{v}_i, w_i) \quad \text{and} \quad z_i$$

respectively.

The elements z_i of $\boldsymbol{\Upsilon}_N$ are instances of the random vector z_t that are obtained by physically measuring the components of z_t at the end of each of the N random trials. Based on these training examples, the learning machine $\mathcal{LM}$ selects a strategy that specifies the best approximation $w^{\mathcal{LM}}$ of the transformer's response for each instance $\mathbf{v}_t$. Once this strategy is selected, it will be used on all future situations $\mathbf{v}_t$ arising in the environment $\mathcal{E}$, in order to predict the transformer's responses. This

strategy, which is mathematically a mapping function from V into W, is called a *decision rule* and is chosen from a fixed functional space $\mathcal{H}$ called the *decision rule space*.

The goal of $\mathcal{LM}$ is then to select, from the space $\mathcal{H}$, that particular decision rule which best approximates the transformer's response. The expression "best approximation of the transformer's response" means "closeness to the transformer's 'general tendency' $g^{\mathcal{T}}$." The latter function is defined as

$$g^{\mathcal{T}}(\mathbf{v}_t) = \mathbf{E}(w_t \mid \mathbf{v}_t) = \int_W w_t \, P_{w|\mathbf{v}_t}(w_t \mid \mathbf{v}_t) \, dw. \tag{15}$$

This function will be indifferently called *general tendency* or *response function*. Closeness is understood in the sense of the metric $\mathcal{D}$ defined in the following way:

$$\begin{aligned} \forall h \in \mathcal{H}, \qquad \mathcal{D}(h, g^{\mathcal{T}}) &= \sqrt{\mathbf{E}(\; l(h(\mathbf{v}_t), g^{\mathcal{T}}(\mathbf{v}_t))\;)} \\ &= \sqrt{\int_V l(h(\mathbf{v}_t), g^{\mathcal{T}}(\mathbf{v}_t)) \, P_{\mathbf{v}_t}(\mathbf{v}_t) \, d\mathbf{v}_t} \end{aligned} \tag{16}$$

where l is defined throughout this paper as the quadratic loss:

$$\forall (a, b) \in \Re^2, \qquad l(a, b) = |a - b|^2.$$

After receiving the sequence Υ_N of training examples, the learning machine $\mathcal{LM}$ selects that particular decision rule h_0 which minimizes $\mathcal{D}(h, g^{\mathcal{T}})$ on the space $\mathcal{H}$ (h designates an element of $\mathcal{H}$ and $g^{\mathcal{T}}$ the transformer's general tendency). Formally, this means finding the minimum of the function:

$$\begin{array}{rccc} \mathcal{D}(\,.\,, g^{\mathcal{T}}): & \mathcal{H} & \to & \Re+ \\ & h & \mapsto & \mathcal{D}(h, g^{\mathcal{T}}) \end{array}$$

and the decision rule h_0 at which this minimum is attained. To do so, $\mathcal{LM}$ implements an algorithm $\mathcal{A}$ whose ultimate goal is to find h_0 on the basis of the finite sequence Υ_N of training examples.

Note that w_t is related to $g^{\mathcal{T}}(\mathbf{v}_t)$ through the following relationship:

$$w = g^{\mathcal{T}}(\mathbf{v}_t) + \epsilon \tag{17}$$

where ϵ is the noise associated with the probabilistic environment $\mathcal{E}$. By the properties of conditional expectation, it follows from equation (17) that:

$$\mathbf{E}(\epsilon \mid \mathbf{v}_t) = 0. \tag{18}$$

Remark 2. The decision rule space $\mathcal{H}$ is considered to be indexed by a subset of $\Re^n$ for some $n \geq 1$, that is, there exist an integer $n \geq 1$ and a subset $T \subseteq \Re^n$, such that the space $\mathcal{H}$ can be expressed as $\mathcal{H} = \{h_{\mathbf{p}} \mid \mathbf{p} \in T\}$. This is the case for most engineering systems.

https://doi.org/10.25088/ComplexSystems.14.1.63

4. Overcoming the first obstacle in minimizing the value of $\mathcal{D}$ over the space $\mathcal{H}$

The objective of the learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$ is to minimize the distance $\mathcal{D}(h, g^{\mathcal{T}})$ over the entire decision rule space $\mathcal{H}$. This distance involves two functions: h and $g^{\mathcal{T}}$. The function h is an element of the space $\mathcal{H}$ and, as such, it is well known to $\mathcal{LM}$: once the components of $\mathbf{v}_t$ are measured, the value of $h(\mathbf{v}_t)$ is readily computable. The problem however is $g^{\mathcal{T}}$. Not only is it an unknown function and impossible to derive from first principles (recall that the systems we are dealing with are complex ones), but there is no operational way of getting even sample measurements or any empirical information about it. $g^{\mathcal{T}}$ is indeed buried in noise. What we can measure, with respect to the transformer's response, is the outcome w_t, and w_t contains in it both the value of $g^{\mathcal{T}}$ and noise, all mixed up.

So how should $\mathcal{LM}$ proceed to minimize $\mathcal{D}(h, g^{\mathcal{T}})$, when the only information it can get is in the form of noise-corrupted measurements of the outcome w_t and, of course, the instance $\mathbf{v}_t$? Theorem 1 will be of great help. Before stating it, we need the following definition.

Definition 1 (Expected risk) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $h \in \mathcal{H}$ be a decision rule. The expected risk $R(h)$ of h is defined as the expected value of the random variable:

$$l(h(\mathbf{v}_t), w_t) = |h(\mathbf{v}_t) - w_t|^2$$

when the vector $z_t = (\mathbf{v}_t, w_t)$ is drawn at random in the sample space $Z = V \times W$ according to the PDF $P_{z_t} = P_{(\mathbf{v}_t, w_t)}$ corresponding to environment $\mathcal{E}$. Formally, it is:

$$R(h) = \mathbf{E}(\, l(h(\mathbf{v}_t), w_t)\,) = \int_{V \times W} l(h(\mathbf{v}_t), w_t)\, P_{(\mathbf{v}_t, w_t)}(\mathbf{v}_t, w_t)\, d\mathbf{v}_t\, dw_t. \quad (19)$$

Also, to simplify the notations, we need the following definition.

Definition 2 (Simplifying notations) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. For every decision rule $h \in \mathcal{H}$, we define the real-valued function l_h on the sample space $Z = V \times W$ as follows:

$$\forall (\mathbf{v}_t, w_t) \in V \times W, \qquad l_h(\mathbf{v}_t, w_t) = l(h(\mathbf{v}_t), w_t). \quad (20)$$

Hence, using z-notation, equations (19) and (20) become:

$$\forall h \in \mathcal{H}, \qquad R(h) = \mathbf{E}(l_h(z_t)) = \int_Z l_h(z_t)\, P_{z_t}(z_t)\, dz_t \quad (21)$$

$$\forall z_t = (\mathbf{v}_t, w_t) \in Z, \qquad l_h(z_t) = l(h(\mathbf{v}), w_t). \quad (22)$$

Theorem 1 (Transition $\mathcal{D}(h, g^{\mathcal{T}}) \longrightarrow R(h)$) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $h_0 \in \mathcal{H}$ be a fixed decision rule. Then the function:

$$h \mapsto \mathcal{D}(h, g^{\mathcal{T}})$$

is minimal at h_0 if and only if the function:

$$h \mapsto R(h)$$

is minimal at h_0.

Proof. Using equation (15), it can be shown that the equality

$$R(h) = \int_{V \times W} [w - g^{\mathcal{T}}(\mathbf{v}_t)]^2 \, P_{(\mathbf{v}_t, w_t)}(\mathbf{v}_t, w_t) \, d\mathbf{v}_t \, dw_t + [\mathcal{D}(h, g^{\mathcal{T}})]^2 \tag{23}$$

holds true for all $h \in \mathcal{H}$. Since the integral

$$\int_{V \times W} [w - g^{\mathcal{T}}(\mathbf{v}_t)]^2 \, P_{(\mathbf{v}_t, w)}(\mathbf{v}_t, w) \, d\mathbf{v}_t \, dw$$

is independent of h, it follows that $\mathcal{D}(h, g^{\mathcal{T}})$ is minimal if and only if $R(h)$ is minimal, and that both functions attain their minimum at the same function h_0. ■

Theorem 1 is very important in simplifying the learning problem that $\mathcal{LM}$ is faced with. What it means is that minimizing $\mathcal{D}(h, g^{\mathcal{T}})$ or, equivalently, the square of it $[\mathcal{D}(h, g^{\mathcal{T}})]^2$ over $\mathcal{H}$ amounts to minimizing $R(h)$ over the decision rule space. Look at the expressions of these two functions $[\mathcal{D}(h, g^{\mathcal{T}})]^2$ and $R(h)$:

$$[\mathcal{D}(h, g^{\mathcal{T}})]^2 = \mathbf{E}(\, l(h(\mathbf{v}_t), g^{\mathcal{T}}(\mathbf{v}_t))) \tag{24}$$

and

$$R(h) = \mathbf{E}(\, l(h(\mathbf{v}_t), w_t)). \tag{25}$$

From these expressions, it can be seen that, in the course of minimizing $\mathcal{D}(h, g^{\mathcal{T}})$, Theorem 1 allows us to replace the unknown and unmeasurable noise-free value $g^{\mathcal{T}}(\mathbf{v}_t)$ by the measurable noise-corrupted value w_t, without losing information on that decision rule h_0 at which the minimum of $\mathcal{D}(h, g^{\mathcal{T}})$ is attained.

The following corollary will be helpful for system uncertainty model development.

Corollary 1 (First inequality) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Then the inequality:

$$[\mathcal{D}(h, g^{\mathcal{T}})]^2 \leq R(h) \tag{26}$$

holds true for any rule $h \in \mathcal{H}$.

Proof. This inequality is a direct consequence of equation (23). ■

https://doi.org/10.25088/ComplexSystems.14.1.63

5. Second obstacle: P_{z_t} is not known to $\mathcal{LM}$

Theorem 1 is still not enough for $\mathcal{LM}$ to proceed to the determination of the rule h_0 that minimizes $\mathcal{D}(h, g^{\mathcal{T}})$. This is because $R(h)$ is function of the PDF P_{z_t}: this PDF embodies all sources of uncertainty in the environment $\mathcal{E}$ and, as such, it is not known. The objective—and the power—of the framework developed here consists in avoiding any strong *a priori* assumption regarding the sources of uncertainty in $\mathcal{E}$. Consequently, in what follows, P_{z_t} is considered fixed but unknown.

Now, having taken this stand on P_{z_t}, we have to find a way of minimizing $R(h)$ on the basis of only a finite number N of training examples $z_1, z_2, \ldots, z_N$. How to do that? By introducing a principle called inductive principle of empirical risk minimization (IPERM). This principle emerged in the 1980s as a result of the extensive research work in [4, 5, 6] and that in [7, 8, 9].

6. Inductive principle of empirical risk minimization

Before we state the IPERM, we need to define the meaning of the empirical risk of a decision rule.

Definition 3 (Empirical risk) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $h \in \mathcal{H}$ be a decision rule and $\boldsymbol{\Upsilon}_N = (z_1, z_2, \ldots, z_N)$ a finite sequence of N training examples generated and measured in the probabilistic environment $\mathcal{E}$ as a result of one realization of this same environment. The *empirical risk* $R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$ of h on the sequence $\boldsymbol{\Upsilon}_N$ is defined as the arithmetic mean of the sequence of numbers:

$$(l_h(z_i))_{i=1,2,\ldots,N}$$

that is,

$$R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h) = \frac{1}{N} \sum_{i=1}^{N} l_h(z_i). \tag{27}$$

Having introduced the concept of empirical risk, we can now define what is meant by an uncertainty model.

Definition 4 (Uncertainty model) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $\boldsymbol{\Upsilon}_N$ be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a fixed real number in the interval $]0,1[$. Let $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$ be a decision rule at which the empirical risk $R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$ reaches its minimum. An *η-uncertainty model* (or simply *uncertainty model*) of the transformer $\mathcal{T}$ is any inequality of the type:

$$\mathcal{D}(h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}, g^{\mathcal{T}}) \leq \varphi(e_1, e_2, \ldots, e_l) \tag{28}$$

that satisfy the following two conditions.

1. $\mathbf{Pr}\left[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}}) \leq \varphi(e_1, e_2, \ldots, e_l)\right] \geq 1 - \eta.$
2. $e_1, e_2, \ldots, e_l$ are a set of uncertainty control variables and φ is a real-valued function of these variables that satisfy the following:

$$\begin{cases} \text{the variables } e_i \text{ and the function } \varphi \\ \text{are readily determinable/computable.} \end{cases} \tag{29}$$

Expected and empirical risks, $R(h)$ and $R^{\Upsilon_N}_{\text{emp}}(h)$, may seem to introduce new concepts in this framework, but they are not if we go back to the concepts of probability theory. To see that, fix a decision rule h in the space $\mathcal{H}$. Since z_t is a random variable, the number $l_h(z_t)$ is then also a random variable. Denote it as ξ, that is,

$$\xi = l_h(z_t).$$

(Recall that h is fixed.) From probability theory, we know that there are two measures of the central tendency of a random variable such as ξ.

- An *empirical measure*. Given a series of realizations $\xi_1, \xi_2, \ldots, \xi_N$ of the variable ξ, this measure is constructed by computing the arithmetic average $(\sum_i \xi_i)/N$ of this series.
- A *mathematical measure*. This measure is expressed in terms of the PDF P_ξ of ξ, that is: $\int \xi P_\xi(\xi)\, d\xi$. It is called the *expected value*.

In this framework, $R^{\Upsilon_N}_{\text{emp}}(h)$ represents the empirical measure of the central tendency of $\xi = l_h(z_t)$ and $R(h)$ represents the mathematical one. The former measure is approximate but computable, the latter is exact but unknown. Also note that, under some conditions with respect to the dependency and heterogeneity of the realizations ξ_i, the empirical measure converges to the mathematical one when N is made infinitely large [13]. This is known as the Law of Large Numbers in probability theory. Applying this law to the case of the expected and empirical risks, we get that $R^{\Upsilon_N}_{\text{emp}}(h)$ converges (in probability) to $R(h)$ as N is made infinitely large. That is,

$$R^{\Upsilon_N}_{\text{emp}}(h) \to R(h) \qquad \text{as} \qquad N \to \infty. \tag{30}$$

The reader should note a very important fact here: the convergence equation (30) is valid for a fixed decision rule h in the space $\mathcal{H}$. This is called *pointwise convergence*, as opposed to another type of convergence (called *uniform convergence*) that is discussed briefly in the next sections. The term "pointwise" refers to the fact that the convergence equation (30) occurs only for fixed points of $\mathcal{H}$ and not for all points of this space simultaneously.

Now we state the IPERM. This principle consists of implementing the following two actions.

https://doi.org/10.25088/ComplexSystems.14.1.63

- **Action 1**. *Replace* the expected risk $R(h)$ by the empirical risk $R_{\text{emp}}^{\Upsilon_N}(h)$ computed on the basis of one training sequence Υ_N.
- **Action 2**. *Take* the decision rule $h_{\text{emp}}^{\Upsilon_N}$ at which $R_{\text{emp}}^{\Upsilon_N}(h)$ reaches its minimum as a good representation of the best rule h_0 that minimizes the expected risk $R(h)$.

Therefore, the implementation of the IPERM comes down to minimizing the empirical risk $R_{\text{emp}}^{\Upsilon_N}(h)$, instead of the expected one $R(h)$, over the space $\mathcal{H}$ and then choosing that decision rule $h_{\text{emp}}^{\Upsilon_N}$ at which the minimum of $R_{\text{emp}}^{\Upsilon_N}(h)$ is reached to describe the transformer's behavior. Engineering systems modelers (in various areas of engineering such as chemical, civil, or environmental) have been using this procedure for system model identification for years. The reader may then wonder: Why are we developing a new mathematical framework, if all we are going to do is turn back to the traditional model identification procedure? What is the point?

This framework is not about inventing new procedures, but rationalizing existing ones and modeling the uncertainty that is associated with them. Engineering systems modelers have been using the traditional identification procedure without being aware of the transitions

$$\mathcal{D}(h, g^{\mathcal{T}}) \longrightarrow R(h) \longrightarrow R_{\text{emp}}^{\Upsilon_N}(h). \tag{31}$$

Their decision to rely on empirical risk minimization may be explained by the fact that mechanistic models (as opposed to balck-box models) are usually assumed to contain adequate *a priori* information about the real system and, as a result, very little information would be lost in the transition

$$R(h) \longrightarrow R_{\text{emp}}^{\Upsilon_N}(h). \tag{32}$$

Now we know that this is not true for a complex system, since all existing models represent just a simplified picture of the real system behavior. If the sequence Υ_N is a finite one, then there is definitely a loss of information in the transition equation (32), that has always been ignored by engineering systems modelers. The aim of this framework is to rationalize and investigate the validity of this transition. First, we determine in what cases the replacement of $R(h)$ by $R_{\text{emp}}^{\Upsilon_N}(h)$ can be legitimized and, second, evaluate the loss of information that occurs in the course of this replacement. To do so, we need to examine the applicability of the IPERM, for which Vapnik's results will be of great help.

7. Applicability of the inductive principle of empirical risk minimization

In the transition

$$\mathcal{D}(h, g^{\mathcal{T}}) \longrightarrow R(h) \tag{33}$$

there is absolutely no information loss, by virtue of Theorem 1. As a result, $R(h)$ can be considered as an exact measure of the performance of the decision rule h when this rule is selected by $\mathcal{LM}$ as an approximation of $g^{\mathcal{T}}$. The transition that is problematic is the second one:

$$R(h) \longrightarrow R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h).$$

$R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$ is indeed just an estimation of $R(h)$. Of course, one may argue that replacing $R(h)$ by $R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$, as suggested in Action 1 of the IPERM, can be legitimized by the fact that, according to the Law of Large Numbers, $R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$ becomes a perfect estimation of $R(h)$ when the size N of the sequence $\boldsymbol{\Upsilon}_N$ is made infinitely large. But, this fact cannot be used to justify Action 2 of the IPERM. Here is indeed the problem.

> As was done above, denote the decision rules that minimize $R(h)$ and $R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)$ as h_0 and $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$, respectively. This is equivalent to writing
>
> $$R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}) = \inf_{h \in \mathcal{H}} R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h) \tag{34}$$
>
> and
>
> $$R(h_0) = \inf_{h \in \mathcal{H}} R(h). \tag{35}$$
>
> Action 2 of the IPERM stipulates taking $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$ as a good representation of the best rule h_0. For this to be justified, we need to ensure that $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$ is very "close" to minimizing the expected risk $R(h)$ which is, as pointed out previously, an exact measure of the rule's performance (meaning the rule's closeness to $g^{\mathcal{T}}$ in the sense of $\mathcal{D}$). In more concrete terms, we need that the value $R(h_{\text{emp}}^{\boldsymbol{\Upsilon}_N})$ of the expected risk at $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$ be close to the minimum one $R(h_0)$, for N sufficiently large. That is,
>
> $$R(h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}) \to R(h_0) \qquad \text{as} \qquad N \to \infty \tag{36}$$
>
> (convergence is understood in probability).

It has been shown [9] that the pointwise convergence equation (30) does not guarantee the one that is really required for the purpose of the IPERM, that is, convergence equation (36). In other words, it is possible that convergence equation (30) be satisfied, but $R(h_{\text{emp}}^{\boldsymbol{\Upsilon}_N})$ remains always far from $R(h_0)$—even for large values of N—meaning that $h_{\text{emp}}^{\boldsymbol{\Upsilon}_N}$ would never constitute a good approximation to the transformer's behavior. It is therefore important to verify whether the IPERM is applicable or not before using it in any learning problem.

Taking into consideration the foregoing comments, the following definition shall be adopted for the meaning of the applicability of the IPERM.

https://doi.org/10.25088/ComplexSystems.14.1.63

Definition 5 (Applicability of the IPERM) Let $\mathcal{E} = (\mathcal{T}, OM, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $\boldsymbol{\Upsilon}_N$ be a finite sequence of N training examples from the environment $\mathcal{E}$ and let $h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}$ and h_0 be two decision rules that minimize the risks $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$ and $R(h)$, respectively (refer to equations (34) and (35)). The IPERM is said to be *applicable* to $(\mathcal{E}, \mathcal{LM})$ if, for any $\varepsilon > 0$, the following equality holds true:

$$\lim_{N\to\infty} \mathbf{Pr}\left(\sup_{h\in\mathcal{H}} \delta\left[R(h), R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)\right] > \varepsilon\right) = 0, \tag{37}$$

with δ being a deviation measure defined on the real line.

Now that the applicability of the IPERM has been defined, we need to develop a simple method of verifying it. In the foregoing discussion, it has been pointed out that the pointwise convergence equation (30) is not enough to guarantee the applicability of the IPERM. A more stringent condition regarding the empirical risk convergence needs to be imposed. Vapnik and Chervonenkis showed in [9] that, for the IPERM to be applicable, it is necessary and sufficient that the empirical risk $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$ converges uniformly to the expected risk $R(h)$ over the whole space $\mathcal{H}$ (convergence is understood in probability). Mathematically, uniform convergence means that equation (37) holds true. Intuitively, it means that, as N is made infinitely large, the whole curve of $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$ converges to that of $R(h)$ over the space $\mathcal{H}$. In this presentation, the theoretical part of such questions will not be detailed. Instead, the reader is referred to Vapnik's book *Statistical Learning Theory* [6] for details. In what follows, Vapnik's results are presented in a more practical fashion, allowing direct application to the cases under study in this paper (i.e., engineering systems). The mathematical rigor is, however, preserved throughout the whole presentation.

A criterion to verify the applicability of the IPERM is not the only thing that is needed here. We also want to know how much information is lost when $R(h)$ is replaced by $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$. Here again, to evaluate this information loss, we need to define a measure of the deviation between $R(h)$ and $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$. For this purpose, two deviation relative measures are introduced.

- *Relative measure* δ_1 defined by:

$$\forall (a_1, a_2) \in \mathbb{R}^2, \qquad \delta_1[a_1, a_2] = \frac{a_1 - a_2}{\sqrt{a_1}}. \tag{38}$$

- *Relative measure* δ_2 defined by:

$$\forall (a_1, a_2) \in \mathbb{R}^2, \qquad \delta_2[a_1, a_2] = \frac{a_1 - a_2}{a_1}. \tag{39}$$

Each of these measures will be associated with a different weak prior information about $(\mathcal{E}, \mathcal{LM})$.

Using these measures, Theorem 2 defines sufficient conditions for the applicability of the IPERM and helps evaluate the loss of information that occurs when $R(h)$ is replaced by $R_{\text{emp}}^{\Upsilon_N}(h)$.

Theorem 2 (Applicability of the IPERM) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let Υ_N be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a real number in the interval $]0, 1[$. Let δ be one of the deviation measures δ_1 or δ_2. If it is possible to establish some weak prior information $\mathcal{WPI}$ about $(\mathcal{E}, \mathcal{LM})$ and construct a function C dependent on N, the whole set $\mathcal{H}$, $\mathcal{WPI}$, and the number η such that both Statements 1 and 2 listed below hold true, then the IPERM is applicable to $(\mathcal{E}, \mathcal{LM})$. When such a function

$$C = C(N, \mathcal{H}, \mathcal{WPI}, \eta)$$

exists, the IPERM is said to be δ-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound $C(N, \mathcal{H}, \mathcal{WPI}, \eta)$.

Statement 1. For any $\eta \in]0, 1[$, the inequality

$$\sup_{h \in \mathcal{H}} \delta\left[R(h), R_{\text{emp}}^{\Upsilon_N}(h)\right] \leq C(N, \mathcal{H}, \mathcal{WPI}, \eta)$$

is satisfied with probability of at least $1 - \eta$.

Statement 2. When $\mathcal{H}$, η, and $\mathcal{WPI}$ are fixed, then:

$$\lim_{N \to \infty} C(N, \mathcal{H}, \mathcal{WPI}, \eta) = 0.$$

Proof. Let $\varepsilon > 0$ and $\eta \in]0, 1[$ be two fixed numbers. From Statement 2, we infer that:

$$\exists N_0 \in \aleph, \qquad \forall N > N_0, \qquad C(N, \mathcal{H}, \mathcal{WPI}, \eta) < \varepsilon.$$

Then, from Statement 1, we get that for $N > N_0$, the inequality

$$\sup_{h \in \mathcal{H}} \delta\left[R(h), R_{\text{emp}}^{\Upsilon_N}(h)\right] \leq \varepsilon$$

is satisfied with probability of at least $1 - \eta$. That is,

$$\mathbf{Pr}\left(\sup_{h \in \mathcal{H}} \delta\left[R(h), R_{\text{emp}}^{\Upsilon_N}(h)\right] > \varepsilon\right) < \eta.$$

Thus, we have shown that, for any $\varepsilon > 0$:

$$\forall \eta \in]0, 1[, \quad \exists N_0 \in \aleph, \quad \forall N > N_0, \quad \mathbf{Pr}\left(\sup_{h \in \mathcal{H}} \delta\left[R(h), R_{\text{emp}}^{\Upsilon_N}(h)\right] > \varepsilon\right) < \eta$$

which means, by definition, that

$$\lim_{N\to\infty} \mathbf{Pr}\left(\sup_{h\in\mathcal{H}} \delta\left[R(h), R^{\Upsilon_N}_{\text{emp}}(h)\right] > \varepsilon\right) = 0. \quad \blacksquare$$

Now recall that the objective of this study is to develop uncertainty models (see Definition 4) for complex engineering systems. The following theorem defines a way of developing such models.

Theorem 3 (Uncertainty model) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let Υ_N be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a real number in the interval $]0, 1[$. Let $\mathcal{WPI}$ be some weak prior information about $(\mathcal{E}, \mathcal{LM})$ and $h^{\Upsilon_N}_{\text{emp}}$ a decision rule at which the empirical risk $R^{\Upsilon_N}_{\text{emp}}(h)$ reaches its minimum.

- If the IPERM is δ_1-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound $C(N, \mathcal{H}, \mathcal{WPI}, \eta)$, then the inequality

$$[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}) + \frac{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}{2} \times \left(1 + \sqrt{1 + \frac{4\, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}}\right) \tag{40}$$

  holds true with probability of at least $1 - \eta$.

- If the IPERM is δ_2-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound $C(N, \mathcal{H}, \eta, \mathcal{WPI})$, then the inequality

$$[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le \frac{R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{(1 - C(N, \mathcal{H}, \mathcal{WPI}, \eta))_+} \tag{41}$$

  holds true with probability of at least $1 - \eta$, where $(a)_+ = \sup(a, 0)$.

Proof. If the IPERM is δ_1-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound $C(N, \mathcal{H}, \eta, \mathcal{WPI})$, then, from Theorem 2, it follows that (all inequalities hold with probability of at least $1 - \eta$):

$$\frac{R(h^{\Upsilon_N}_{\text{emp}}) - R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{\sqrt{R(h^{\Upsilon_N}_{\text{emp}})}} \le C(N, \mathcal{H}, \mathcal{WPI}, \eta).$$

Hence:

$$R(h^{\Upsilon_N}_{\text{emp}}) \le R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}) + \frac{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}{2}\left(1 + \sqrt{1 + \frac{4\, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}}\right)$$

and then, from Theorem 1, it follows that

$$[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}) + \frac{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}{2}\left(1 + \sqrt{1 + \frac{4\, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{C^2(N, \mathcal{H}, \mathcal{WPI}, \eta)}}\right).$$

Similarly, if the IPERM is δ_2-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound $C(N, \mathcal{H}, \eta, \mathcal{WPI})$, then, from Theorem 2, it follows that

$$\frac{R(h^{\Upsilon_N}_{\text{emp}}) - R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{R(h^{\Upsilon_N}_{\text{emp}})} \le C(N, \mathcal{H}, \mathcal{WPI}, \eta)$$

and then

$$[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le R(h^{\Upsilon_N}_{\text{emp}}) \le \frac{R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{(1 - C(N, \mathcal{H}, \mathcal{WPI}, \eta))_+}. \quad \blacksquare$$

The bound on the squared distance $[\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2$, when it exists, is called the *guaranteed deviation* between $h^{\Upsilon_N}_{\text{emp}}$ and $g^{\mathcal{T}}$, and denoted as φ or as

$$\varphi(N, \mathcal{H}, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}), \mathcal{WPI}, \eta).$$

8. The Vapnik–Chervonenkis dimension

One of the objects which the guaranteed deviation φ is dependent on is the whole set $\mathcal{H}$ of decision rules. Now we need to know exactly what characteristic of $\mathcal{H}$ affects φ and the uncertainty equations (40) and (41). Intuitive analysis of uncertainty in engineering systems shows that this characteristic is the complexity of $\mathcal{H}$ [12]. The objective of this section is to define a measure of this complexity. This measure is known as the Vapnik–Chervonenkis dimension, or simply VC dimension, named in honor of its originators, Vapnik and Chervonenkis [7]. The definition of this dimension is quite difficult to assimilate from the first reading. Because of this, an intuitive interpretation of the VC dimension will be given first and, at the end of this section, a series of illustrative examples are presented.

8.1 Intuitive introduction

Consider the following concrete example:

- $V_1 = \Re$ and $W_1 = \Re$;
- $\mathcal{H} = \mathcal{H}_{\text{line}}$ is the set of all functions h from V into W such that:

  $$\forall x \in V, \qquad h(x) = p_1 x + p_2$$

  with $\mathbf{p} = (p_1, p_2) \in \Re^2$ as the parameter vector.

If we had to assign a number to the complexity of this set of functions, then intuitively the number two, corresponding to the number of parameters, would be the most suitable one. Consider now this second example:

- $V_2 = \Re$ and $W_2 = \Re$;
- $\mathcal{H} = \mathcal{H}_{\text{sine}}$ is the set of all functions h from V into W such that:

$$\forall x \in V, \qquad h(x) = p_1 \sin(p_2 x)$$

with $\mathbf{p} = (p_1, p_2) \in \Re^2$ as the parameter vector.

Since the number of parameters that define this set is also two, we may be tempted to again assign the number two to the complexity of this set. If we do so, it would mean that $\mathcal{H}_{\text{line}}$ and $\mathcal{H}_{\text{sine}}$ have the same degree of complexity, which is obviously not correct: the set $\mathcal{H}_{\text{line}}$ is a family of just straight lines, while $\mathcal{H}_{\text{sine}}$ is a complex family of curves that can take many different shapes. The "expressive power" of $\mathcal{H}_{\text{sine}}$ is indeed much higher than that of $\mathcal{H}_{\text{line}}$. As a result, it should be expected that the complexity of $\mathcal{H}_{\text{sine}}$ be much higher than that of $\mathcal{H}_{\text{line}}$, and that is what we get when we consider the VC dimension as a measure of the complexity of the decision rule space.

Intuitively, the VC dimension may be considered as equal to the maximum number of points that the curves representing the functions of the decision rule space can pass through simultaneously. Straight lines (functions defined by $h(x) = p_1 x + p_2$, space $\mathcal{H}_{\text{line}}$) can pass through any two points, but not any three points. Parabolas (functions defined by $h(x) = p_1 x^2 + p_2 x + p_3$, space $\mathcal{H}_{\text{parab}}$) can pass through any three points, but not any four points. Sine functions ($h(x) = p_1 \sin(p_2 x)$, space $\mathcal{H}_{\text{sine}}$) can pass through any number of points. Hence, if the VC dimension of a space $\mathcal{H}$ is denoted as $q(\mathcal{H})$, then:

$$\begin{aligned} q(\mathcal{H}_{\text{line}}) &= 2 \\ q(\mathcal{H}_{\text{parab}}) &= 3 \\ q(\mathcal{H}_{\text{sine}}) &= \infty. \end{aligned}$$

The foregoing intuitive interpretation of the VC dimension is approximate. A more precise definition of it is given in section 8.2.

8.2 Definitions

For every set I, the notation 2^I will designate the set of all subsets of I.

Definition 6 (VC dimension of a family of sets) Let $\mathbf{G}$ be some space ($\Re^n$ with $n > 0$, for example, or any other space). Let $\mathcal{G}$ be a family of subsets of $\mathbf{G}$ (examples of $\mathcal{G}$ in the case of $\mathbf{G} = \Re^2$ are the family of all open [or

closed] balls of $\Re^2$ or the family of all half planes of $\Re^2$) and I is a finite subset of $\mathbf{G}$. Let $\Pi^{\mathcal{G}}(I)$ be the subset of 2^I defined as follows:

$$\Pi^{\mathcal{G}}(I) = \{\Lambda \in 2^I | \exists F \in \mathcal{G}, \Lambda = F \cap I\}.$$

The finite set I is said to be *shattered* by the family of sets $\mathcal{G}$ if $\Pi^{\mathcal{G}}(I) = 2^I$. The largest integer q such that some finite subset $I \subset \mathbf{G}$ of *size* q is shattered by $\mathcal{G}$ is called the *Vapnik–Chervonenkis dimension* (VC dimension) of the family $\mathcal{G}$. It is denoted by $q = q(\mathcal{G})$. If such an integer q does not exist, then the VC dimension of $\mathcal{G}$ is said to be infinite.

Definition 7 (VC dimension of a family of functions) Let $\mathcal{F}$ be a family of real-valued functions on some space $\mathbf{G}$ and I is a finite subset of $\mathbf{G}$. For every function $f \in \mathcal{F}$, define the subset $\text{pos}(f)$ of the space $\mathbf{G}$ as follows:

$$\text{pos}(f) = \{a \in \mathbf{G} | \, f(a) > 0\}.$$

Then define the family $\text{pos}(\mathcal{F})$ of subsets of $\mathbf{G}$ as follows:

$$\text{pos}(\mathcal{F}) = \{\text{pos}(f) | \, f \in \mathcal{F}\}.$$

The finite set I is said to be shattered by the family of real-valued functions $\mathcal{F}$, if it is shattered by the family of subsets $\text{pos}(\mathcal{F})$. The VC dimension $q(\mathcal{F})$ of the family $\mathcal{F}$ of real-valued functions is, by definition, equal to the VC dimension of the family of subsets $\text{pos}(\mathcal{F})$:

$$q(\mathcal{F}) = q(\text{pos}(\mathcal{F})).$$

The VC dimension is then a purely combinatorial concept that has, *a priori*, no connection with the geometric notion of dimension. In most situations, it is difficult to evaluate the VC dimension by analytic means. Usually, all that is possible is to determine a bound on the VC dimension, that is, establish an inequality of the form: $q(\mathcal{F}) \leq q_0$ ($q_0 \in \aleph$). Also in some cases the VC dimension is simply approximated by the free parameters of the family $\mathcal{F}$. Theorem 4 shows how to determine it in some particular cases. It also establishes a link with the geometric notion of dimension.

Theorem 4 (VC dimension and vector space) Let $\mathcal{F}$ be a family of real-valued functions on some space $\mathbf{G}$. Fix any function f_0 from $\mathbf{G}$ into $\Re$ and let $\mathcal{F}_0$ be the new family of functions defined by $\mathcal{F}_0 = f_0 + \mathcal{F} = \{f_0 + f | \, f \in \mathcal{F}\}$. If $\mathcal{F}$ is an m-dimensional real vector space, then the VC dimension $q(\mathcal{F}_0)$ of $\mathcal{F}_0$ is equal to m:

$$q(\mathcal{F}_0) = m.$$

Proof. Refer to [14] for the proof of this theorem. ■

8.3 Examples

Example 1. Consider the family of functions $h_{\mathbf{p}}$ defined from the space $\mathbf{G} = \Re^n$ ($n \in \aleph^\circ$) into $\{0, 1\}$ by:

$$\forall \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Re^n, \qquad h_{\mathbf{p}}(\mathbf{x}) = \psi\left(\sum_{i=1}^{n} p_i x_i\right)$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_n, \theta) \in \Re^{n+1}$ is the parameter vector and ψ is defined by (real threshold θ):

$$\psi(a) = \begin{cases} 1 & \text{if } a \geq \theta, \\ 0 & \text{if } a < \theta. \end{cases}$$

This family of functions is known as the *perceptron* and is used in pattern recognition. Its VC dimension is equal to $n + 1$ [15].

Example 2. Consider the family of real-valued functions $h_{\mathbf{p}}$ defined on some space $\mathbf{G}$ by

$$\forall \mathbf{x} \in \mathbf{G}, \qquad h_{\mathbf{p}}(\mathbf{x}) = \sum_{i=1}^{n} p_i \psi_i(\mathbf{x})$$

where $\mathbf{p} = (p_1, p_2, \ldots, p_n) \in \Re^n$ is the parameter vector and $\psi_1, \psi_2, \ldots, \psi_n$ is a sequence of n linearly independent real-valued functions. The VC dimension of this family of functions is equal to n [4]. Note that the determination of this VC dimension results directly from Theorem 4.

Example 3. Consider the family of functions $h_{\mathbf{p}}$ defined on $\mathbf{G} = \Re^2$ by

$$\forall (x, y) \in \Re^2, \qquad h_{\mathbf{p}}(x, y) = (y - \mathrm{poly}_n(x, \mathbf{p}))^2$$

where $\mathbf{p} = (p_0, p_1, p_2, \ldots, p_n) \in \Re^{n+1}$ is the parameter vector and $\mathrm{poly}_m(x, \mathbf{p})$ is a polynomial function of degree n defined by

$$\forall x \in \Re, \qquad \mathrm{poly}_n(x, \mathbf{p}) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n.$$

The VC dimension of this family of functions $h_{\mathbf{p}}$ is at most $2n + 2$ [5].

Example 4. Consider the family of functions $h_{\mathbf{p}}$ defined on $\mathbf{G} = \Re$ by

$$\forall x \in \Re, \qquad h_{\mathbf{p}}(x) = p_1 \sin(p_2 x)$$

where $\mathbf{p} = (p_1, p_2) \in \Re^2$ is the parameter vector. The VC dimension of this family of functions is infinite [6].

From these examples it can be seen that, generally speaking, the VC dimension of a family of functions is not always related to the number of parameters. It can be larger (Example 4), equal (Examples 1 and 2), or

smaller (see [5] where new types of learning machines were constructed) than the number of parameters.

9. Vapnik–Chervonenkis dimension and applicability of the inductive principle of empirical risk minimization

In section 7, the concept of applicability of the IPERM and that of guaranteed deviation between the decision rule $h_{\text{emp}}^{\Upsilon_N}$ that minimizes the empirical risk and the transformer's response function $g^{\mathcal{T}}$ were introduced. However, no methodology has been developed to determine the expression of the function $C = C(N, \mathcal{H}, \mathcal{WPI}, \eta)$ (see Theorems 2 and 3), which is the key function in implementing those concepts. In this section, some fundamental results with respect to determining such a function are presented. These results make use of the VC dimension concept defined in section 8 and are due to [6]. Extensive discussion and application of these results to model identification and quality evaluation can be found in [12].

Before stating these results, we need to define a new space $l_{\mathcal{H}}$ and five different conditions.

Definition 8 (Space $l_{\mathcal{H}}$) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. For every decision rule $h \in \mathcal{H}$ and a real number $\beta \in \Re^+$, we define the real-valued functions $l_{h,\beta}$ on the sample space $Z = V \times W$ as follows:

$$\forall z_t \in Z, \qquad l_{h,\beta}(z_t) = l_h(z_t) - \beta.$$

The functional space of all functions $l_{h,\beta}$ will be denoted by $l_{\mathcal{H}}$:

$$l_{\mathcal{H}} = \{l_{h,\beta} | \ (h,\beta) \in \mathcal{H} \times \Re^+\}.$$

Now we define the conditions C.1, C′.1, C.2, C.3, and C′.3.

C.1 *Weak prior information (1)*. There exists a positive number $M \in]0, +\infty[$ such that:

$$\sup_{h \in \mathcal{H}, z_t \in Z} l_h(z_t) = M.$$

C′.1 *Weak prior information (2)*. There exists a pair $(s, \tau) \in \Re^2$ with $s > 2$ and $\tau < +\infty$ such that:

$$\sup_{h \in \mathcal{H}} \frac{\mathbf{E}^{1/s}([l_h(z_t)]^s)}{R(h)} < \tau.$$

C.2 *VC dimension*. The VC dimension $q = q(l_{\mathcal{H}})$ of the functional space $l_{\mathcal{H}}$ is finite.

C.3 *I.i.d. condition*. The training examples $z_1, z_2, \ldots, z_N$ of the sequence $\boldsymbol{\Upsilon}_N$ are independent and identically distributed (i.i.d.).

C′.3 *Weaker i.i.d. condition*. The real-valued random variables

$$l_h(z_1);\ l_h(z_2);\ \cdots;\ l_h(z_N)$$

obtained by computing the values of l_h at each one of the training examples z_i of the sequence $\boldsymbol{\Upsilon}_N$, are i.i.d. for any $h \in \mathcal{H}$.

Corollary 2 (IPERM applicability and VC (1)) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $\boldsymbol{\Upsilon}_N$ be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a real number in the interval $]0, 1[$. If the conditions C.1, C.2, and C.3 are satisfied, then the IPERM is δ_1-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound:

$$C = \sqrt{M\zeta} \tag{42}$$

where the number ζ is:

$$\zeta = 4\frac{\left[q\left(\ln\left(\frac{2N}{q}\right)+1\right) - \ln\left(\frac{\eta}{4}\right)\right]}{N}$$

and q is the VC dimension $q(l_{\mathcal{H}})$ of the space $l_{\mathcal{H}}$.

Proof. In [6] Vapnik showed that, for any $\varepsilon > 0$, the following inequality holds true:

$$\mathbf{Pr}\left(\sup_{h\in\mathcal{H}} \delta_1[R(h), R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)] > \varepsilon\right) < 4\ \exp\left[\left(\frac{q\left(\ln\left(\frac{2N}{q}\right)+1\right)}{N} - \frac{\varepsilon^2}{4M}\right)N\right] \tag{43}$$

when conditions C.1, C.2, and C.3 are satisfied [6], (see inequalities 5.24 and 5.12 at pages 197 and 192 of [6] respectively). Set the right-hand side of the above inequality equal to η. Then the expression of ε is

$$\varepsilon = \sqrt{M\zeta}$$

and, therefore, from Vapnik's inequality, it follows that the inequality

$$\sup_{h\in\mathcal{H}} \delta_1[R(h), R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)] < \sqrt{M\zeta}$$

holds true with probability of at least $1 - \eta$. ■

Corollary 3 (IPERM applicability and VC (2)) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} =$

$(\mathcal{H}, \mathcal{A})$. Let $\boldsymbol{\Upsilon}_N$ be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a real number in the interval $]0, 1[$. If the conditions $C'.1$, $C.2$, and $C.3$ are satisfied, then the IPERM is δ_2-applicable to $(\mathcal{E}, \mathcal{LM})$ with the bound

$$C = \gamma(s)\,\tau\sqrt{\zeta} \tag{44}$$

where

$$\gamma(s) = \sqrt[s]{\frac{1}{2}\left(\frac{s-1}{s-2}\right)^{s-1}},$$

the number ζ is:

$$\zeta = 4\frac{\left[q\left(\ln\left(\frac{2N}{q}\right)+1\right) - \ln\left(\frac{\eta}{4}\right)\right]}{N},$$

and q is the VC dimension $q(l_{\mathcal{H}})$ of the space $l_{\mathcal{H}}$.

Proof. In [6] Vapnik showed that, for any $\varepsilon > 0$, the following inequality holds true:

$$\mathbf{Pr}\left(\sup_{h\in\mathcal{H}} \delta_2[R(h), R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)] > \gamma(s)\,\tau\,\varepsilon\right) < 4\,\exp\left[\left(\frac{q\left(\ln\left(\frac{2N}{q}\right)+1\right)}{N} - \frac{\varepsilon^2}{4}\right)N\right] \tag{45}$$

when conditions $C'.1$, $C.2$, and $C.3$ are satisfied [6], (see inequalities 5.43 and 5.12 at pages 210 and 192 of [6] respectively). Set the right-hand side of the above inequality equal to η. Then the expression of ε is:

$$\varepsilon = \sqrt{\zeta}$$

and, therefore, the inequality:

$$\sup_{h\in\mathcal{H}} \delta_2[R(h), R_{\text{emp}}^{\boldsymbol{\Upsilon}_N}(h)] < \gamma(s)\,\tau\sqrt{\zeta}$$

holds true with probability of at least $1 - \eta$. ■

Note that $\mathcal{WPI}$ is represented by the number M in Theorem 2 and by the numbers s and τ in Theorem 3.

Theorem 5 uses a weaker i.i.d. condition ($C'.3$).

Theorem 5 (Using condition $C'.3$) If the third condition $C.3$ in Theorems 2 and 3 is replaced by the condition $C'.3$ and the two other conditions, $C.1$ and $C.2$ for Theorem 2 and $C'.1$ and $C.2$ for Theorem 3, are kept unchanged, then the IPERM is still applicable to $(\mathcal{E}, \mathcal{LM})$ with respect to the same deviation measures δ_1 and δ_2 and with the same bounds of equations (42) and (43), respectively.

Proof. To prove equations (43) and (45), Vapnik [4, 5] made use of the weaker i.i.d. condition only. As a result, these inequalities remain true if condition C.3 is replaced by condition C′.3. Consequently, the foregoing proofs of Theorems 2 and 3 are still valid with condition C′.3. ■

Using Theorems 2, 3, and 5, it is now possible to develop uncertainty models for $(\mathcal{E}, \mathcal{LM})$ with a guaranteed deviation φ that is readily computable.

Corollary 4 (Uncertainty model and VC) Let $\mathcal{E} = (\mathcal{T}, \mathcal{OM}, z_t, P_{z_t})$ be a probabilistic environment and, associated with it, a learning machine $\mathcal{LM} = (\mathcal{H}, \mathcal{A})$. Let $\boldsymbol{\Upsilon}_N$ be a finite sequence of N training examples from the environment $\mathcal{E}$ and η is a real number in the interval $]0, 1[$. Let $h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}$ be a decision rule at which the empirical risk $R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h)$ reaches its minimum.

- If the conditions C.1, C.2, and C′.3 are satisfied, then the inequality

$$[\mathcal{D}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}) + \frac{M\zeta}{2}\left(1 + \sqrt{1 + \frac{4\, R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}})}{M\zeta}}\right) \tag{46}$$

  holds true with probability of at least $1 - \eta$.

- If the conditions C′.1, C.2, and C′.3 are satisfied, then the inequality

$$[\mathcal{D}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le \frac{R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}})}{(1 - \gamma(s)\,\tau\sqrt{\zeta})_+} \tag{47}$$

  holds true with probability of at least $1 - \eta$.

  - ⋆ $(a)_+ = \sup(a, 0)$ for any number $a \in \mathbb{R}$.
  - ⋆ $\gamma(s) = \sqrt[s]{\frac{1}{2}\left(\frac{s-1}{s-2}\right)^{s-1}}$.
  - ⋆ The number ζ is

$$\zeta = 4\frac{\left[q\left(\ln\left(\frac{2N}{q}\right) + 1\right) - \ln\left(\frac{\eta}{4}\right)\right]}{N}. \tag{48}$$

  - ⋆ q is the *VC* dimension $q(l_{\mathcal{H}})$ of the space $l_{\mathcal{H}}$.

Proof. This theorem is a direct consequence of Theorems 5 and 3. ■

Theorem 4 establishes two uncertainty models, $\mathcal{UM}_1$ and $\mathcal{UM}_2$, for $(\mathcal{E}, \mathcal{LM})$. The first one, $\mathcal{UM}_1$, is based on the weak prior information $\mathcal{WPI}(1)$ and is defined by equation (46). The right-hand side of this inequality represents the guaranteed deviation φ_1 between $h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}$ and $g^{\mathcal{T}}$, developed on the basis of $\mathcal{WPI}(1)$. Using this function φ_1, the uncertainty model $\mathcal{UM}_1$ can be rewritten as follows:

$$\mathcal{UM}_1 : [\mathcal{D}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le \varphi_1(N, \mathcal{H}, R^{\boldsymbol{\Upsilon}_N}_{\text{emp}}(h^{\boldsymbol{\Upsilon}_N}_{\text{emp}}), \mathcal{WPI}(1), \eta) \tag{49}$$

with

$$\varphi_1(N, \mathcal{H}, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}), \mathcal{WPI}(1), \eta) = R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}) + \frac{M\zeta}{2}\left(1 + \sqrt{1 + \frac{4\, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{M\zeta}}\right). \tag{50}$$

The second model, $\mathcal{UM}_2$, is based on the weak prior information $\mathcal{WPI}(2)$ and is defined by equation (47). Denoting the right-hand side of this inequality as φ_2 (guaranteed deviation developed on the basis of $\mathcal{WPI}(2)$), the uncertainty model $\mathcal{UM}_2$ can be rewritten as

$$\mathcal{UM}_2 : [\mathcal{D}(h^{\Upsilon_N}_{\text{emp}}, g^{\mathcal{T}})]^2 \le \varphi_2(N, \mathcal{H}, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}), \mathcal{WPI}(2), \eta) \tag{51}$$

with

$$\varphi_2(N, \mathcal{H}, R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}}), \mathcal{WPI}(2), \eta) = \frac{R^{\Upsilon_N}_{\text{emp}}(h^{\Upsilon_N}_{\text{emp}})}{(1 - \gamma(s)\,\tau\sqrt{\zeta})_+}. \tag{52}$$

10. Conclusions

A mathematical framework for modeling the uncertainty in complex engineering systems is developed. This framework uses the results of computational learning theory and is based on the premise that a system model is a learning machine. A definition of an uncertainty model is given and a principle called inductive principle of empirical risk minimization (IPERM) is introduced. The applicability of this principle is examined and the concept of "guaranteed deviation" defined. The system model complexity is measured using the Vapnik–Chervonenkis (VC) dimension. Based on this dimension, two different uncertainty models were developed.

Acknowledgments

This work was financially supported by CIDA and NSERC.

References

[1] U. Jeppsson, *Modelling Aspects of Wastewater Treatment Processes*, Ph.D. Thesis, Lund Institute of Technology, Department of Industrial Electrical Engineering and Automation, Lund, Sweden, 1996.

[2] C. Zheng and G. Bennett, *Applied Contaminant Transport Modeling—Theory and Practice* (Van Nostrand Reinhold, New York, 1995).

[3] L. Konikow and J. Bredehoeft, "Ground-water Models Cannot Be Validated," *Advances in Water Resources*, **19**(2) (1992) 75–83.

[4] N. Vapnik, *Estimation of Dependencies Based on Empirical Data* (Springer-Verlag, New York, 1982).

[5] N. Vapnik, *The Nature of Statistical Learning Theory* (Springer-Verlag, New York, 1995).

[6] N. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).

[7] N. Vapnik and A. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to their Probabilities," *Soviet Mathematics Doklady*, **9** (1968) 915–918.

[8] N. Vapnik and A. Chervonenkis, "Necessary and Sufficient Conditions for the Uniform Convergence of the Means to their Expectations," *Theory of Probability and Its Applications*, **26** (1981) 532–553.

[9] N. Vapnik and A. Chervonenkis, "The Necessary and Sufficient Conditions for Consistency of the Method of Empirical Risk Minimization," *Pattern Recognition and Image Analysis*, **1**(3) (1991) 284–305.

[10] A. A. Guergachi and G. G. Patry, "Using Statistical Learning Theory to Rationalize System Model Identification and Validation. Part II: Application to Biological Wastewater Treatment Systems," *Complex Systems*, to be submitted.

[11] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers* (Prentice-Hall, New Jersey, 1994).

[12] A. Guergachi, *Uncertainty Management in the Activated Sludge Process—Innovative Applications of Computational Learning Theory*, Ph.D. Thesis, University of Ottawa, Ottawa, Canada, 2000.

[13] H. White, *Asymtotic Theory for Econometricians* (Academic Press, Orlando, 1984).

[14] R. Wenocur and R. Dudley, "Some Special Vapnik–Chervonenkis Classes," *Discrete Mathematics*, **33** (1981) 313–318.

[15] M. Anthony and N. Biggs, *Computational Learning Theory: An Introduction* (Cambridge University Press, Cambridge, 1992).