Estimation of the Hypothesis Real Risk

The simplest method to estimate the objective quality of a learning hypothesis *h* is to divide the examples set in two independent sets: the first, noted $\mathcal{A}$ is used for the training of *h* and the second, noted $\mathcal{T}$, is used to measure its quality. This second set is called *test sample (or examples set)*. One has $\mathcal{S} = \mathcal{A} \cup \mathcal{T}$ and $\mathcal{A} \cap \mathcal{T} = \varnothing$.

As we will see, the measurement of the errors made by *h* on the test set $\mathcal{T}$ is an estimate of the real risk of *h*. This estimate is noted:

$$\hat{R}_{real}(h).$$

Let us examine initially the particular case of the learning of a separating function in the case of *classification rule*.

Let us point out initially the definition of a confusion matrix:

Definition 1.3

The confusion matrix *M(i, j) of a classification rule* h *is a matrix* $\mathcal{C} \times \mathcal{C}$ *of which the generic element gives the number of examples of the test set* $\mathcal{T}$ *of the class i which was classified in the class j.*

In the case of a binary classification, the matrix of confusion is thus of the form:

	‘+’	‘-‘
‘+’	true positives	false positives
‘-‘	false negatives	true positives

If all the errors have the same gravity, the sum of the nondiagonal terms of *M*, divided by the size *t* of the test set, is an estimate $\hat{R}_{real}(h)$ on $\mathcal{T}$ of the real risk of *h*

$$\hat{R}_{real}(h) = \frac{\sum_{i \neq j} M(i, j)}{t}.$$

Nothing t_{err} the number of objects of the test set misclassified, one has:

$$\hat{R}_{real}(h) = \frac{t_{err}}{t}.$$

The empirical confusion matrix is the confusion matrix defined on the training set; for this matrix, the sum of the nondiagonal terms is proportional to the empirical risk, but is not an estimate of the real risk.

The estimation by the confidence interval

Which confidence cans one grant to the estimate $\hat{R}_{real}(h)$? Can it express numerically?

The answer to these two questions is given in a simple way by traditional statistical considerations. If the random samples of training and test are independent then the precision of the estimate depends only on *t* the number of examples of the test set and of $\hat{R}_{real}(h)$.

The sufficient approximation if *t* is rather large (beyond the hundred) is given by *the confidence interval* of $\hat{R}_{real}(h)$:

$$\left[\frac{t_{err}}{t} \pm \zeta(x) \sqrt{\frac{\frac{t_{err}}{t}\left(1 - \frac{t_{err}}{t}\right)}{t}} \right]$$

The function $\zeta(x)$ has in particular the following values:

x	50%	68%	80%	90%	95%	98%	99%
$\zeta(x)$	0.67	1.00	1.28	1.64	1.96	2.33	2.58

The estimate of the real error rate on a test sample $\mathcal{T}$ independent of the training sample $\mathcal{A}$ provides an unbiased estimate of $R_{real}(h)$ with a controllable confidence interval, depending only on the size *t* of the test sample. The larger is this one, the more reduced is the confidence interval and consequently more the empirical error rate gives an indication of the real error rate.

Unfortunately, in the majority of the applications, the number of examples, i.e. the observations for which an expert provided a label, is limited. Generally each new example is expensive to

obtain and hence the training sample and the test sample cannot be increased arbitrarily. There is a conflict between the interest to have the largest possible learning sample $\mathcal{A}$, and the largest possible sample $\mathcal{T}$ to test the result of the training. As it is necessary that the two samples are independent, which is given to one is withdrawn to the other. This is why this method of validation is called the *hold-out method*. It can be applied when the data are abundant. If on the other hand the data are parsimonious, it is necessary to recourse to other methods.

The estimation by cross validation

The idea of the cross validation *(N-fold cross -validation)* consist:

1. To divide the training data $\mathcal{S}$ in *N* subsamples of equal sizes.
2. To retain one of these samples, let by *i*, for the test and to learn on the others $N-1$.
3. To measure the empirical error rate $\hat{R}_{real}(h)$ on the sample *i*.
4. To start again *N* time varying the sample *i* from 1 with *N*.

The final error is given by the average of the measured errors:

$$\hat{R}_{real}(h) = \frac{1}{N}\sum_{i=1}^{N} \hat{R}_{real,i}(h)$$

One can show that this procedure provides an unbiased estimate of the real error rate. It is common to take for *N* values ranging between 5 and 10. In this manner, one can use a great part

of the examples for the training while obtaining a precise measurement of the real error rate. On the other hand, it is necessary to carry out the training procedure of N time.

The question arises however of knowing which learned hypothesis one must finally use. It is indeed probable that each learned hypothesis depends on the sample i used for the training and that one thus obtains N different hypotheses.

Note that if the learned hypotheses are very different the ones from the others (have supposing that one can measure this difference), it should be perhaps there an indication of the inadequacy of $\mathcal{H}$. That indeed seem to show a great variance (in general associated to a great *Vapnik-Chervonenkis* dimension), and thus the training risk has a little importance.

Best is then to remake the training on the total set $\mathcal{S}$. The precision will be good and the estimate of the error rate is known by the N previously training.

- **The estimate by the *leave-one-out* method**

When the available data are poorly, it is possible to push to the extreme the method of cross-validation. In this case, one retains each time one example for the test, and one repeats the training *N* time for all the other training examples.

It should be noted that if one of the interests of the *leave-one-out* validation is to produce less variable hypotheses, one shows, on the other hand, that the estimate of the error rate is often more variable than for cross-validation with a smaller *N*.

This method has the advantage of simplicity and speed. However, when the total number of examples that one lays out is restricted, it can be interesting not to distinguish between the training and test sets, but to use techniques requiring several training passing. In this case, one will lose in computing times but one will gain in smoothness of the estimation relative to the quantity of available data.

- **Some alternatives of the cross validation method: *bootstrap, jackknife***

These techniques differ from the preceding ones in the use of the sampling with replacement over the examples set. The process is as follows: one draws randomly an example, to place it in a set called *bootstrap*[1]. The process is repeated *n* time and the training is then carried out on the bootstrap set. A test is made on the examples nonpresent in this set, computing P_1 a first value of the classifier errors. Another test is made on the complete set of examples, computing P_2. The whole set of operation is repeated *K* time. A certain linear combination of $\overline{P}_1$, the averaged values of P_1 and of $\overline{P}_2$, the average values o P_2 give the value $\hat{R}_{real}(h)$. The theory (Hastie, 2001) proposes the formula:

$$\hat{R}_{real}(h) = 0.636\overline{P}_1 + 0.368\overline{P}_2$$

[1] It is known that the Baron of Munchausen could rise in the airs while drawing on his boots. The omonime, method gives quite astonishing results (here justified theoretically and practically).

based on the fact that the mean of the proportion of the not repeated elements in the test set is equal to 0.368. For small samples, the bootstrap method provides a remarkably precise estimate of $R_{real}(h)$. On the other hand, it asks a great value of K (several hundreds), i.e. a high number of trainings of the classification rule.

There is, finally, another method close but more complex called *jackknife* who aims to reduce the bias of the error rate by plugging-in, when the data are used in the same time for learning and for testing.

We send the interested reader to Ripley (1996) pp.72-73. There are also good references for the problem of the estimate of performance in general.

Algorithms Tuning by a Validation Set

The seeking of the best method for solving a given learning problem implies:

- the choice of the inductive principle;
- the choice of a measurement of the performance, which often implies the choice of a cost function;
- the choice of a training algorithm;
- the choice of the hypotheses space, which depends partly on the choice of the algorithm;
- the tuning of the parameters controlling the algorithm running. Generally the operator tests several methods on the learning problem in order to determine that which seems most suitable with the class of concerned problems. How does it have to proceed?

It is necessary to be wary of an approach that seems natural. One could indeed believe that it is enough to measure for each method the empirical performance using an above described technique. While proceeding of these kind, one arrange to minimize the risk measured on the test sample and hence to tune the method according to this sample. That is dangerous because it may be that, as in the case of the over-fitting phenomenon, working towards this end so much, one moves away from a reduction in the real risk. This is why one envisages, beside the training sample and the test sample, a third sample independent of both others: *the validation sample*, on which one evaluates the real performance of the method. Hence one divides the supervised data $\mathcal{S}$ in three parts: the training set $\mathcal{A}$ the test set $\mathcal{T}$ and the validation set $\mathcal{V}$

The separation of the examples in three sets is also useful to determine at which moment certain training algorithms converge.

The *ROC* curve

Up to now we primarily described evaluation methods of the performances taking into account only one number: the estimate of the real risk. However, in a context of decision-making, it is perhaps useful to be finer in the performance evaluation and to take, in account not only the number of errors, but also the rate of "false positive" and "false negative" (available from the confusion matrix). Often, indeed, the cost of bad classification is not symmetrical and one can rather prefer to have an error rate a little worse if that makes it possible to reduce the type of the most expensive error (for example it is better to operate appendix wrongly (false positive), than of not to detect appendicitis (false negative). The *ROC* curve (*Receiver Operating Characteristic)* allows tuning this compromise[2].

[2] These curves were used for the first time in the Second World War when, one wanted to quantify the radars capacity to discern the interferences of random nature of the signal really indicating the presence of aircraft.

Let us suppose that one characterizes the shapes of data (for example patients) by a size that can result from the combination of examinations (for example the age of the patient, the family antecedents, its blood pressure, etc.). One can then establish a graph for each class giving the probability of belonging to this class according to the data shape (see Figure 1).

In a second stage, one determines for each value of the data size the probability that a positive diagnosis is correct. This probability is given starting from the fraction of the training sample for which the prediction is exact (see Figure 2).

The third stage corresponds to the construction of the ROC curve. For each value of data, one computes the ratio of "true positives" to that of "false positives". If a line is obtained, one must conclude that the test has 50 % of chances to lead to the good diagnosis. The more the curve upwards, the more the test is consistent (the ratio of "true positives" on the "false positive" increases). The consistency is measured by the surface under the curve; it increases with its curvature (see Figure 3)

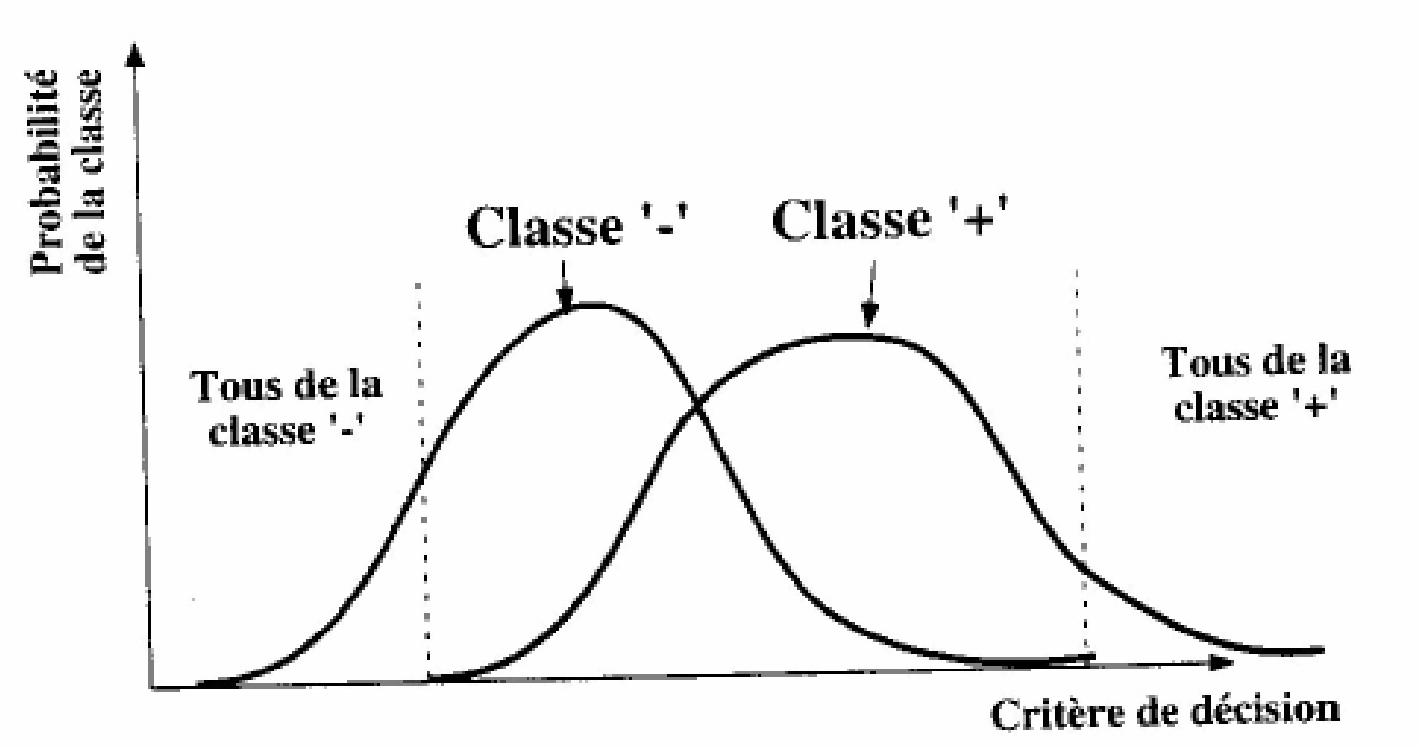

Figure Error! No text of specified style in document. *Curves corresponding to the classes ' + ' and ' - '*

When one found a system of classification judged sufficient good it remains to choose the threshold for a diagnosis 'class +' / 'class -'. The choice of the threshold must provide a high proportion of true positives without involving an unacceptable proportion of false positive. Each point of the curve represents a particular threshold, ranging from the most severe (limiting the number of false positive to the price of many examples of the class ' + ' not diagnosed, i.e., strong proportion of false negative and small proportion of true positive), to the more laxest (increasing the number of true positive at the price of many false positive, see Figure 3). The optimal threshold for a given application depends on factors such as the relative costs of the false positives and false negative, like that of the prevalence of the class ' + '. For example an operator (of telephony or of cabled chain) seeks to detect the churners (in the jargon of the domain, subscribers likely to leave it). These leaving subscribers are very few, but very expensive. One will thus seek to detect the maximum of it in order to try to retain them, even if that means to detect some false churners. One will then use a "laxest" threshold.

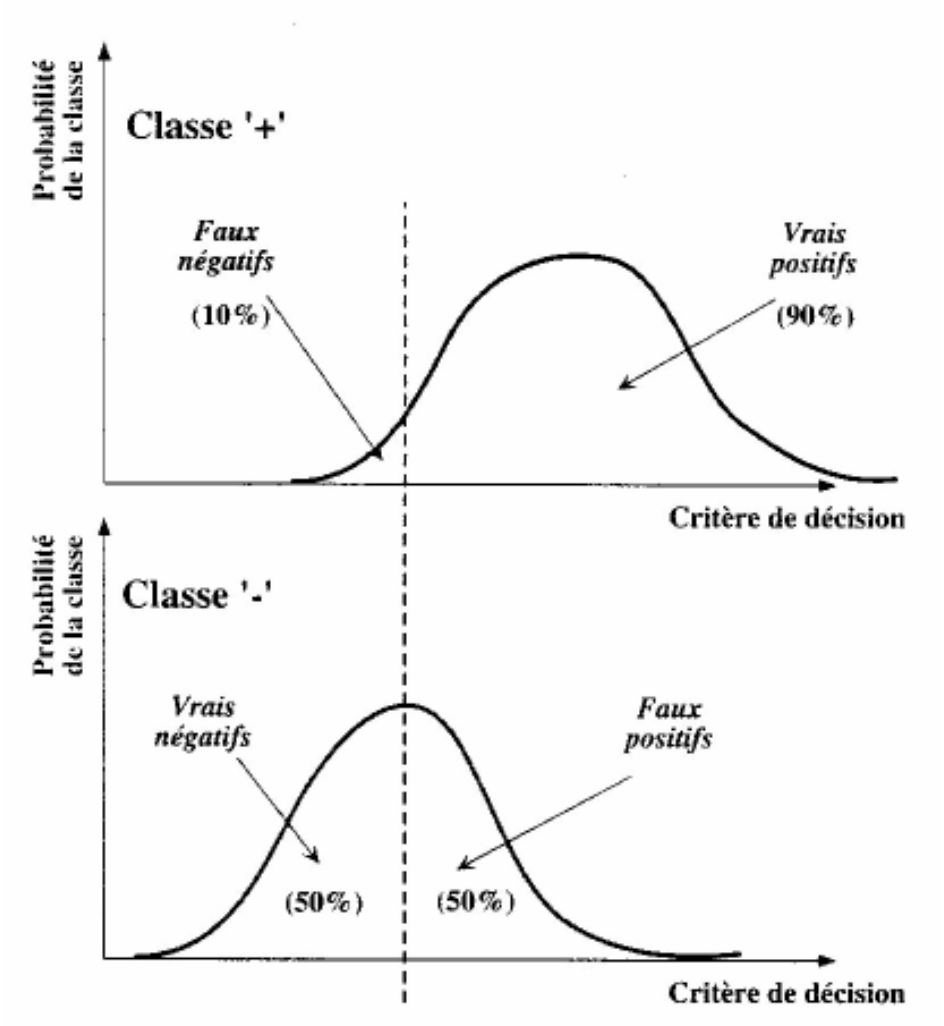

Figure 2 *Threshold deciding for each class the "true positive", "false negative", "false positive" and "true negative ".*

- **Other evaluation criteria**

More of numerical criteria, there is a certain number of qualities that allow distinguishing a hypothesis among others. It can be enumerated:

1. The intelligibility of the training results;
2. The simplicity of the produced hypotheses.

This criterion rise of a traditional rhetoric argument, the *Occam* razor, who affirms that it is wasteful to multiply the useless "entities"[3], in other words that a simple explanation is better than a complicated one.

[3] *Frustra fit per plura, quod fieri potest per pauciora,* classically translated by: It is futile to do with more what can be done with less. Alternatively, *Essentia non sunt multiplicanda praeter necessitatem.* Entities should not be multiplied unnecessarily. Guillaume d' Occam (1288-1348).

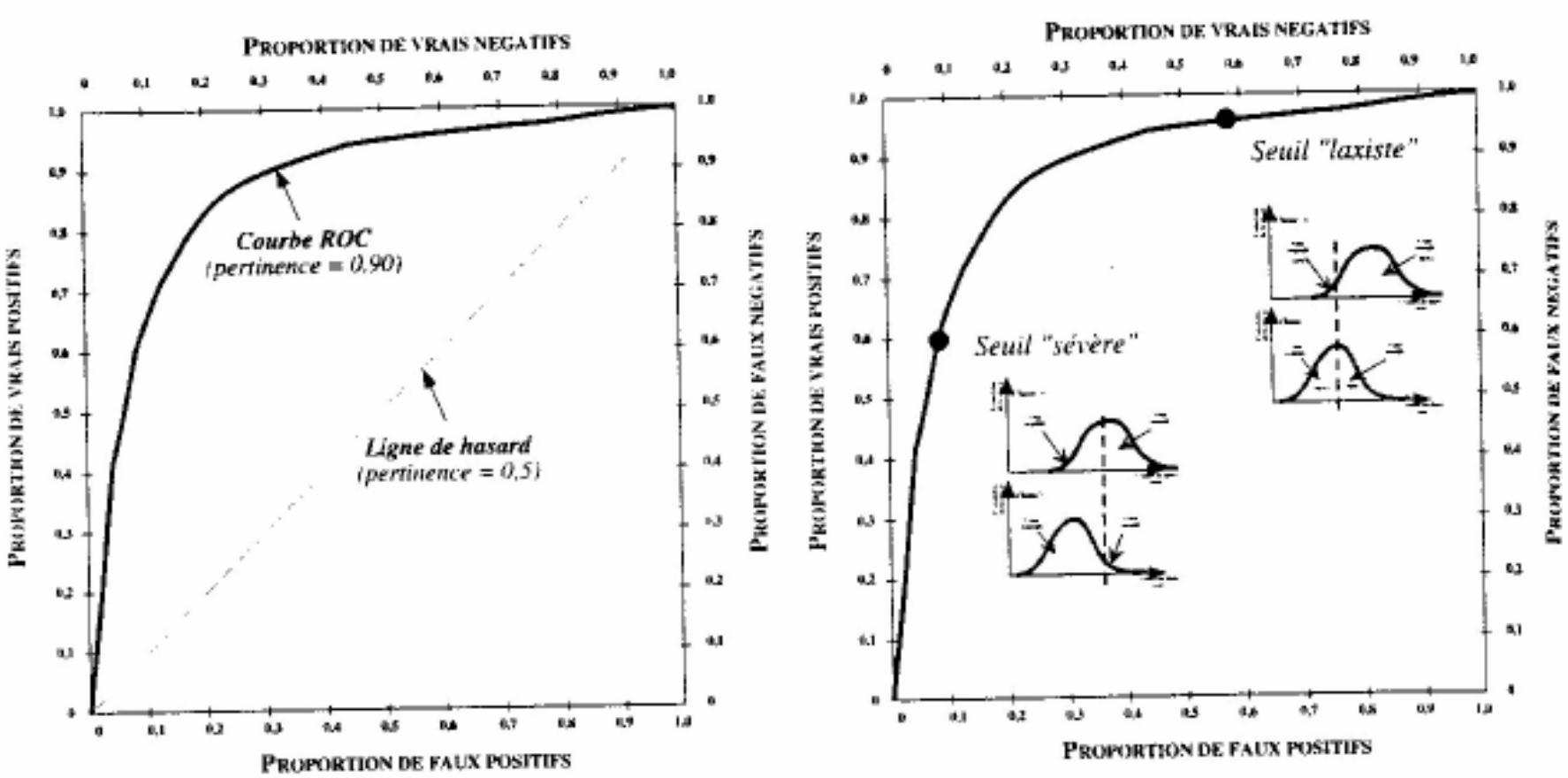

Figure 3 *A ROC curve on the left. Two thresholds on this curve on the right*

Comparison of the Learning Methods

One can always use various algorithms of training for the same task. How to interpret the difference of the performance measured empirically between algorithms? More concretely, an algorithm whose error rate in binary classification is 0.17 is better than another whose measured performance is 0.20?

The answer is not obvious, because the measured performance depends at the same time on the characteristic of the empirical tests carried out and the used tests samples. To decide between two systems knowing only one measurement is thus problematic. The same question arises besides two hypotheses produced by the same algorithm starting from different initial conditions.

A whole of literature pone on this domain. Here we only list the main results.

Comparison of two Hypotheses Produced by the same Algorithm on two Different Test Samples.

Let h_1 and h_2 produced by the same training[4] algorithm and two test sets $\mathcal{T}_1$ and $\mathcal{T}_2$ of size t_1 and t_2. It is supposed that these two test samples are independent: i.e. they are i.i.d. We want to estimate the quantity:

$$\delta_R(h_1, h_2) = R_{real}(h_1) - R_{real}(h_2).$$

If we lay out estimators of these two real risks, it is possible to show that an estimator of their difference is written like the difference of the estimators:

$$\widehat{\delta}_R(h_1, h_2) = \widehat{R}_{real}(h_1) - \widehat{R}_{real}(h_2).$$

Noting with $t_{err,1}$ the data of $\mathcal{T}_1$ misclassified by the hypothesis h_1 and with $t_{err,2}$ the data of $\mathcal{T}_2$ misclassified by the hypothesis h_2 one has:

[4]

$$\widehat{\delta}_R\left(h_1,h_2\right)=\frac{t_{err,1}}{t_1}-\frac{t_{err,2}}{t_2}.$$

The confidence interval of this value is given by the formula

$$\left[\hat{\delta}_R\pm\zeta\left(x\right)\sqrt{\frac{\frac{t_{err,1}}{t_1}\left(1-\frac{t_{err,1}}{t_1}\right)}{t_1}+\frac{\frac{t_{err,2}}{t_2}\left(1-\frac{t_{err,2}}{t_2}\right)}{t_2}}\right]$$

Comparison of two Algorithms on Different Test Sets

We take now a situation that one often meets in the practice: one has training data and one seeks which is the algorithm to be applied to them, among available panoply. For example if one seeks a concept in $\mathbb{R}^d$, thus the learning data are numerical and labeled positive or negative, one can

use a separating function in the form of a hyperplan, or the output of a neural network, or the *k*-nearest neighbors etc. How to choose the best?

Let us suppose to have at our disposal two algorithms A^1 and A^2 and a set of supervised data. The simplest method consists in dividing this set in two subsets $\mathcal{S}$ and $\mathcal{T}$, training A^1 and A^2 on $\mathcal{S}$ (eventually tuning the parameters using a subset $\mathcal{V}$ of $\mathcal{S}$), then compare the performances obtained on $\mathcal{T}$. Two questions arise:

- can one trust this comparison?
- can one make a more precise comparison with the same data?

The answer to the first question is rather negative. The principal reason is that the training sample being the same one for the two methods, its characteristics will be magnified by the comparison. What one seeks is the better algorithm not on $\mathcal{S}$ but on the values of the target functions of which the examples of $\mathcal{S}$ are only random selection.

To answer positively the second question, is necessary to use a technique which browse randomly the training and test data, like the cross validation. An effective algorithm is given below.

Algorithm 1.1 The comparison of two training algorithms

1. Divide the training data $\mathcal{D} = \mathcal{S} \cup \mathcal{T}$ in *K* equal parts. They are noted $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_K$

2. For *i* = 1. *K* makes

$$\mathcal{S}_i \leftarrow \mathcal{D} - \mathcal{T}_i$$

Train the algorithm A^1 on $\mathcal{S}_i$. It provides the hypothesis h_i^1.

Train the algorithm A^2 on $\mathcal{S}_i$. It provides the hypothesis h_i^2.

$\delta_i \leftarrow R_i^1 - R_i^2$ where R_i^1 and R_i^2 are error rates of h_i^1 and h_i^2 on $\mathcal{T}_i$.

end.

3. $\overline{\delta} \leftarrow \frac{1}{K}\sum_{i=1}^{K}\delta_i$.

The confidence interval of $\overline{\delta}$ who is an estimator of the difference in performance between the two algorithms, is then given by the formula:

$$\left[\overline{\delta} \pm \zeta\left(x, K\right)\sqrt{\frac{1}{K\left(K-1\right)}\sum_{i=1}^{K}\left(\delta_i - \overline{\delta}\right)^2}\right].$$

The function $\zeta\left(x, K\right)$ tends towards $\zeta\left(x\right)$ when K increase, but the formula above is valid only if the size of each $\mathcal{T}_i$ is at least of thirty examples. Let us give some values:

x	**90%**	**95%**	**98%**	**99%**
$\zeta(x,2)$	2.92	4.30	6.96	9.92
$\zeta(x,5)$	2.02	2.57	3.36	4.03
$\zeta(x,10)$	1.81	2.23	2.76	3.17
$\zeta(x,30)$	1.70	2.04	2.46	2.75
$\zeta(x,\infty)=\zeta(x)$	1.64	1.96	2.33	2.58

Comparison of two Algorithms on the Same Test Set

If the tests sets on which the two algorithms are evaluated are the same ones, the confidence intervals can be much tighter insofar as one eliminates the variance due to the difference between the test samples.

Dietterich published in 1997 a long paper on the comparison of the training algorithms on the same test set. It examined five statistical tests in order to study the probability of detecting a difference between two algorithms whereas it does not have.

Let h_1 and h_2 be two classification hypotheses. Let us note:

n_{00} = the number of test examples badly classified by h_1 and h_2 and;

n_{01} = the number of test examples badly classified by h_1, but not by h_2;

n_{10} = the number of tests examples badly classified by h_2, but not by h_1;

n_{11} = the number of tests examples correctly classified by h_1 and h_2.

Let be the statistics

$$z = \frac{|n_{01} - n_{10}|}{\sqrt{n_{01} + n_{10}}}$$

The assumption that h_1 and h_2 have the same error rate can be rejected with a probability higher than $x\%$ if $|z| > \zeta(x)$.

The test is known under the name of *paired test* of McNemar or Gillick.

Discussion and Perspective

This chapter introduced the study of various reasonable inductive principles. Those transform a problem of learning into a problem of optimization have providing a criterion that must be optimized by the ideal hypothesis. The majority of the learning methods can then be seen like manners of specifying the hypotheses space to be considered as well as the technique of exploration of this space in order to find there the best hypothesis. This vision of the learning is of a great force. It makes possible to conceive methods of learning, to compare them, and even to build new inductive principles, as those that control automatically the hypotheses space. It is easy to be seduced and to start to reason in the terms of this approach. However, if one reflects about, it will find a surprising framework to the learning approach.

On the one side, there is an indifferent Nature, which distils messages, the data, in a random way, excluding by there the situations of organized or at least benevolent learning. On the other side, there is a solitary learner, completely passive, which awaits the messages, and, in general, does nothing before it have collected them all. One evacuates thus, the continuous collaborative learning, with an evolution of the learner. In the same way the learning in non-stationary environments are excluded, a negative vote for a science, which should above all be a science of dynamics. Moreover, the LM on average it optimizes a mean of the risk, but it really does not seek to identify the target concept. Otherwise it would undoubtedly have interest to devote its resources to the areas of the space in which the target function presents a great dynamics (strong variations) and less in the regions where the things occur quietly. This implies to have a hypotheses space with variable geometry: rich data in the areas of strong dynamics and poor data elsewhere. In addition, the role of *a priori* knowledge, so important in the natural learning, it is here reduced to a very poor expression related only to the choice of the hypotheses space.

Finally, the performances criteria take into account only the mean of error or the risk, and at all criteria of intelligibility or fruitfulness of the produced knowledge. Therefore, one is far from a framework that analyzes the whole diversity of the learning situations. Therefore, this much purified framework appears of a great effectiveness in the analysis of data, which corresponds to a vast field of application.