Bayesian interpretation of the Neural Net output and its application to test the agreement between two empirical distributions

LLuís Garrido[1,2], *Vicens Gaitan*[1], *Miquel Serra-Ricart*[3]

(1) Laboratori de Física d'Altes Energies
Universitat Autònoma de Barcelona, E-08193 Bellaterra (Barcelona), Spain.
(2) Departament d'Estrutura i Constituens de la materia
Universitat de Barcelona, Diagonal 647, E-08028 Barcelona, Spain.
(3) Instituto de Astrofísica de Canarias
E-38200 La Laguna (Tenerife), Spain

We will show that a multilayer feed-forward neural net trained from a sample of examples minimizing the quadratic error function are approximations to a Bayesian Decision Rule. This fact is used to introduce a new method to test the agreement between two n-dimensional empirical distributions that is not restricted to work with cumulative distributions in fewer dimensions due to the lack of data, and with the relevant fact that it is free of binning.

Analitical interpretation of the net output

Let us assume that we have two classes of patterns defined by two probability distributions $P_1(\vec{x})$ and $P_2(\vec{x})$, both normalized to 1 ($\int P_1(\vec{x})d\vec{x} = \int P_2(\vec{x})d\vec{x} = 1$). We want to distinguish the two classes using a feed-forward layered neural net trained with back propagation [1] over a sample with an α_i proportion of class i. The input layer will have as many neurons as the dimension of vector $\vec{x}$ and their activation will be $\vec{x}$. The output layer will consist of only one neuron with the desired output of 1 for the first class and 0 for the second one.

The usual error function defined by equation:

$$E = \frac{1}{2}(o - d)^2 \tag{1}$$

where o is the activation of output neuron and d is its desired output, can be written in terms of the probability distributions of the different categories in the training sample

$$E = \int d\vec{x}(\alpha_1 P_1(\vec{x})(o(\vec{x}) - 1)^2 + \alpha_2 P_2(\vec{x})(o(\vec{x}))^2) \tag{2}$$

where $o(\vec{x})$ is the net output and obviously $\alpha_1 + \alpha_2 = 1$.

Assuming no constraint on the functional form of $o(\vec{x})$, the minimum of E can be reached by minimizing the integrand independently for each $\vec{x}$, using $o(\vec{x})$ as a parameter. The result is

$$o(\vec{x}) = \frac{\alpha_1 P_1(\vec{x})}{\alpha_1 P_1(\vec{x}) + \alpha_2 P_2(\vec{x})} \tag{3}$$

Notice that this equation gives the probability for a given $\vec{x}$ to belong to class 1 (Bayes Theorem).

The learning procedure over the net maps the input vector $\vec{x}$ to the output unit by a function $n(\vec{x})$. The network learns by adjusting its weights in order to approximate $n(\vec{x})$ to $o(\vec{x})$. The functional form of $n(\vec{x})$ and its similitude to $o(\vec{x})$ will depend upon the net architecture, the ability to scape from local minima during the training step and the existence of adequate data. This shows that this type of neural net methods are approximations to a Bayesian Decision Rule, but with the important fact that these approximations can be built with a miminum of a priori information based on the generalization power of the nets . This means that is not necessary to know the full information on the $P_i(\vec{x})$.

The generalization for n classes where the test sample has different proportions α_i' from the training sample, is straightforward [4].

A method based on Neural Networks to test the agreement between two empirical distributions

A every day task in all areas of science is the check of theoretical models against experimental data. In some cases the theoretical models can not be checked directly because their explicit analytical form does not exits (example: the models are the result of a large number of algorithms) and/or they are additional effects not included in the model (example: detector effects during the adquisition of the experimental data). A common used strategy to overcome this problem is to generate Monte Carlo events according to the theoretical model and fold them with the additional effects. The result is a simulated data (MC) that can be checked against the experimental data.

The check between the experimental data and MC is normally done by the comparison of the single distributions of events for the most relevant variables. To evaluate the agreement between them usually statistical test like the χ^2, Likelihood or the unbinned Kolmogorov test are done. In cases where the number of variables is low, bidimensional distributions of pairs of variables are also checked in order to see higher order effects.

The best method must be to compare the n-dimensional distributions globally, in which case the standard methods need to bin the variable space. Even for a low number of bins per variable, important amounts of data (not ever avaliable) can be needed, because the number of data points needed to fill the bins with statistical significance grows exponentially with the number of variables.

Here we want to present an additional test free of binning. The method transform a difficult test in the n-dimensional input space to a single test in one dimension, making use of the the known ability of the NN to fit sparse data [3].

The basic idea is to train a layered feed-forward neural net to distinguish the experimental data from MC, minimizing a quadratic error function. The input of such net contains n neurons that are activated by the quantities of the n relevant variables to the problem and the output is only one neuron, which desired activation is 1 for MC and 0 for the experimental data. After the training step, during which a sample

containing a mixture of experimental data and MC is presented to the net, we obtain a function $n(\vec{x}) : R^n \rightarrow R$ that maps the n-dimensional space of the input variables to $[0,1] \in R$.

As it was mention in last section, this function n is an approximation to the probability that the input pattern belong to MC. This function n can now be regarded as a transformation that can be applied to MC and data such that the distributions of the number of events as a function of their "n" value could not be compatible in cases where data is different from MC. Since this point of view the similitude of $n(x)$ to $o(x)$ is not essential but is related to the ability to distinguish the two samples.

More detailed information can be found in reference [5].

Examples and results

In order to show how this Neural method performs in pathological conditions, we build an artificial example with inherent drawbacks for standard work methods. In this example we will try to distinguish between two 10-dimensional distribution with the same cumulative function in one dimension.

In this case the probability distribution function for both samples is:

$$P(\vec{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{det(M)}} e^{-\frac{1}{2}(x-\bar{x})^T M^{-1}(x-\bar{x})} \tag{4}$$

where M is the 10X10 covariance matrix of the x variables. Both samples have the same means and standard deviations, but in the first one there are no correlations between variables and in the second one, the three first variables have a correlation of 20% between them (Fig. 1.a, 1.b).

The neural net used here has 10 input units, 5 units in a single hidden layer, and obviously only one unit as a output. After the training step the NN is able to see the slightly differences between the two distributions as can be see in Fig. 1.c. Both distributions are located around .5 (The number of examples used for each distribution is the same.), but if we look at the histograms for each sample separately, a shift evidence the different correlations.

Another important point is that the network output can be used to understand the differences between the samples. The expected correlation between the input variables x_i and the output of the net is 0. for MC and also for data when MC and experimental data are compatible. Nevertheless, even in this case, due to the approximation introduced by the net, a correlation different from 0. can appear , but has to be the same for MC and data. Only in cases where data and MC are not compatible the variables originating this effect must have different correlations with the output "n" for the two samples. Fig. 1.d shows the correlations between the input variables and the net output for sample 1 versus the same quantities for sample 2. We can see that variables out of the diagonal are the ones originating the difference, as we expect from how they have been originated.

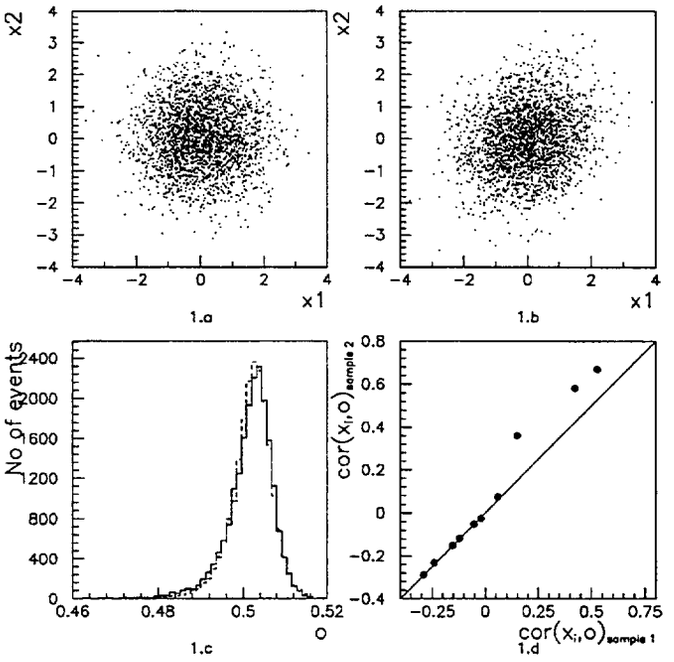

Figure 1:
1.a Bidimensional cumulative distribution (x_1 vs x_2) for the sample with 20% of correlation.
1.b Bidimensional cumulative distribution (x_1 vs x_2) for the sample without correlation.
1.c Output neuron distribution for the two samples (solid and dashed line).
1.d Correlation between the output neuron and each input variables for one sample versus the same quantity for the other in the first example. The three points out of the diagonal corresponds to the variables with correlation.

References

[1] D.E. Rumelhart et al.,"Learning representations by back-propagating errors", Nature 323 (1986) 533.

[2] W.T. Eadie et al, "Statistical Methods in experimental physics". North Holland Publishing Company (1971).

[3] J. Denker et al. "Large automatic learning, rule extraction, and generalization", Complex Systems 1 (1987) 877-922.

[4] Ll. Garrido and V. Gaitan, "Using Neural Nets to measure the τ polarization and its Bayesian interpretation". International Journal of Neural Systems Vol 2. No. 3 (1991).

[5] Ll. Garrido,V. Gaitan,M. Serra-Ricart, "A Method based on Neural Networks to test the agreement between two empirical distributions" UAB-IFAE 92-02.