entropy

Article

Information Theoretical Measures for Achieving Robust Learning Machines

Pablo Zegers [1,*], B. Roy Frieden [2], Carlos Alarcón [3] and Alexis Fuentes [1]

[1] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Mons. Álvaro del Portillo 12455, Las Condes, Santiago 7620001, Chile; afuentes@miuandes.cl

[2] College of Optical Sciences, University of Arizona, 1630 E. University Blvd., Tucson, AZ 85721, USA; friedenr@optics.arizona.edu

[3] Independent, Santiago 7620001, Chile; ceag.201175@gmail.com

[*] Correspondence: pzegers@miuandes.cl or pablozegers@gmail.com; Tel.: +56-2-2618-2104

Academic Editors: Badong Chen and Jose C. Principe

Received: 13 May 2016; Accepted: 8 August 2016; Published: 12 August 2016

Abstract: Information theoretical measures are used to design, from first principles, an objective function that can drive a learning machine process to a solution that is robust to perturbations in parameters. Full analytic derivations are given and tested with computational examples showing that indeed the procedure is successful. The final solution, implemented by a robust learning machine, expresses a balance between Shannon differential entropy and Fisher information. This is also surprising in being an analytical relation, given the purely numerical operations of the learning machine.

Keywords: information theoretical learning; Shannon entropy; Kullback–Leibler divergence; relative entropy; cross-entropy; Fisher information; relative information

1. Introduction

This work is focused on designing learning machines that are robust to perturbations in their parameters using information theoretical principles.

Traditionally, the uncertainty in the output of a system product of the presence of uncertainty in its inputs is studied by a field called sensitivity analysis. In the case of machine learning, some aspects of the sensitivity problem started to be studied by Vapnik and Chervonenkis [1,2] in the nineteen sixties and made their way to all of the community, clarifying the relation between: (i) the intrinsic difficulty of a problem, i.e., the number of samples required to learn it; (ii) the capacity of a learning machine, i.e., its Vapnik–Chervonenkis dimension; and (iii) how the empirical measurements obtained in the training stages represented the ideal values. A brilliant application of these ideas is the support vector machine [3], developed in the nineteen nineties and now widely used. The influence of the statistical learning theory developed by Vapnik and Chervonenkis has been extremely important for understanding core aspects of the learning machine problem, but does not answer all of them. As an example, from the point of view of sensitivity analysis, the effect in the output of noise in the inputs is another class of important problems that also needs to be addressed. Not surprisingly, it has also been studied for a long time, and it is now possible to find impressive solutions that can produce solid results under very constrained cases. One example of these systems are the deep denoising autoencoders [4], learning machines that have proven to work unexpectedly well with corrupted images.

The research pointed out in the previous paragraph is extremely important, but given that this work is focused on obtaining information theoretical criteria, it is important what has been done in this respect in the field. Not surprisingly, there is a long and abundant trail of works with this focus, as well. An example is the work of Csiszar et al. [5–7] that relates large deviations with the Sanov theorem in the context of Markov distributions. A recent example can be seen in the work of da Veiga [8], who extends previous sensitivity analysis based on the variance using the work of Csiszar.

Nevertheless, none of the works seen by the authors explicitly study the sensitivity of the systems to parameter perturbations. Whereas previous works focus on the sensitivity of a system to uncertainty in the inputs, this work focuses on studying the sensitivity of the behavior of a system to parameter perturbations. In order to do so, this work uses Fisher information and related quantities to establish bounds that control a training process in a direction that produces learning machines robust to uncertainty in the parameters. As far as the authors can tell, this is the first work that addresses the uncertainty of a learning machine problem product of uncertainty in its parameters using Fisher information-related quantities. Interestingly, the procedure that is presented in this work not only helped to control the uncertainty produced by perturbations in the parameters, but also that induced by changes in the inputs. This goes in agreement with the effects of "dropout" [9], a recently-discovered technique, which is very popular in deep learning, that uses changes in the parameters to overcome overfitting in pattern recognition problems.

This work develops a learning algorithm based on information measures. It starts showing why information theoretical approaches to machine learning are preferred approaches and continues with the development of information theoretical performance measures that guide learning processes towards robust solutions. The advantages of using such an approach are exemplified with the help of numerical simulations.

2. Learning Machine Problem Definition

2.1. The Components of a Machine Learning Problem

From a mathematical point of view, every machine learning problem is composed of the following elements:

1. A source density function f_{S} with support $\mathcal{S}$ that needs to be characterized.
2. A set of i.i.d. samples $\{\mathbf{s}\}_n \equiv \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ generated by f_{S}. Sometimes, $\{\mathbf{s}\}_n$ is completely available from the start, i.e., the n samples are readily available for processing. In other cases, the data are acquired incrementally, sample by sample, and data-related processing is performed accordingly.
3. A family of density functions $f_{\text{L};\boldsymbol{\psi},\boldsymbol{\theta}}$ that serves as the learning machine. Each of these functions is indexed by the set of parameters $\boldsymbol{\psi} \in \Psi$, and the parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^m$. The parameter sets differ in that $\boldsymbol{\psi}$ groups all of the parameters not related to metric spaces, i.e., discrete parameters, while $\boldsymbol{\theta}$ groups those that are associated with metric distances, i.e., continuous parameters. As an example, imagine a learning machine that implements a mathematical function defined by a sum of one-dimensional Gaussians. In that case, the $\boldsymbol{\psi} = \{n\}$, with n the number of Gaussians, and $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$, where $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ define the mixing coefficients, means and standard deviations of the Gaussians correspondingly. In machine learning jargon: $\boldsymbol{\psi}$ defines the architecture of the learning machine and $\boldsymbol{\theta}$ those parameters that are adjusted by means of a learning process.

Remark 1. *It is important to point out that f_{S} does not depend either on $\boldsymbol{\psi}$ or $\boldsymbol{\theta}$. Only $f_{\text{L};\boldsymbol{\psi},\boldsymbol{\theta}}$ depends on these parameters.*

Remark 2. *The scope of this work is limited to studying how $\boldsymbol{\theta}$ affects a learning machine. This might seem to be a rather arbitrary decision, but it corresponds to the typical approach followed in the machine learning field. Finding discrete values for $\boldsymbol{\psi}$ in a learning machine is difficult, and it involves discrete optimization techniques specifically refined for those discrete searches. Many times, these values are arbitrarily selected by the learning machine designer in an ad hoc manner. In other cases, these values are chosen with the help of evolutionary* *techniques or discrete optimization techniques. In stark contrast, finding* $\boldsymbol{\theta}$ *can be done using the traditional optimization techniques defined for metric spaces, normally based on some type of gradient descent algorithm. Given the different treatments, the machine learning community has traditionally divided the search of these parameters into two different sub-problems. The same division is followed in this work, and the analysis focuses on the relation between the behavior of a learning machine and its continuous parameters* $\boldsymbol{\theta}$*.*

2.2. The Learning Machine Problem

This is characterized as follows:

1. A physical implementation, which demands:
   (a) Available elements that are rich enough to allow the implementation of all possible mathematical functions. This is regarded in the literature as implementing a universal function approximator.
   (b) Dependence on existing elements, that is on given physical and economic constraints.
2. Its internal organization mainly responds to the constraints imposed by the curse of dimensionality, which normally forces an incremental development:
   (a) Subsystems are built incrementally, where modules are added either in series (cascades), in parallel (as in boosting) or connected through feedback.
   (b) The entire system needs to be robust to changes in $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, i.e., changes in either the product of an inexact fabrication process, or failure while operating, or simply parameter noise when operating, etc.
3. The operation of a learning machine, which has two main aspects:
   (a) Its behavior has a two-fold component. One is external and reflects those actions that make the learning machine behave in a way that complies with Equation (1) and its implications, and another that reflects its inner changes in adaptation, also known as the learning process. Even though the current literature normally divides the two types of behavior into two different stages (first train, then use), nothing stops a general learning machine from producing both types of behavior at the same time.
   (b) Use with maximum efficiency the spatio-temporal resources upon which the machine is built.
4. The behavior of the learning machine is such that:
   (a) It hopefully duplicates the source such that:

$$f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}} = f_{\mathrm{S}} \tag{1}$$

   throughout the support $\mathcal{S}$.
   (b) It is robust to:
      (i) Novel changes in the external signals in case they exist. These external changes can be explained by new, unseen data not present in the dataset $\{\mathbf{s}\}_n$. When a learning machine is robust in this sense, it is said that it has generalized.
      (ii) Noise in the external, if they exist, or internal signals.

3. Robust Information Theoretical Learning

When it comes to training a learning machine, a.k.a finding an optimal set of parameters $\boldsymbol{\theta}$ that minimizes some performance index, there are many alternatives. Specifically, just for the performance index, there are so many, each of them portraying some advantage, that it is legitimate to ask whether there is at least a preferred one. The simplest and most traditional approach that is commonly used is minimizing the Euclidean error. However, it is more logical to use performance measures that directly take into account that the problem is to identify a density function. With this in mind, researchers have for a long time used many approaches drawn from information theory [10–16].

3.1. Basic Information Theoretical Definitions

Before establishing a useful information theoretical point of view, it is necessary to define some useful expressions. The first one is the Shannon differential entropy definition:

Definition 1 (Shannon differential entropy)**.**

$$\mathrm{h_S}\left(f_{\mathrm{S}}\right) \equiv -\int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \ln f_{\mathrm{S}}(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{2}$$

And the cross entropy term defined as:

Definition 2 (Cross entropy)**.**

$$\mathrm{h_{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \equiv -\int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \ln f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{3}$$

It is also useful to define the Kullback–Leibler divergence [17–19], also called relative entropy. This measure allows one to quantify the departure between the source density function and the function implemented by the learning machine. The measure is:

Definition 3 (Kullback–Leibler Divergence)**.**

$$\begin{aligned} \mathrm{d}_{\mathrm{KL}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) &\equiv \int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})} \mathrm{d}\mathbf{u} && (4) \\ &= -\mathrm{h_S}\left(f_{\mathrm{S}}\right) + \mathrm{h_{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) && (5) \end{aligned}$$

when the corresponding integrals exist.

Interestingly, given that $\mathrm{d}_{\mathrm{KL}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \geq 0$, then:

$$\mathrm{h_S}\left(f_{\mathrm{S}}\right) \leq \mathrm{h_{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \tag{6}$$

Thus, the global minimum value of $\mathrm{h_{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$ is $\mathrm{h_S}\left(f_{\mathrm{S}}\right)$, and it is achieved when $f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}} = f_{\mathrm{S}}$. When this bound is reached, then $\mathrm{d}_{\mathrm{KL}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) = 0$, as well.

Furthermore, from the Kullback–Leibler definition, it is easy to see that:

$$\frac{\partial \mathrm{d}_{\mathrm{KL}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial \theta_k} = \frac{\partial \mathrm{h_{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial \theta_k} \tag{7}$$

with θ_k an element of the vector of parameters $\boldsymbol{\theta}$. Hence, searching for the global minimum of the cross entropy is the same as searching for the global minimum of the Kullback–Leibler divergence.

3.2. The Kullback–Leibler Divergence Is Useful

If a set of i.i.d. random variables is available and the mean of the underlying density function needs to be determined, the weak law of large numbers tells us that the sample average of the random variables converges to the mean. Thus, the average can be used to approximate the mean. This is a classic result that is continuously used in a wide variety of problems.

In the same manner, given a set of i.i.d. random variables, but now needing to find the underlying density function, it is possible to invoke the weak law of large numbers to prove a general pair of bounds that can be used to estimate such a density function. Specifically, it was recently proven [20] that the absolute value of the derivative of the Kullback–Leibler divergence is upper and lower bounded as:

Theorem 1. *Given a positive real number ε, any parameter vector $\boldsymbol{\theta}$ and an i.i.d. sequence with n elements, then the components of $\nabla_{\boldsymbol{\theta}} \mathrm{d}_{\mathrm{KL}}\left(f_{\mathbf{S}}; f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$ comply with:*

$$\frac{\sqrt{\mathrm{I}_{\mathrm{R}}^{\theta_k}} - \frac{1}{n}\sqrt{\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}}}{\sqrt{\mathrm{I}_{\mathrm{L}}}} \leq \left|\frac{\partial \mathrm{d}_{\mathrm{KL}}\left(f_{\mathbf{S}}; f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial \theta_k}\right| \leq \frac{\sqrt{\mathrm{I}_{\mathrm{R}}^{\theta_k}} + \frac{1}{n}\sqrt{\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}}}{\sqrt{\mathrm{I}_{\mathrm{L}}}} \tag{8}$$

where:

$$\mathrm{I}_{\mathrm{R}}^{\theta_k} \equiv \int_{\mathcal{M}_\varepsilon^n} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left|\frac{\partial}{\partial \theta_k}\left\{-\frac{1}{n}\ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u}) - \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)\right\}\right|^2 \mathrm{d}\mathbf{u} \tag{9}$$

$$\mathcal{M}_\varepsilon^n \equiv \left\{\{\mathbf{s}\}_n \in \mathcal{S}^n : \left|-\frac{1}{n}\ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\{\mathbf{s}\}_n) - \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)\right| \leq \varepsilon\right\} \tag{10}$$

$$f_{\{\mathbf{S}\}_n} \equiv f_{\{\mathbf{S}\}_n}(\{\mathbf{s}\}_n) = \prod_{i=1}^{n} f_{\mathbf{S}}(\mathbf{s}_i) \tag{11}$$

$$f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}} \equiv f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\{\mathbf{s}\}_n) = \prod_{i=1}^{n} f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{s}_i) \tag{12}$$

$$\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \equiv \int_{\mathcal{S}^n} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\{\mathbf{S}\}_n}(\mathbf{u})}{f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right)^2 \mathrm{d}\mathbf{u} \tag{13}$$

$$\mathrm{I}_{\mathrm{L}} \equiv \int_{\mathcal{M}_\varepsilon^n} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{14}$$

It is important to note that the term $\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}$ corresponds to the relative information [21]. This term can be re-expressed as:

$$\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} = \int_{\mathcal{S}} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\{\mathbf{S}\}_n}(\mathbf{u})}{f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right)^2 \mathrm{d}\mathbf{u} \tag{15}$$

$$= \int_{\mathcal{S}} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial \ln f_{\{\mathbf{S}\}_n}(\mathbf{u})}{\partial \theta_k} - \frac{\partial \ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}{\partial \theta_k}\right)^2 \mathrm{d}\mathbf{u} \tag{16}$$

$$= \int_{\mathcal{S}} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial \ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}{\partial \theta_k}\right)^2 \mathrm{d}\mathbf{u} \tag{17}$$

$$= \mathrm{j}_{\mathrm{CI}}\left(f_{\{\mathbf{S}\}_n}, f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \tag{18}$$

where the last term is defined as the cross information [21], and it is defined as:

$$\mathrm{j}_{\mathrm{CI}}\left(f_{\{\mathbf{S}\}_n}, f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \equiv \int_{\mathcal{S}} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial \ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}{\partial \theta_k}\right)^2 \mathrm{d}\mathbf{u} \tag{19}$$

From these definitions, it is clear that for the case analyzed in this work, where $f_{\{\mathbf{S}\}_n}$ does not depend on the parameters, then it is true:

$$\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} = \mathrm{j}_{\mathrm{CI}}\left(f_{\{\mathbf{S}\}_n}, f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \geq 0 \tag{20}$$

However, it is not clear what is the global minimum of this value. It is also clear that when $f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}} = f_{\{\mathbf{S}\}_n}$, from a functional point of view, both the relative information and the cross information become equal to the Fisher information of $f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}$. It is not possible to determine what are the values taken by this function when $f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}} \neq f_{\{\mathbf{S}\}_n}$.

The previous definitions allow one to express the bounds in Equation (8) as a function of the cross entropy and cross information as follows:

$$\frac{\sqrt{\mathrm{I}_{\mathrm{R}}^{\theta_k}} - \frac{1}{n}\sqrt{\mathrm{j}_{\mathrm{CI}}\left(f_{\{\mathbf{S}\}n}, f_{\{\mathbf{L}\}n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}}}{\sqrt{\mathrm{I}_{\mathrm{L}}}} \leq \left|\frac{\partial \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial\theta_k}\right| \leq \frac{\sqrt{\mathrm{I}_{\mathrm{R}}^{\theta_k}} + \frac{1}{n}\sqrt{\mathrm{j}_{\mathrm{CI}}\left(f_{\{\mathbf{S}\}n}, f_{\{\mathbf{L}\}n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}}}{\sqrt{\mathrm{I}_{\mathrm{L}}}} \tag{21}$$

3.3. Suitable Estimators

In order to better understand the applications of Theorem 1, the following definition is presented:

Definition 4 (Suitable Estimator)**.** *For some data source $f_{\mathbf{S}}$ and an estimator $f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}$, it is said that the latter is a suitable estimator if:*

$$\lim_{n\to\infty} \mathrm{I}_{\mathrm{R}}^{\theta_k} = 0 \tag{22}$$

$$\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}n}; f_{\{\mathbf{L}\}n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} < \infty \tag{23}$$

The first condition, according to the definition in Equation (9), says that in the limit of the sensibility of the difference between the sample estimation of the cross entropy and the cross entropy to parameter variation has to be zero. In other words, this constraint requires that when the number of samples is large, the sample average estimation of the cross entropy behaves like the real cross entropy. The second condition is naturally complied with by all compact density functions that are smooth. This comprises a broad class of practical density functions, so it is naturally complied with in practical problems.

Thus, any suitable estimator can effectively activate the bounds in Equations (8) and (21), squeezing the absolute value of the derivative of the Kullback–Leibler divergence or the absolute value of the derivative of the cross entropy expression, close to zero. This makes it possible to establish a set of equations whose the global solution minimizes the cross entropy, minimizes the Kullback–Leibler divergence and becomes a close estimation of the sought underlying density function. Interestingly, the converse is also true: non-suitable estimators or small datasets effectively block the possibility of identifying the correct source density function. Hence, with a suitable estimator and enough data, it is valid to search for the density function that explains the data with the following optimization program:

$$\begin{aligned} \boldsymbol{\theta}^{\star} &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \\ \mathcal{L}(\boldsymbol{\theta}) &\equiv \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \end{aligned} \tag{24}$$

3.4. Using Relative Fisher Information to Find a Robust Machine

The previous analysis shows that suitable estimators make it natural to search for machine learning solutions associated with the optimization program expressed by Equation (24). However, the bounds expressed in Equations (8) and (21) also indicate that suitable estimators drive Kullback–Leibler divergence and the cross entropy variations caused by parameter perturbations to small values. Both sets of bounds, i.e., Equations (8) and (21), clearly state that the absolute values of this variations are driven to zero in the case of suitable estimators. This is very interesting, because these variations quantify the robustness of the learning machine to parameter variations. Thus, robustness is defined as follows:

Definition 5 (Learning machine robustness measure)**.** *The robustness of a learning machine $f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}$ to perturbations in some parameter θ_k is measured by the quantity:*

$$\left|\frac{\partial \mathrm{d}_{\mathrm{KL}}\left(f_{\mathbf{S}}; f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial\theta_k}\right| \tag{25}$$

or, equivalently, by:

$$\left| \frac{\partial \mathrm{h}_{\mathrm{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial \theta_k} \right| \tag{26}$$

Thus, the closer these values are to zero for all of the parameters, the more robust a machine is.

The previous definitions use the absolute value of the derivatives, which may be difficult to manipulate mathematically. In order to simplify the ensuing analysis, the absolute value is replaced by a power of two. Hence, it is possible to obtain:

$$\begin{aligned}
\left(\frac{\partial \mathrm{d}_{\mathrm{KL}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)}{\partial \theta_k}\right)^2 &= \left(\frac{\partial}{\partial \theta_k} \int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})} \mathrm{d}\mathbf{u}\right)^2 && (27)\\
&= \left(\int_{\mathcal{S}} \left(\frac{\partial f_{\mathrm{S}}(\mathbf{u})}{\partial \theta_k}\right) \ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})} \mathrm{d}\mathbf{u} + \int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right) \mathrm{d}\mathbf{u}\right)^2 && (28)\\
&= \left(\int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right) \mathrm{d}\mathbf{u}\right)^2 && (29)\\
&= \left(\int_{\mathcal{S}} \sqrt{f_{\mathrm{S}}(\mathbf{u})}\sqrt{f_{\mathrm{S}}(\mathbf{u})} \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right) \mathrm{d}\mathbf{u}\right)^2 && (30)\\
&\leq \left(\int_{\mathcal{S}} \left(\sqrt{f_{\mathrm{S}}(\mathbf{u})}\right)^2 \mathrm{d}\mathbf{u}\right) \cdot \left(\int_{\mathcal{S}} \left(\sqrt{f_{\mathrm{S}}(\mathbf{u})} \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right)\right)^2 \mathrm{d}\mathbf{u}\right) && (31)\\
&= \int_{\mathcal{S}} f_{\mathrm{S}}(\mathbf{u}) \left(\frac{\partial}{\partial \theta_k}\left(\ln \frac{f_{\mathrm{S}}(\mathbf{u})}{f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}\right)\right)^2 \mathrm{d}\mathbf{u} && (32)\\
&= \mathrm{d}_{\mathrm{J(I)}}\left(f_{\mathrm{S}}; f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} && (33)\\
&= \mathrm{j}_{\mathrm{CI}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} && (34)
\end{aligned}$$

Henceforth, minimizing $\mathrm{j}_{\mathrm{CI}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}$ can be used to control the robustness of the Kullback–Leibler divergence to parameter perturbations. This makes it possible to improve the optimization program established in Equation (24) and turn it into the following one:

$$\begin{aligned}
\{\lambda^\star, \boldsymbol{\theta}^\star\} &= \arg\min_{\{\lambda, \boldsymbol{\theta}\}} \mathcal{L}(\lambda, \boldsymbol{\theta}) && (35)\\
\mathcal{L}(\lambda, \boldsymbol{\theta}) &\equiv e^{-\lambda^2}\left(\eta \sum_{k=1}^{m} \mathrm{j}_{\mathrm{CI}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k}\right) + (1 - e^{-\lambda^2})\mathrm{h}_{\mathrm{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)
\end{aligned}$$

where the added term guides the search of solutions towards robust ones, λ is a scalar used to control the relative importance of the terms and η is another scalar that can be used to normalize the sum of cross information terms.

Interestingly, the condition $\frac{\partial \mathcal{L}(\lambda, \boldsymbol{\theta})}{\partial \lambda} = 0$ implies:

$$\eta \sum_{k=1}^{m} \mathrm{j}_{\mathrm{CI}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} - \mathrm{h}_{\mathrm{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) = 0 \tag{36}$$

Thus, when $f_{\mathrm{S}} = f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}$, it is true that:

$$\mathrm{j}_{\mathrm{CI}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} = \mathrm{j}_{\mathrm{F}}\left(f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \tag{37}$$

with $\mathrm{j}_{\mathrm{F}}\left(\cdot\right)_{\theta_k}$ the Fisher information of the corresponding density function, and:

$$\mathrm{h}_{\mathrm{CE}}\left(f_{\mathrm{S}}, f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) = \mathrm{h}_{\mathrm{S}}\left(f_{\mathrm{S}}\right) = \mathrm{h}_{\mathrm{S}}\left(f_{\mathrm{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \tag{38}$$

Hence, when the density functions are equal, the following condition is true:

$$\eta \sum_{k=1}^{m} \mathrm{j_F}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} - \mathrm{h_S}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) = 0 \tag{39}$$

which can be readily transformed into:

$$\eta \mathrm{Tr}\left(\mathrm{FIM}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)\right) - \mathrm{h_S}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) = 0 \tag{40}$$

with $\mathrm{Tr}\left(\mathrm{FIM}\left(\cdot\right)\right)$ the trace of the Fisher information matrix. This condition says that in optimal and robust learning machines, a delicate balance is stricken between their associated entropy, which defines their behavior, and their associated Fisher information values, which relates to their internal structure.

Furthermore, the condition $\frac{\partial \mathcal{L}(\lambda, \boldsymbol{\theta})}{\partial \theta_k} = 0$ implies:

$$\frac{\partial \mathcal{L}(\lambda, \boldsymbol{\theta})}{\partial \theta_k} = e^{-\lambda^2} \frac{\partial}{\partial \theta_k} \left(\eta \sum_{k=1}^{m} \mathrm{j_{CI}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \right) + (1 - e^{-\lambda^2}) \frac{\partial}{\partial \theta_k} \left(\mathrm{h_{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)\right) = 0 \tag{41}$$

Hence:

$$e^{-\lambda^2} \left(\frac{\partial}{\partial \theta_k} \left(\eta \sum_{k=1}^{m} \mathrm{j_{CI}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} - \mathrm{h_{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \right) \right) + \frac{\partial}{\partial \theta_k} \left(\mathrm{h_{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)\right) = 0 \tag{42}$$

which does not seem to be a very useful expression. However, in the case where $f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}} = f_{\mathbf{S}}$, the cross entropy $\mathrm{h_{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$ does reach its global minimum, hence its derivatives are equal to zero. Thus,

$$e^{-\lambda^2} \frac{\partial}{\partial \theta_k} \left(\sum_{k=1}^{m} \mathrm{j_{CI}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \right) = e^{-\lambda^2} \frac{\partial}{\partial \theta_k} \left(\sum_{k=1}^{m} \mathrm{j_F}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\theta_k} \right) = 0 \tag{43}$$

which implies the trivial solution $\lambda = \pm\infty$, or it implies that in this case, the trace of the Fisher information matrix is at least a local minima of the sum of the cross information terms.

3.5. Empirical Measurements

It may seem that establishing the optimization program in Equation (35) is enough. However, it is not very useful given that it is impossible to calculate it: the expressions depend on $f_{\mathbf{S}}$, the very density function that is sought! Hence, there is the need to resort to the weak law of large numbers and to approximate the expressions in that optimization program with the corresponding sample averages to obtain the following approximated optimization program:

$$\begin{aligned} \{\lambda^\star, \boldsymbol{\theta}^\star\} &= \arg\min_{\{\lambda, \boldsymbol{\theta}\}} \mathcal{L}(\lambda, \boldsymbol{\theta}) \\ \mathcal{L}(\lambda, \boldsymbol{\theta}) &\equiv e^{-\lambda^2} \left(\eta \sum_{k=1}^{m} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\partial \ln f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{s}_i)}{\partial \theta_k} \right)^2 \right) + (1 - e^{-\lambda^2}) \left(-\frac{1}{n} \sum_{i=1}^{n} \ln f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{s}_i) \right) \end{aligned} \tag{44}$$

with λ and η defined as before. This is the expression that is proposed in this work in order to search for learning machines that both adjust to certain desired behavior and, at the same time, are robust to parameter variations.

4. A Toy Example

In order to test the previous ideas, a very simple example was set. A source density function based on a one-dimensional Gaussian mixture model (GMM) was defined:

$$f_S(s) = \sum_{m=1}^{N_S} \frac{1}{\sqrt{2\pi}} \frac{\alpha_{S;m}}{\sigma_{S;m}} \exp\left(-\frac{1}{2}\left(\frac{s-\mu_{S;m}}{\sigma_{S;m}}\right)^2\right) \tag{45}$$

with:

$$\sum_{m=1}^{N_S} \alpha_{S;m} = 1. \tag{46}$$

The dataset $\{\mathbf{s}\}_n$ is obtained from this source density function.

A learning machine also based on a one-dimensional GMM is defined:

$$f_{L;\boldsymbol{\psi},\boldsymbol{\theta}}(l) = f_{L;N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(l) = \sum_{k=1}^{N_L} \frac{1}{\sqrt{2\pi}} \frac{\alpha_{L;k}}{\sigma_{L;k}} \exp\left(-\frac{1}{2}\left(\frac{l-\mu_{L;k}}{\sigma_{L;k}}\right)^2\right) \tag{47}$$

with:

$$\boldsymbol{\psi} \equiv [N_L] \tag{48}$$

$$\boldsymbol{\theta} \equiv \begin{bmatrix} \boldsymbol{\alpha}_L \\ \boldsymbol{\mu}_L \\ \boldsymbol{\sigma}_L \end{bmatrix} = \begin{bmatrix} \alpha_{L;1} \\ \vdots \\ \alpha_{L;N_L} \\ \mu_{L;1} \\ \vdots \\ \mu_{L;N_L} \\ \sigma_{L;1} \\ \vdots \\ \sigma_{L;N_L} \end{bmatrix} \tag{49}$$

$$\sum_{k=1}^{N} \alpha_{L;k} = 1 \tag{50}$$

where the vector $\boldsymbol{\theta} \in \mathbb{R}^{3N_L}$ defines the parameters that will be sought through the minimization program defined in Equation (44).

The learning machine uses a GMM architecture because of the expressive power of this family of density functions. However, it is very important to note that both the source and the learning machine are thus described by the same family of density functions. Although this might seem to be an unrealistically easy problem, it is not so for the following reasons:

- All density functions can be approximated by a GMM density function up to any desired error level [22]. Thus, all source density functions can be thought of as GMMs.
- What really sets apart the source density function and the learning machine one is that there is no prior knowledge of the correct number of terms N_S. Without this piece of knowledge, it is perfectly possible to end in cases such as $N_L < N_S$, thus crippling the capacity of the estimator, or $N_S < N_L$, which gives space for over fitting and the production of spurious behavior.
- Taking these considerations into account, the experiments that follow are designed to show what happens in the improbable case where $N_S = N_L$ and in the more probable case where $N_S < N_L$. The $N_S = N_L$ case is a low probability one because it is normally almost impossible to guess this value from the data. Nevertheless, taking advantage of all of the knowledge that is available in this case, the experiment is done to set up a benchmark against which all others can be compared.

Another important reason for choosing a learning machine based on the GMM family is that this set of mathematical functions can implement suitable estimators as stated in Definition 4. This is proven in the following paragraphs with an analysis based on the $\alpha_{L;k}$ parameter, but valid for the other parameters of $\boldsymbol{\theta}$, as well.

In order to determine the validity of the constraints imposed by Theorem 1, it is first required to calculate:

$$\mathrm{I}_{\mathrm{R}}^{\alpha_{L;k}} \equiv \int_{\mathcal{M}_{\varepsilon}^{n}} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left| \frac{\partial}{\partial \alpha_{L;k}} \left\{ -\frac{1}{n} \ln f_{\{\mathbf{L}\}_n;N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(\mathbf{u}) - \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}\right) \right\} \right|^2 \mathrm{d}\mathbf{u} \tag{51}$$

To carry through, note that:

$$\begin{aligned} -\frac{1}{n} \ln f_{\{\mathbf{L}\}_n;N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(\mathbf{u}) &= -\frac{1}{n} \ln \prod_{i=1}^{n} f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u_i) &(52)\\ &= -\frac{1}{n} \sum_{i=1}^{n} \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u_i) &(53) \end{aligned}$$

Thus,

$$\frac{\partial}{\partial \alpha_{L;k}} \left\{ -\frac{1}{n} \ln f_{\{\mathbf{L}\}_n;N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(\mathbf{u}) \right\} = -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u_i)}{\partial \alpha_{L;k}} \tag{54}$$

Continuing the derivation,

$$\mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}\right) \equiv -\int_{\mathcal{S}} f_{\mathbf{S}}(u) \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u) du \tag{55}$$

Hence:

$$\frac{\partial}{\partial \alpha_{L;k}} \left\{ \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}\right) \right\} = -\int_{\mathcal{S}} f_{\mathbf{S}}(u) \frac{\partial \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u)}{\partial \alpha_{L;k}} du \tag{56}$$

Hence, thanks to the weak law of large numbers, from Equation (51), it is clear that:

$$\lim_{n\to\infty} \mathrm{I}_{\mathrm{R}}^{\alpha_{L;k}} = \lim_{n\to\infty} \int_{\mathcal{M}_{\varepsilon}^{n}} f_{\{\mathbf{S}\}_n}(u) \left| \frac{\partial}{\partial \alpha_{L;k}} \left\{ -\frac{1}{n} \ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(u) - \mathrm{h}_{\mathrm{CE}}\left(f_{\mathbf{S}}, f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right) \right\} \right|^2 \mathrm{d}u \tag{57}$$

$$\begin{aligned} &= \lim_{n\to\infty} \int_{\mathcal{M}_{\varepsilon}^{n}} f_{\{\mathbf{S}\}_n}(u) \left| -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u_i)}{\partial \alpha_{L;k}} + \int_{\mathcal{S}} f_{\mathbf{S}}(v) \frac{\partial \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(v)}{\partial \alpha_{L;k}} dv \right|^2 \mathrm{d}u &(58)\\ &= 0 &(59) \end{aligned}$$

Analogous conclusions can be extracted for the other parameters, as well.

From the examination of the relative information term, focusing again on $\alpha_{L;k}$ and without losing any generality in the analysis:

$$\begin{aligned} \mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\alpha_{L;k}} &= \int_{\mathcal{S}^n} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\frac{\partial \ln f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{u})}{\partial \alpha_{L;k}} \right)^2 \mathrm{d}\mathbf{u} &(60)\\ &= \int_{\mathcal{S}^n} f_{\{\mathbf{S}\}_n}(\mathbf{u}) \left(\sum_{i=1}^{n} \frac{\partial \ln f_{\mathbf{L};N_L,\boldsymbol{\alpha}_L,\boldsymbol{\mu}_L,\boldsymbol{\sigma}_L}(u_i)}{\partial \alpha_{L;k}} \right)^2 \mathrm{d}\mathbf{u}. &(61) \end{aligned}$$

It is not clear what value this can take. However, assuming that the density functions involved are well behaved, it is possible to assume that $\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\alpha_{L;k}} < \infty$. Then, it is possible to ensure that:

$$\lim_{n\to\infty} \frac{1}{n} \sqrt{\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\alpha_{L;k}}} = 0 \tag{62}$$

Interestingly, minimizing $\mathrm{d}_{\mathrm{J(I)}}\left(f_{\{\mathbf{S}\}_n}; f_{\{\mathbf{L}\}_n;\boldsymbol{\psi},\boldsymbol{\theta}}\right)_{\alpha_{L;k}}$ not only makes a more robust machine in the sense studied in this work: minimization also accelerates the convergence of the expression described in Equation (62) to zero. This is extremely important in machine learning given that this implies that now a smaller number n is sufficient to reach the vicinity of the limit value. Given that data are often expensive to acquire or to process, the minimization procedure described in this work provides this second benefit, one not foreseen before developing the given approach.

The previous analysis shows that GMMs can be used to build suitable estimators (see Definition 4 and Theorem 1). Thus, it is natural to guide the search of such a learning machine by minimizing the Kullback–Leibler divergence. Given that, as stated before, all density functions can be approximated by GMMs, this conclusion extends to all density functions that can be represented as one-dimensional GMMs.

The following experiments were programmed in Python 2.7.6 with the Theano 0.8.2 libraries [23] in order to have the capacity to produce the derivatives required by the Fisher information quantity. The Python–Theano environment generates symbolic derivatives, which are automatically translated into optimized C code, which are in turn seamlessly integrated into this programming environment. In this way, the generated code can work with arbitrarily complex density functions. Furthermore, this programming environment ensures portability across many hardware platforms. It transparently allows the use of graphic processing units (GPUs) to speed up numerical calculations on such platforms. In order to minimize the objective functions detailed in this work, the sequential least squares quadratic programming (SLSQP) provided by Python–Numpy was used. All of the experiments were run on an Intel Core i5-3320M CPU @ 2.60GHz × 4 CPU with 3.5 GiB of RAM, running the Ubuntu 14.04 LTS environment. Some of these experiments lasted minutes, others hours. For a maximally objective test, the test source density curves were randomly generated.

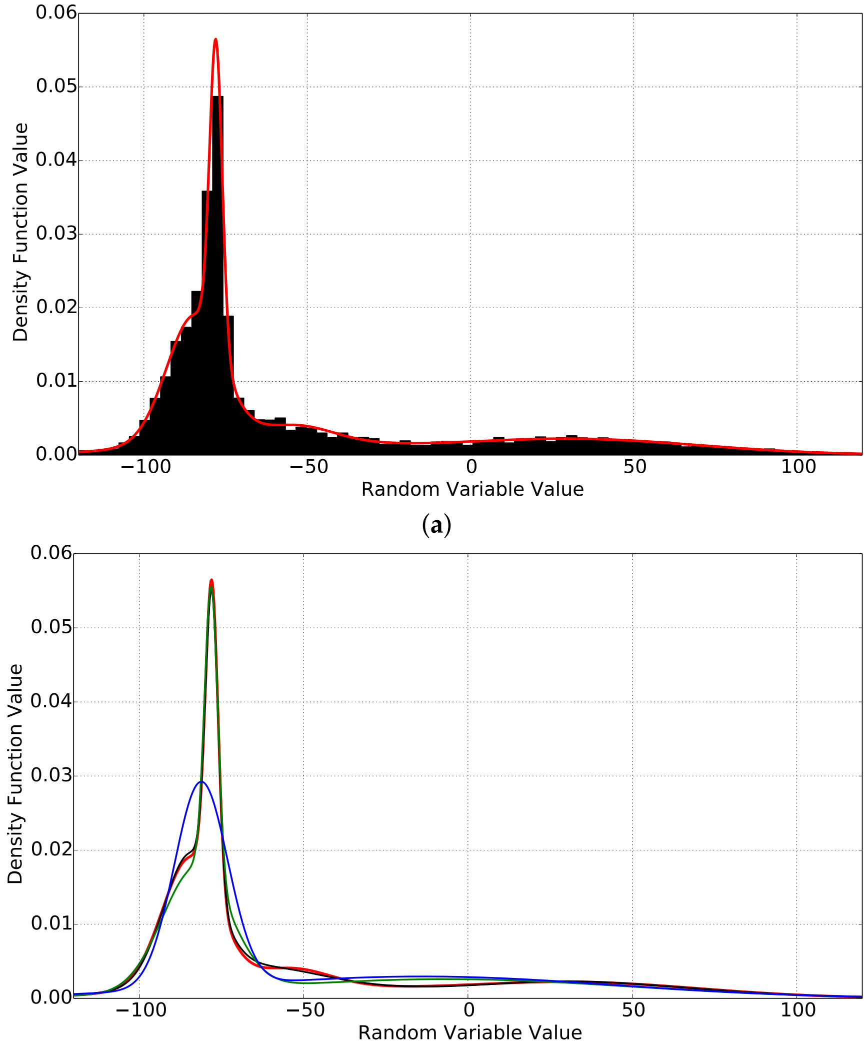
4.1. Optimal Learning Case: Abundant Data, Strong Learning Machine

In this case, $N_S = 5$, $|\{\mathbf{s}\}_n| = 10{,}000$ and $N_L = 8$. With these parameters, it is possible to be sure that the learning machine is completely capable of identifying the source density function. Figure 1, in its upper plot, shows the histogram of the data in black bars and the source density function in red. In its lower plot, the source density function is again in red, the function obtained by the learning machine when $e^{-\lambda^2} = 0.0$ (check Equation (44)) in black, $e^{-\lambda^2} = 0.1$ in green and $e^{-\lambda^2} = 0.5$ in blue. From the lower plot of this figure, it can be seen that the black curve is the closest to the red one, followed by the green and blue learning machine functions. However, the blue one is clearly the farthest from the reference density function. This proves that the higher the influence of the cross information terms, the greater the difference from the desired solution. Thus, it can be said that:

- Enforcing some robustness on the learning machine, using the compound information theoretical objective stated in Equation (44) is at the expense of some precision in the final result.

What can be said about the robustness? Is the green density function ($e^{-\lambda^2} = 0.1$) more robust than the blue one ($e^{-\lambda^2} = 0.5$)? In order to assess this, the Kolmogorov–Smirnov (KS) test was used. This measures the maximum discrepancy between two cumulative density functions, hence showing a worst case number for the difference between two density functions. Figure 2 shows how the KS values change as the parameters of the obtained density functions are perturbed. These KS values compare the reference system's density function with the perturbed ones. The plotted KS values are the average values of 1000 KS measurements from each of these two density functions. The horizontal axis shows the corresponding perturbation values. As an example, a value of 0.5 on the horizontal axis indicates that the parameters were perturbed with a value drawn from a Gaussian density function centered on the original value, but with the standard deviation equal to 0.5 times the absolute value of the original value. On the upper plot of this figure is shown the actual KS value and in the lower plot, the same values divided by that obtained with no perturbation. Dividing the perturbed values by the unperturbed one helps to show how much a learning machine would depart from its expected behavior due to parameter perturbations. Both plots show that:

- Using $e^{-\lambda^2} = 0.1$ effectively produces a slightly more robust machine at the expense of a small departure from the desired solution.
- Using smaller values of $e^{-\lambda^2}$ produces even more robust output curves, but at the price of unacceptably high error in these curves, as indicated by higher KS values.



(**a**)

(**b**)

Figure 1. Reference system, data histogram and learned functions. $N_S = 5$, $|\{\mathbf{s}\}_n| = 10{,}000$ and $N_L = 8$. (**a**) shows in red the density function of the reference system and in black the data histogram; (**b**) shows in red the reference system, and it also shows the results obtained after testing the learning algorithm with values $e^{-\lambda^2} = 0.0$ (black curve), $e^{-\lambda^2} = 0.1$ (green curve) and $e^{-\lambda^2} = 0.5$ (blue curve). The black one is the closest to the red reference, the desired solution. The green one follows closely, but the match is not perfect: the price that is paid for asking for robustness. The blue one is the most different one. As expected, as $e^{-\lambda^2}$ decreased, the influence of the cross information terms increased, and the solutions departed from the desired reference density function.

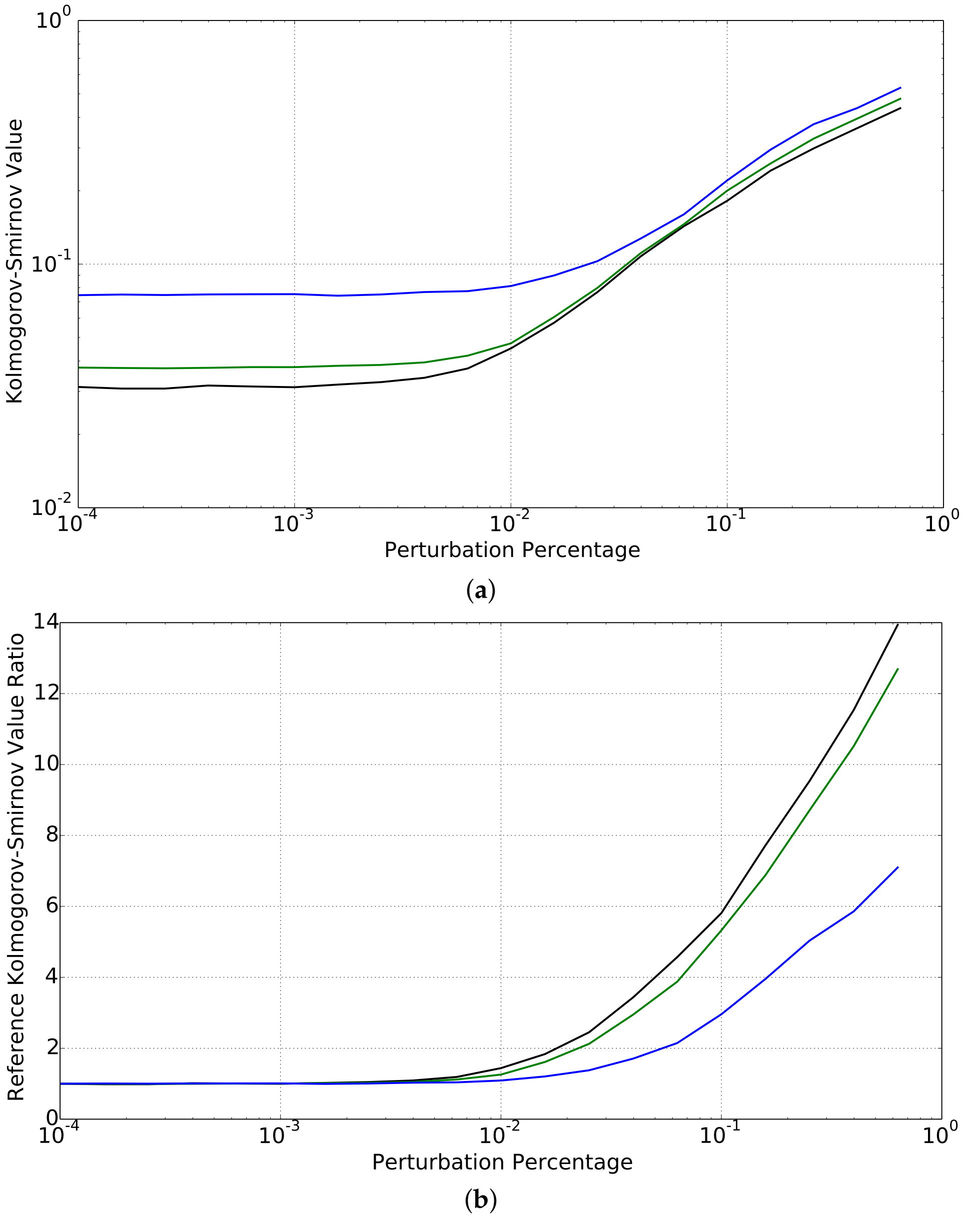


Figure 2. Kolmogorov–Smirnov (KS) values obtained when comparing the obtained learning machines against the reference system, indexed by the amount of perturbation in the parameters. $N_S = 5$, $|\{\mathbf{s}\}_n| = 10{,}000$ and $N_L = 8$. The equivalence between the value of $e^{-\lambda^2}$ and the color code of the curves is the same as in Figure 1. (**a**) shows the KS values indexed by the perturbation percentage; (**b**) shows the same values divided by the corresponding KS value obtained without parameter perturbation. As expected, the black curve has the lowest KS values, but it is the least robust one: it is quickly affected by the ramping perturbations. The green one follows closely and exhibits some robustness. However, it is the blue one, the most different one, that exhibits the highest degree of robustness. All of these curves show the trade-off that exists between obtaining an ideal solution and a robust one. These curves also show that a linear change in the perturbation produces non-linear changes in the learning machines.

4.2. Worst Case Learning Problem: Small Dataset, Strong Learning Machine

Figures 3 and 4 show that the effect of the cross information terms is stronger in learning problems with smaller datasets. Thus, as before, there is a trade-off between the quality of the solution and its robustness to parameter perturbations.

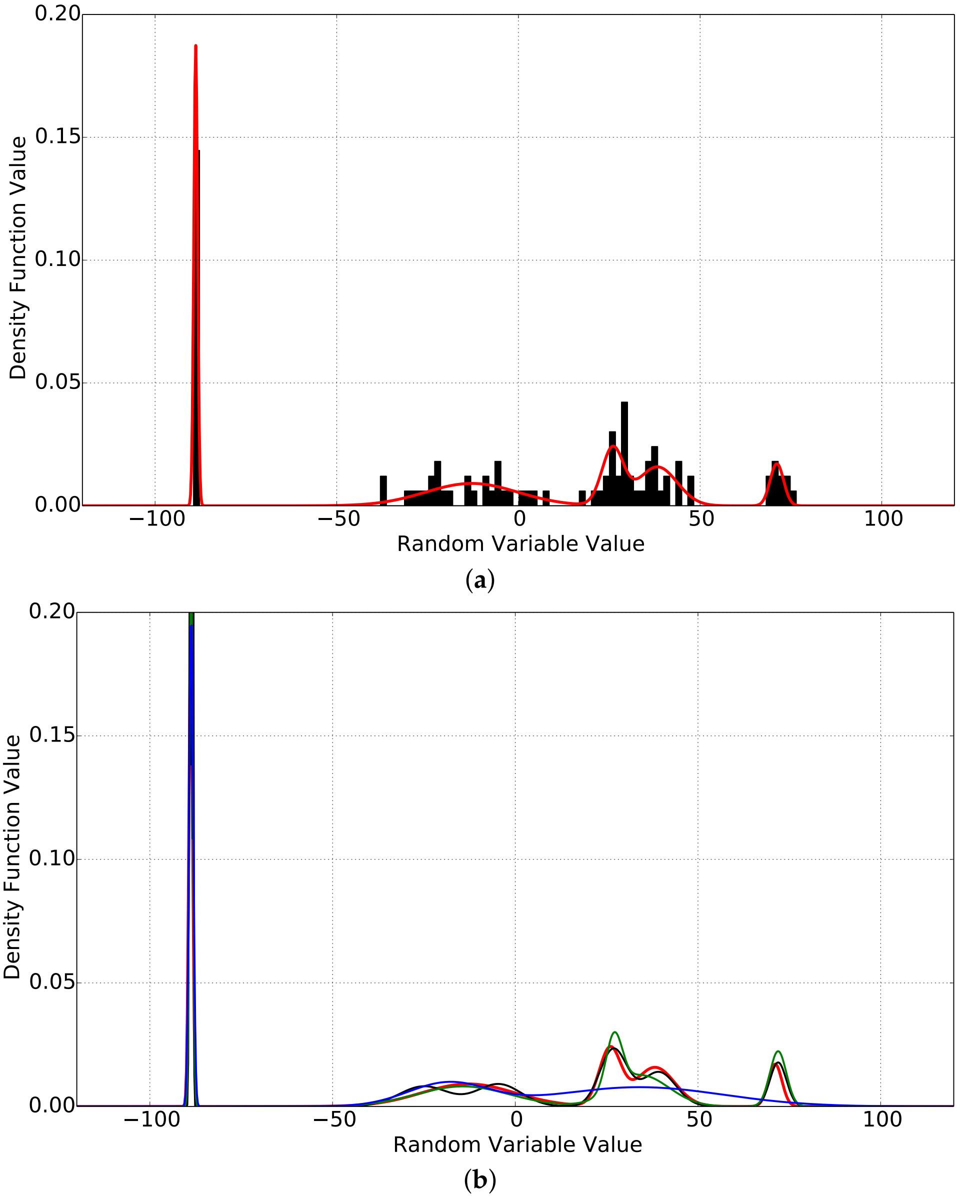
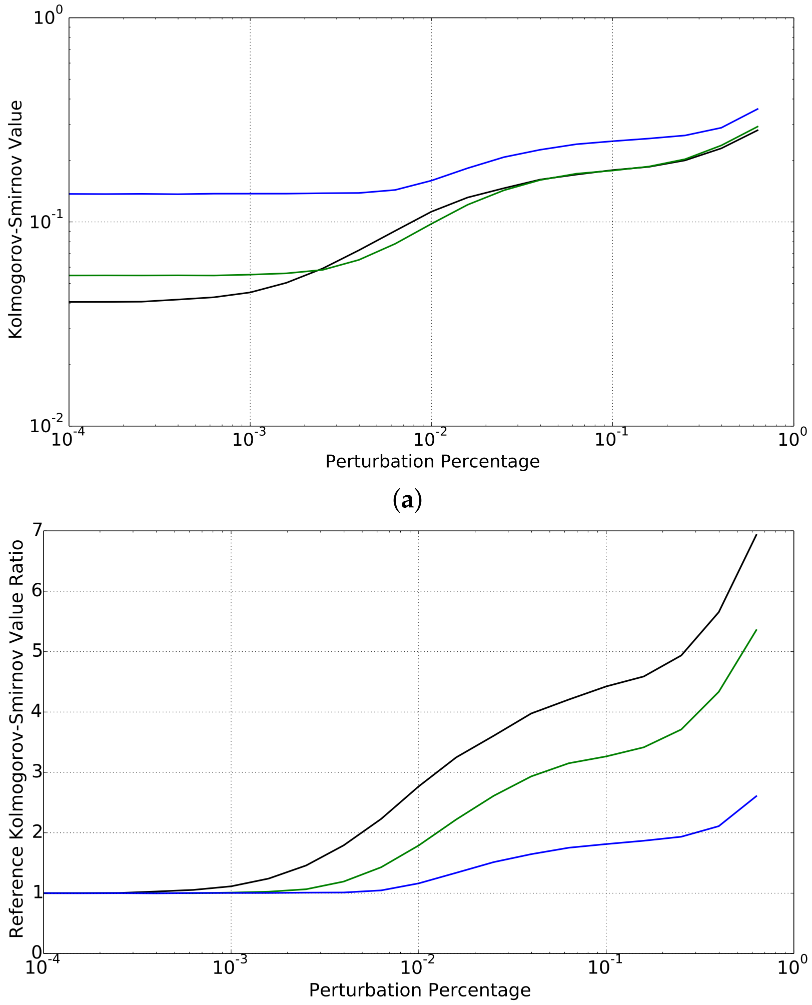


Figure 3. Reference system, data histogram and learned functions. $N_S = 5$, $|\{\mathbf{s}\}_n| = 100$ and $N_L = 8$. (**a**) shows in red the density function of the reference system and in black the data histogram; (**b**) follows the same color convention as the previous figures and shows all of the learning curves. It is clearly appreciated that in this case, i.e., fewer data samples, it is more difficult for the learning machines to find the suitable solutions.



(**a**)

(**b**)

Figure 4. Kolmogorov–Smirnov (KS) values obtained when comparing the obtained learning machines against the reference system, indexed by the amount of perturbation in the parameters. $N_S = 5$, $|\{\mathbf{s}\}_n| = 100$ and $N_L = 8$. The equivalence between the value of $e^{-\lambda^2}$ and the color code of the curves is the same as in Figure 1. (**a**) shows the KS values indexed by the perturbation percentage; (**b**) shows the same values divided by the corresponding KS value obtained without parameter perturbation. It is clear that the black curve ($e^{-\lambda^2} = 0.0$) is the most brittle one in the sense that it is affected by the perturbations the soonest. In this case, the green one ($e^{-\lambda^2} = 0.1$) acts like an intermediate case and the blue one ($e^{-\lambda^2} = 0.5$) is the sturdiest of the group. These plots show that in small datasets the value of lambda has a stronger effect on the trade-off between the quality of the solution and its robustness.

5. An Application in Handwritten Digit Recognition

The objective of this section is to apply the developed framework into a real machine learning problem. A now classical problem is the recognition of handwritten digits from an image database obtained from the American Census Bureau employees and American high schools students (see Figure 5) [24]. In this database, the 28 × 28 pixels images of the digits are transformed into a vector of $784 = 28 \times 28$ dimensions using a lexicographical ordering. Thus, the MNIST machine learning problem consists of designing a learning machine that correctly identifies the digits present in the images. Following the notation used in this work, in this case, the source random variable corresponds to $\mathbf{s} \equiv (\mathbf{y}|\mathbf{x})$, with $\mathbf{y} \in]0,1[^{10}$, where the components of $\mathbf{y}$ comply with $\sum_{k=1}^{10} y_k = 1$ such that they define a probability of belonging to one of the ten digit classes, and $\mathbf{x} \in \mathbb{R}^{784}$ is a vector random variable that represents the image. As an example, an image of a handwritten number five is represent by some vector $\mathbf{x}$, and it has associated a vector $\mathbf{y}$ with $y[6] = 1$ and zeros in the other components. Therefore, the learning problem in this case consists of estimating the density function $f_{\mathbf{y}|\mathbf{x}}$.

Figure 5. A collage of 8 × 8 randomly-picked images of handwritten digits from the MNIST training database [24]. Each image has 28 × 28 pixels, and it is processed by the learning algorithms as a vector of $784 = 28 \times 28$ components, where the transformation is done in a lexicographical manner. The MNIST database has a total of 7000 examples per digit with 60,000 images for training and 10,000 for testing. The testing images are never used in the training process, and they are used for calculating the final performance values.

In this experiment, type of learning machine that uses a multi-class logistic regression was used. Each of the components of the $\mathbf{y}$ vector is then defined by the following expression:

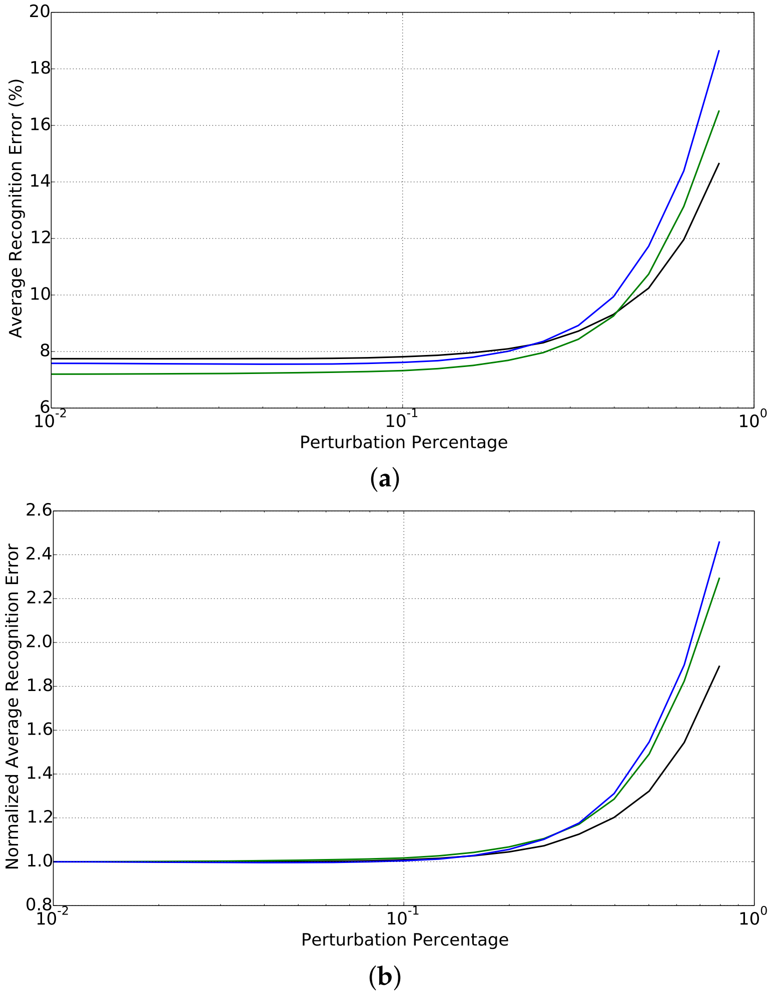
$$y_k = \frac{e^{\mathbf{w}_k \cdot \mathbf{x} + b_k}}{\sum_{l=1}^{10} e^{\mathbf{w}_l \cdot \mathbf{x} + b_l}} \tag{63}$$

for $k \in \{0, 1, \ldots, 9\}$. Thus, continuing with the notation of this work, in this case:

$$\boldsymbol{\psi} \equiv \emptyset \tag{64}$$

$$\boldsymbol{\theta} \equiv \begin{bmatrix} \mathbf{w}_1 \\ b_1 \\ \mathbf{w}_2 \\ b_2 \\ \vdots \\ \mathbf{w}_{10} \\ b_{10} \end{bmatrix} \tag{65}$$

Training was done according to the optimization program stated in Equation (44) after setting the normalization factor $\eta = 1.0$. Training only used the training dataset, and the final performance of the learning machine was determined just using the testing one. In this problem, the performance is measured as the percentage of wrong classifications done by the system. Just to give a reference, the current state of the art in this problem can achieve a testing error of 0.23% [25]. The learning machine used in this experiment was extremely simple, so a much higher error was expected. In this problem, instead of using a conventional optimization routine, as is customary in learning machine problems, the simple gradient descent algorithm was used. This optimization procedure was implemented in Theano and run in the same computer as the last experiment. As before, only the case where $e^{-\lambda^2} \in \{0.0, 0.1, \ldots, 0.5\}$ was considered. Training each of these cases took approximately four hours on the same computer used before. However, the one that only had the cross entropy term was much faster, requiring close to 20 s. Once the training of the eleven learning machines was finished, their errors on the testing set were measured as the parameters were perturbed. These results are presented in Figure 6 and show unexpected results. The learning machines trained with the cross information terms designed to find a robust machine, i.e., the green ($e^{-\lambda^2} = 0.1$) and blue curves ($e^{-\lambda^2} = 0.5$), reached lower error levels than the machine that produced the black curve ($e^{-\lambda^2} = 0.0$) and were not more robust! A comparison between Figures 2, 4 and 6 shows that the learning machine used in the MNIST problem was naturally more robust, i.e., the testing error levels increased only for higher levels of perturbation. Furthermore, in this case, the three learning machines (black, green and blue curves) display rather similar behavior. Furthermore, this plot shows that the unperturbed lowest testing errors of the three machines are 7.74%, 7.20% and 7.58%. In other words, the objective function expressed in Equation (44) also made the system more resilient to changes in the inputs, a very welcome side effect.



(**a**)

(**b**)

Figure 6. Digit recognition error in the MNIST testing set indexed by the amount of perturbation in the parameters. The equivalence between the value of $e^{-\lambda^2}$ and the color code of the curves is the same as in Figure 1. (**a**) shows the KS values indexed by the perturbation percentage; (**b**) shows the same values divided by the corresponding KS value obtained without parameter perturbation. A new behavior is observed now: the solutions with $e^{-\lambda^2} > 0$ were not more robust, but they did achieve a lower testing error! Comparing Figures 2 and 4 with this one shows that the learning machine used in the MNIST problem was sturdier and more difficult to perturb. Furthermore, given that values in this learning machine are calculated with a dot product, the inputs and parameters are duals. Therefore, training for robustness to perturbations in the parameters trickled into the input domain and actually made the machine robust to inputs, as well!

6. Discussion

6.1. Implications for Learning Machines

The objective function described by Equation (44) was very effective in the density function estimation problem with GMMs as learning machines. The learning systems found with this procedure displayed a considerable increase in their robustness to changes in the parameters. However, when applied to the MNIST problem with the very simple logistic regression machine, the procedure did not produce machines with great differences in robustness. All of the trained machines were quite robust compared to those obtained in the density estimation problem. However, the learning machines trained with the procedure detailed in Equation (44) surprisingly did achieve a lower testing error level. Why this unexpected behavior? An explanation might be the following. It is simple to see that the logistic regression approach depends on the dot product $\mathbf{w} \cdot \mathbf{x}$. Thus, introducing perturbations in $\mathbf{w}$ or $\mathbf{x}$ produces similar alterations in the values of the functions. From a mathematical point of view, they are indistinguishable, and they become like duals. Thus, in these types of machines, the proposed training algorithm also prepared the machines for unseen changes in the inputs $\mathbf{x}$. This shows that in machines where inputs and parameters are duals, the robustness induced by the proposed method can also be used to produce machines that generalize better.

Event though very interesting, the proposed method was very slow. Using just the cross entropy term in Equation (44) implied computational times in the order of seconds. However, taking into account the cross information ones was definitely much slower, taking in the order of hours. More work needs to be done in developing faster ways for evaluating the proposed objective functions.

6.2. Connections to Other Frameworks

This delicate balance expressed in Equation (40) with $\eta = 1$ corresponds to the $I - J$ expression formulated in the extreme physical (EPI) principle by Frieden [26]. In all EPI applications, the first term has, in fact, the fixed Fisher form given in Equation (40); and the remaining problem is to know what second term J to use. This generally expresses a form of prior knowledge about the given data scenario (relating to its "physics", such as an equation of continuity). Here, by comparison, the second term is automatically generated, by the learning machine. This is the $\mathrm{h_S}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$ term in (40). This term corresponds to the Shannon differential entropy, and it indicates that the entire principle (40) is based on extremized information measures. Considering that the data used in these learning machine experiments were randomly generated, the $\mathrm{h_S}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$ result seems to be a general one. However, really much more extensive testing has to be done to confirm this strong conclusion. The approach followed in this work shows that for the class of problems studied in this work, it suffices to use $J \equiv \mathrm{h_S}\left(f_{\mathbf{L};\boldsymbol{\psi},\boldsymbol{\theta}}\right)$, i.e., a general intrinsic information term, to explain the underlying laws of an enormous variety of phenomena. In summary, given the generality of the rationale presented in this work, it seems possible to eventually confirm that all processes that can be fitted into the learning machine framework of this work will necessarily obey the EPI principle $I - J$ = minimum, as well. Moreover, the second term J is now automatically constructed from the data rather than being a required input expressing prior knowledge of system physics.

7. Conclusions

In this work, the objective function expressed in Equation (44) specifically designed for training learning machines that exhibit robustness to perturbations in the parameters is presented. The proposed approach is an outgrowth of information-theoretical measures that permits learning machines to exhibit the robustness of output, at various levels, to parameter variation. Computational simulations are used to show the effectiveness of the approach, and associated trade-offs, resulting from the use of the proposed approach.

Acknowledgments: The authors thank CONICYT Chile for the grant FONDECYT 1120680.

Author Contributions: Pablo Zegers and B. Roy Frieden wrote the paper, and designed the experiments. Carlos Alarcón and Alexis Fuentes helped to obtain the mathematical derivations, and checked the mathematical consistency of the statements. Pablo Zegers coded the algorithms and ran the experiments. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: Heidelberg/Berlin, Germany, 1999.
2. Devroye, L.; Lugosi, G. *Combinatorial Methods in Density Estimation*; Springer: Heidelberg/Berlin, Germany, 2001.
3. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992.
4. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
5. Csiszar, I. I-Divergence Geometry of Probability Distributions and Minimization Problems. *Ann. Probab.* **1975**, *3*, 146–158.
6. Csiszar, I. Sanov Property, Generalized I-Projection and A Conditional Limit Theorem. *Ann. Probab.* **1984**, *12*, 768–793.
7. Csiszar, I.; Cover, T.; Choi, B.S. Conditional Limit Theorem under Markov Conditioning. *IEEE Trans. Inf. Theory* **1987**, *33*, 788–801.
8. Da Veiga, S. Global Sensitivity Analysis with Dependence Measures. *J. Stat. Comput. Simul.* **2015**, *85*, 1283–1305.
9. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
10. Shannon, C. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423.
11. Jaynes, E.T. Information Theory and Statistical Mechanics. *Phys. Rev.* **1957**, *106*, 620.
12. Rényi, A. *Probability Theory*; Dover Publications: Mineola, NY, USA, 2007.
13. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. A Learning Algorithm for Boltzmann Machines. *Cognit. Sci.* **1985**, *9*, 147–169.
14. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, USA, 22–24 September 1999.
15. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507.
16. Principe, J. *Information Theoretical Learning*; Springer: Heidelberg/Berlin, Germany, 2010.
17. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
18. Kullback, S. *Information Theory and Statistics*; John Wiley & Sons: New York, NY, USA, 1959.
19. Cover, T.; Thomas, J. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2006.
20. Zegers, P.; Fuentes, A.; Alarcón, C.E. Relative Entropy Derivative Bounds. *Entropy* **2013**, *15*, 2861–2873.
21. Zegers, P. Fisher Information Properties. *Entropy* **2015**, *17*, 4918–4939.
22. Li, J.Q.; Barron, A.R. Mixture density estimation. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2000; Volume 12, pp. 279–285.
23. Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; Bengio, Y. Theano: A CPU and GPU Math Expression Compiler. In Proceedings of the Python for Scientific Computing Conference, Austin, TX, USA, 28 June–3 July 2010.
24. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.

25. Ciresan, D.; Meier, U.; Schmidhuber, J. Multi-column Deep Neural Networks for Image Classification. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
26. Frieden, B.R. *Science from Fisher Information: A Unification*; Cambridge University Press: Cambridge, UK, 1998.