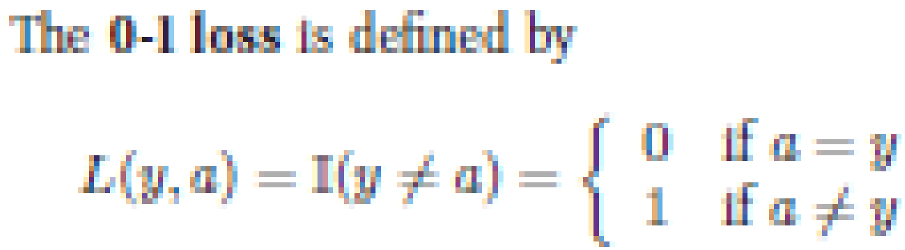
Estimation of the Hypothesis Real Risk

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 A is used for the training of *h* and the second, noted T , is used to measure its quality. This second set is called *test sample (or examples set)*. One has $S = A \cup T$ and $A \cap T = \varnothing$.

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

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

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



The 0-1 loss is defined by

$$L(y,a) = \mathbb{I}(y \neq a) = \begin{cases} 0 & \text{if } a = y \\ 1 & \text{if } a \neq y \end{cases}$$

Definition 1.3

The confusion matrix *M(i, j) of a classification rule* h *is a matrix* $C \times C$ *of which the generic element gives the number of examples of the test set* 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

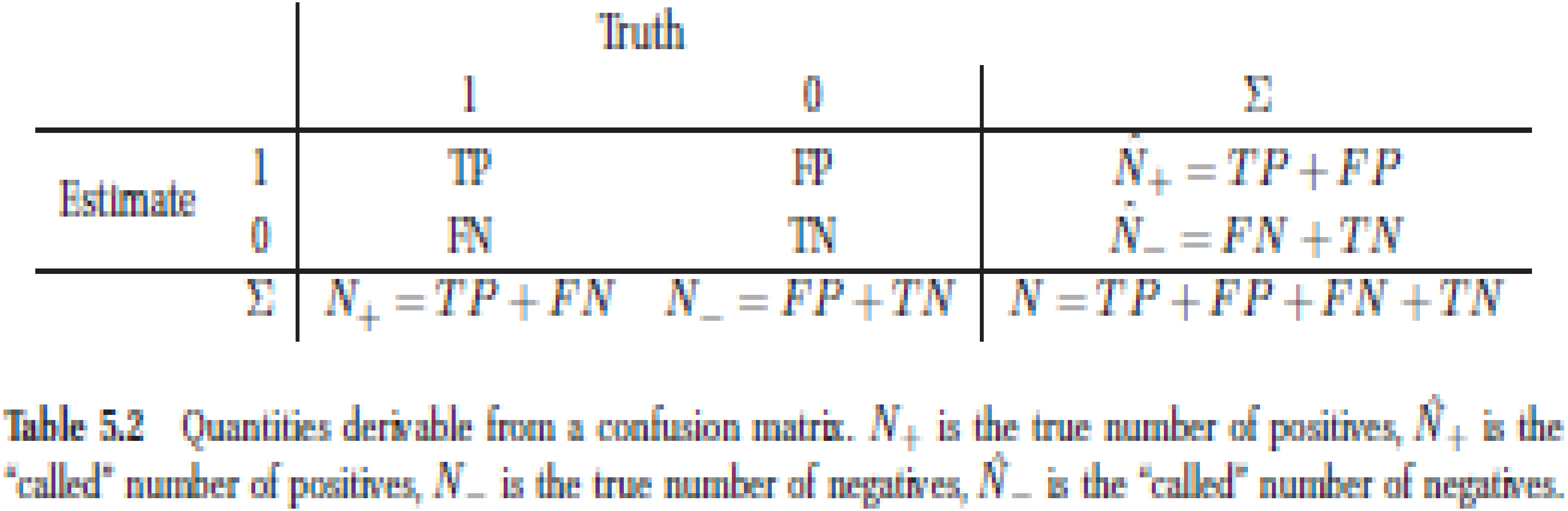
If all the errors have the same gravity, the sum of the nondiagonal terms of *M*, divided by the size *t* of the test set, is an estimate $\hat{R}_{real}\left(h\right)$ on T of the real risk of *h*

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

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

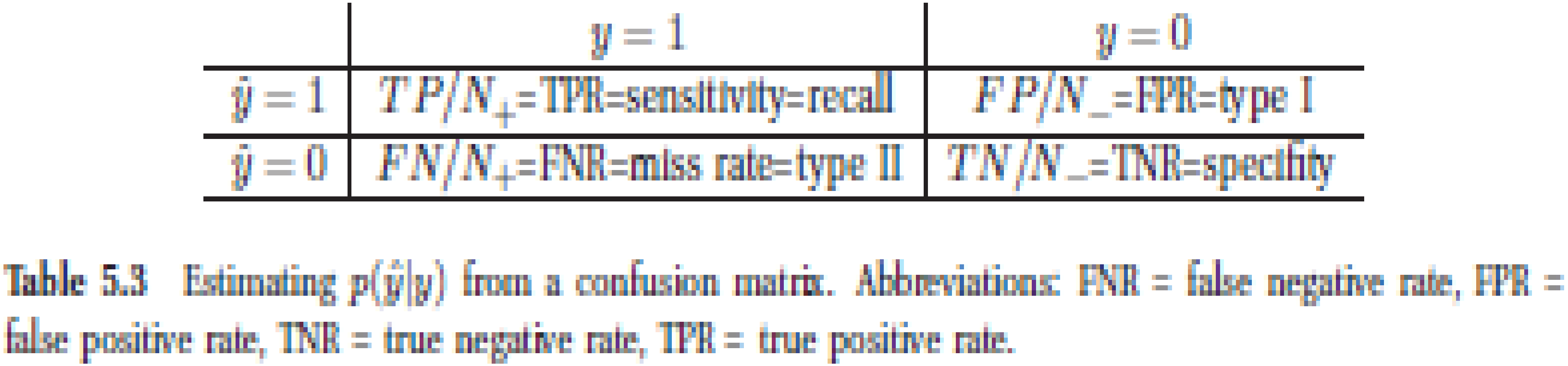
$$\hat{R}_{real}\left(h\right)=\frac{t_{err}}{t}.$$

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.



		Truth		
		1	0	Σ
Estimate	1	TP	FP	$\hat{N}_+ = TP + FP$
	0	FN	TN	$\hat{N}_- = FN + TN$
	Σ	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

Table 5.2 Quantities derivable from a confusion matrix. N_+ is the true number of positives, $\hat{N}_+$ is the "called" number of positives, N_- is the true number of negatives, $\hat{N}_-$ is the "called" number of negatives.



	$y = 1$	$y = 0$
$\hat{y} = 1$	TP/N_+=TPR=sensitivity=recall	FP/N_-=FPR=type I
$\hat{y} = 0$	FN/N_+=FNR=miss rate=type II	TN/N_-=TNR=specifity

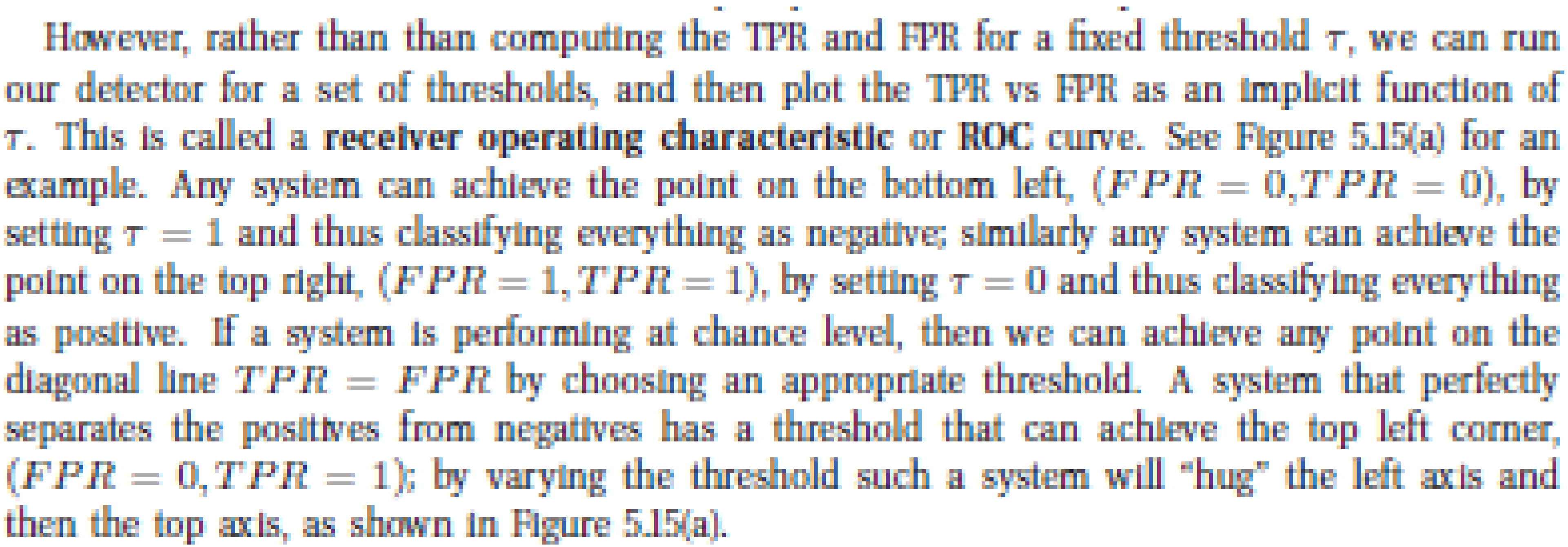
Table 5.3 Estimating $p(\hat{y}|y)$ from a confusion matrix. Abbreviations: FNR = false negative rate, FPR = false positive rate, TNR = true negative rate, TPR = true positive rate.

ROC curves and all that

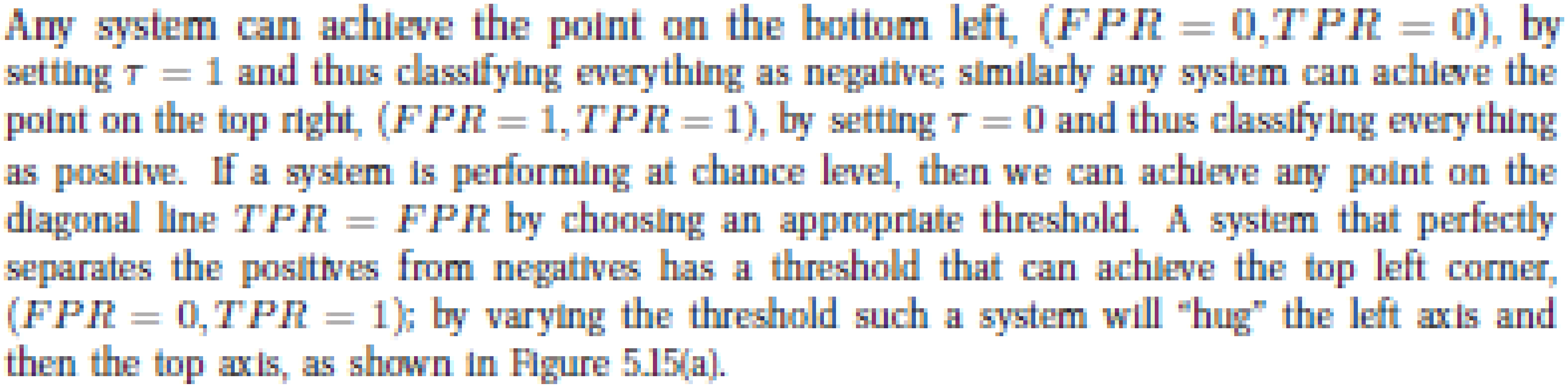
Suppose we are solving a binary decision problem, such as classification, hypothesis testing, object detection, etc. Also, assume we have a labeled data set, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$. Let $\delta(\mathbf{x}) =$

$\mathbb{I}(f(\mathbf{x}) > \tau)$ be our decision rule, where $f(\mathbf{x})$ is a measure of confidence that $y = 1$ (this should be monotonically related to $p(y = 1|\mathbf{x})$, but does not need to be a probability), and τ is some threshold parameter. For each given value of τ, we can apply our decision rule and count the number of true positives, false positives, true negatives, and false negatives that occur, as shown in Table 5.2.

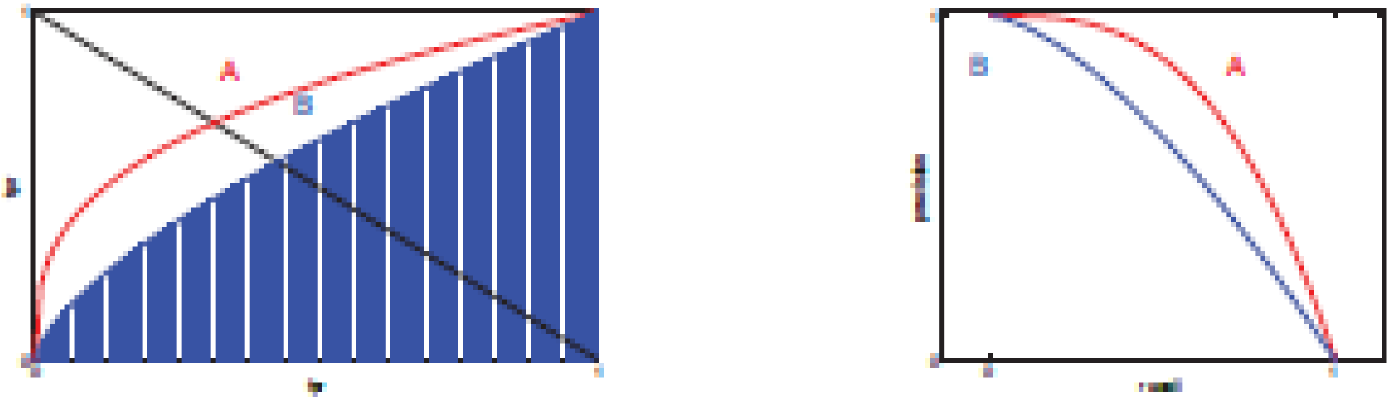
From this table, we can compute the **true positive rate** (TPR), also known as the **sensitivity**, **recall** or **hit rate**, by using $TPR = TP/N_+ \approx p(\hat{y} = 1|y = 1)$. We can also compute the **false positive rate** (FPR), also called the **false alarm rate**, or the **type I error rate**, by using $FPR = FP/N_- \approx p(\hat{y} = 1|y = 0)$. These and other definitions are summarized in Tables 5.3 and 5.4. We can combine these errors in any way we choose to compute a loss function.



However, rather than than computing the TPR and FPR for a fixed threshold τ, we can run our detector for a set of thresholds, and then plot the TPR vs FPR as an implicit function of τ. This is called a **receiver operating characteristic** or ROC curve. See Figure 5.15(a) for an example. Any system can achieve the point on the bottom left, $(FPR = 0, TPR = 0)$, by setting $\tau = 1$ and thus classifying everything as negative; similarly any system can achieve the point on the top right, $(FPR = 1, TPR = 1)$, by setting $\tau = 0$ and thus classifying everything as positive. If a system is performing at chance level, then we can achieve any point on the diagonal line $TPR = FPR$ by choosing an appropriate threshold. A system that perfectly separates the positives from negatives has a threshold that can achieve the top left corner, $(FPR = 0, TPR = 1)$; by varying the threshold such a system will "hug" the left axis and then the top axis, as shown in Figure 5.15(a).



Any system can achieve the point on the bottom left, $(FPR = 0, TPR = 0)$, by setting $\tau = 1$ and thus classifying everything as negative; similarly any system can achieve the point on the top right, $(FPR = 1, TPR = 1)$, by setting $\tau = 0$ and thus classifying everything as positive. If a system is performing at chance level, then we can achieve any point on the diagonal line $TPR = FPR$ by choosing an appropriate threshold. A system that perfectly separates the positives from negatives has a threshold that can achieve the top left corner, $(FPR = 0, TPR = 1)$; by varying the threshold such a system will "hug" the left axis and then the top axis, as shown in Figure 5.15(a).



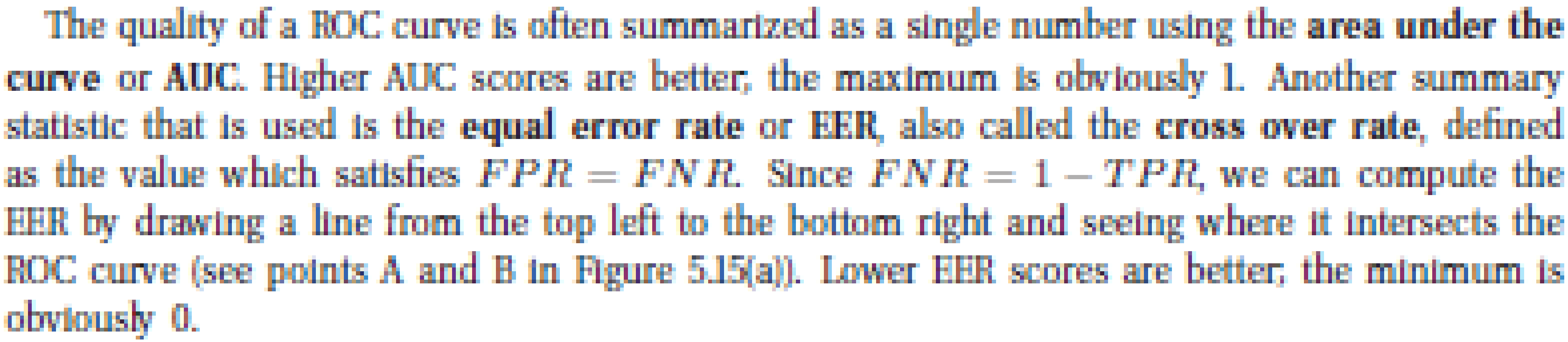
(a)

(b)

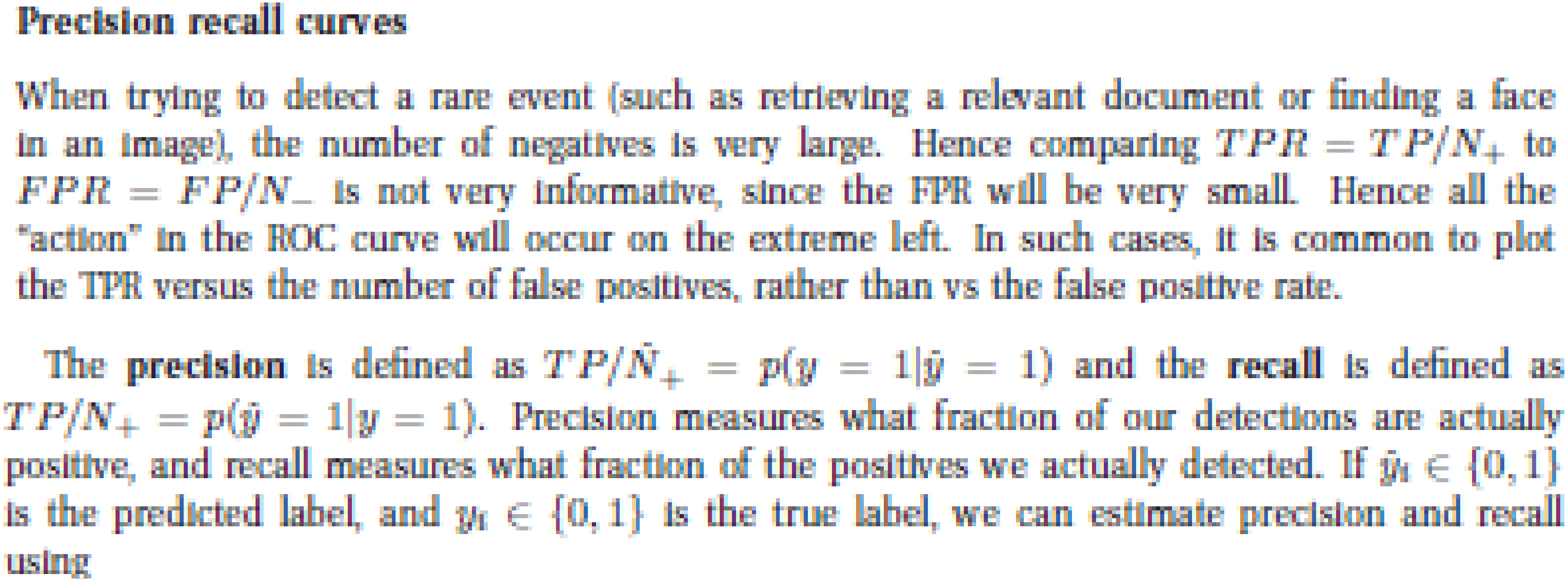
Figure 5.15 (a) ROC curves for two hypothetical classification systems. A is better than B. We plot the true positive rate (TPR) vs the false positive rate (FPR) as we vary the threshold τ. We also indicate the equal error rate (EER) with the red and blue dots, and the area under the curve (AUC) for classifier B. (b) A precision-recall curve for two hypothetical classification systems. A is better than B. Figure generated by PRhand.

	$y=1$	$y=0$
$\hat{y}=1$	$TP/\hat{N}_+$=precision=PPV	$FP/\hat{N}_+$=FDP
$\hat{y}=0$	$FN/\hat{N}_-$	$TN/\hat{N}_-$=NPV

Table 5.4 Estimating $p(y|\hat{y})$ from a confusion matrix. Abbreviations: FDP = false discovery probability, NPV = negative predictive value, PPV = positive predictive value,



The quality of a ROC curve is often summarized as a single number using the **area under the curve** or **AUC**. Higher AUC scores are better; the maximum is obviously 1. Another summary statistic that is used is the **equal error rate** or **EER**, also called the **cross over rate**, defined as the value which satisfies $FPR = FNR$. Since $FNR = 1 - TPR$, we can compute the EER by drawing a line from the top left to the bottom right and seeing where it intersects the ROC curve (see points A and B in Figure 5.15(a)). Lower EER scores are better; the minimum is obviously 0.



Precision recall curves

When trying to detect a rare event (such as retrieving a relevant document or finding a face in an image), the number of negatives is very large. Hence comparing $TPR = TP/N_+$ to $FPR = FP/N_-$ is not very informative, since the FPR will be very small. Hence all the "action" in the ROC curve will occur on the extreme left. In such cases, it is common to plot the TPR versus the number of false positives, rather than vs the false positive rate.

The **precision** is defined as $TP/\hat{N}_+ = p(y = 1|\hat{y} = 1)$ and the **recall** is defined as $TP/N_+ = p(\hat{y} = 1|y = 1)$. Precision measures what fraction of our detections are actually positive, and recall measures what fraction of the positives we actually detected. If $\hat{y}_i \in \{0, 1\}$ is the predicted label, and $y_i \in \{0, 1\}$ is the true label, we can estimate precision and recall using

$$P = \frac{\sum_i y_i \hat{y}_i}{\sum_i \hat{y}_i}, \quad R = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$$

A **precision recall curve** is a plot of precision vs recall as we vary the threshold τ. See Figure 5.15(b). Hugging the top right is the best one can do.

F-scores *

For a fixed threshold, one can compute a single precision and recall value. These are often combined into a single statistic called the **F score**, or **F1 score**, which is the harmonic mean of precision and recall:

$$F_1 \triangleq \frac{2}{1/P + 1/R} = \frac{2PR}{R+P}$$

$$F_1 = \frac{2\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i}$$

To understand why we use the harmonic mean instead of the arithmetic mean, $(P+R)/2$, consider the following scenario. Suppose we recall all entries, so $R = 1$. The precision will be given by the **prevalence**, $p(y = 1)$. Suppose the prevalence is low, say $p(y = 1) = 10^{-4}$. The arithmetic mean of P and R is given by $(P + R)/2 = (10^{-4} + 1)/2 \approx 50\%$. By contrast, the harmonic mean of this strategy is only $\frac{2\times10^{-4}\times1}{1+10^{-4}} \approx 0.2\%$.

In the multi-class case (e.g., for document classification problems), there are two ways to generalize F_1 scores. The first is called **macro-averaged FI**, and is defined as $\sum_{c=1}^{C} F_1(c)/C$, where $F_1(c)$ is the F_1 score obtained on the task of distinguishing class c from all the others. The other is called **micro-averaged FI**, and is defined as the F_1 score where we pool all the counts from each class's contingency table.

False discovery rates *

Suppose we are trying to discover a rare phenomenon using some kind of high throughput measurement device, such as a gene expression micro array, or a radio telescope. We will need to make many binary decisions of the form $p(y_i = 1|\mathcal{D}) > \tau$, where $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ and N may be large. This is called **multiple hypothesis testing**. Note that the difference from standard binary classification is that we are classifying y_i based on all the data, not just based on $\mathbf{x}_i$. So this is a simultaneous classification problem, where we might hope to do better than a series of individual classification problems.

How should we set the threshold τ? A natural approach is to try to minimize the expected number of false positives. In the Bayesian approach, this can be computed as follows:

$$FD(\tau, \mathcal{D}) \triangleq \sum_i \underbrace{(1 - p_i)}_{\text{pr. error}} \underbrace{\mathbb{I}(p_i > \tau)}_{\text{discovery}}$$

where $p_i \triangleq p(y_i = 1|\mathcal{D})$ is your belief that this object exhibits the phenomenon in question. We then define the posterior expected **false discovery rate** as follows:

$$FDR(\tau, \mathcal{D}) \triangleq FD(\tau, \mathcal{D})/N(\tau, \mathcal{D})$$

where $N(\tau, \mathcal{D}) = \sum_i \mathbb{I}(p_i > \tau)$ is the number of discovered items. Given a desired FDR tolerance, say $\alpha = 0.05$, one can then adapt τ to achieve this; this is called the **direct posterior probability approach** to controlling the FDR (Newton et al. 2004; Muller et al. 2004).

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 T independent of the training sample 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 A , and the largest possible sample 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.

The estimation by cross validation

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

1. To divide the training data 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 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 S . The precision will be good and the estimate of the error rate is known by the N previously training.

The one standard error rule

The above procedure estimates the risk, but does not give any measure of uncertainty. A standard frequentist measure of uncertainty of an estimate is the standard error of the mean, defined by

$$se = \frac{\hat{\sigma}}{\sqrt{N}} = \sqrt{\frac{\hat{\sigma}^2}{N}}$$

where $\hat{\sigma}^2$ is an estimate of the variance of the loss:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(L_i - \overline{L})^2, \quad L_i = L(y_i, f_m^{k(i)}(\mathbf{x}_i)) \quad \overline{L} = \frac{1}{N}\sum_{i=1}^{N} L_i$$

Note that σ measures the intrinsic variability of L_i across samples, whereas se measures our uncertainty about the mean $\overline{L}$.

Suppose we apply CV to a set of models and compute the mean and se of their estimated risks. A common heuristic for picking a model from these noisy estimates is to pick the value which corresponds to the simplest model whose risk is no more than one standard error above the risk of the best model; this is called the **one-standard error rule** (Hastie et al. 2001, p216).

- **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 $\hat{R}_{real}\left(h\right)$. The theory (Hastie, 2001) proposes the formula:

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

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

[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).

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 S in three parts: the training set A the test set T and the validation set V

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

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[2] algorithm and two test sets T_1 and 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:

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

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:

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

Noting with $t_{err,1}$ the data of T_1 misclassified by the hypothesis h_1 and with $t_{err,2}$ the data of T_2 misclassified by the hypothesis h_2 one has:

[2]

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

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 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 S and T, training A^1 and A^2 on S (eventually tuning the parameters using a subset V of S), then compare the performances obtained on 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 S but on the values of the target functions of which the examples of 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 $D = S \cup T$ in $K$ equal parts. They are noted $T_1, T_2, ..., T_K$
2.    For $i$ = 1. $K$ makes
              $S_i \leftarrow D - T_i$
      Train the algorithm $A^1$ on $S_i$. It provides the hypothesis $h_i^1$.
      Train the algorithm $A^2$ on $S_i$. 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 $T_i$.
      end.
3.    $\bar{\delta} \leftarrow \frac{1}{K} \sum_{i=1}^{K} \delta_i$.
```

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:

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

The function $\zeta(x,K)$ tends towards $\zeta(x)$ when K increase, but the formula above is valid only if the size of each 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;

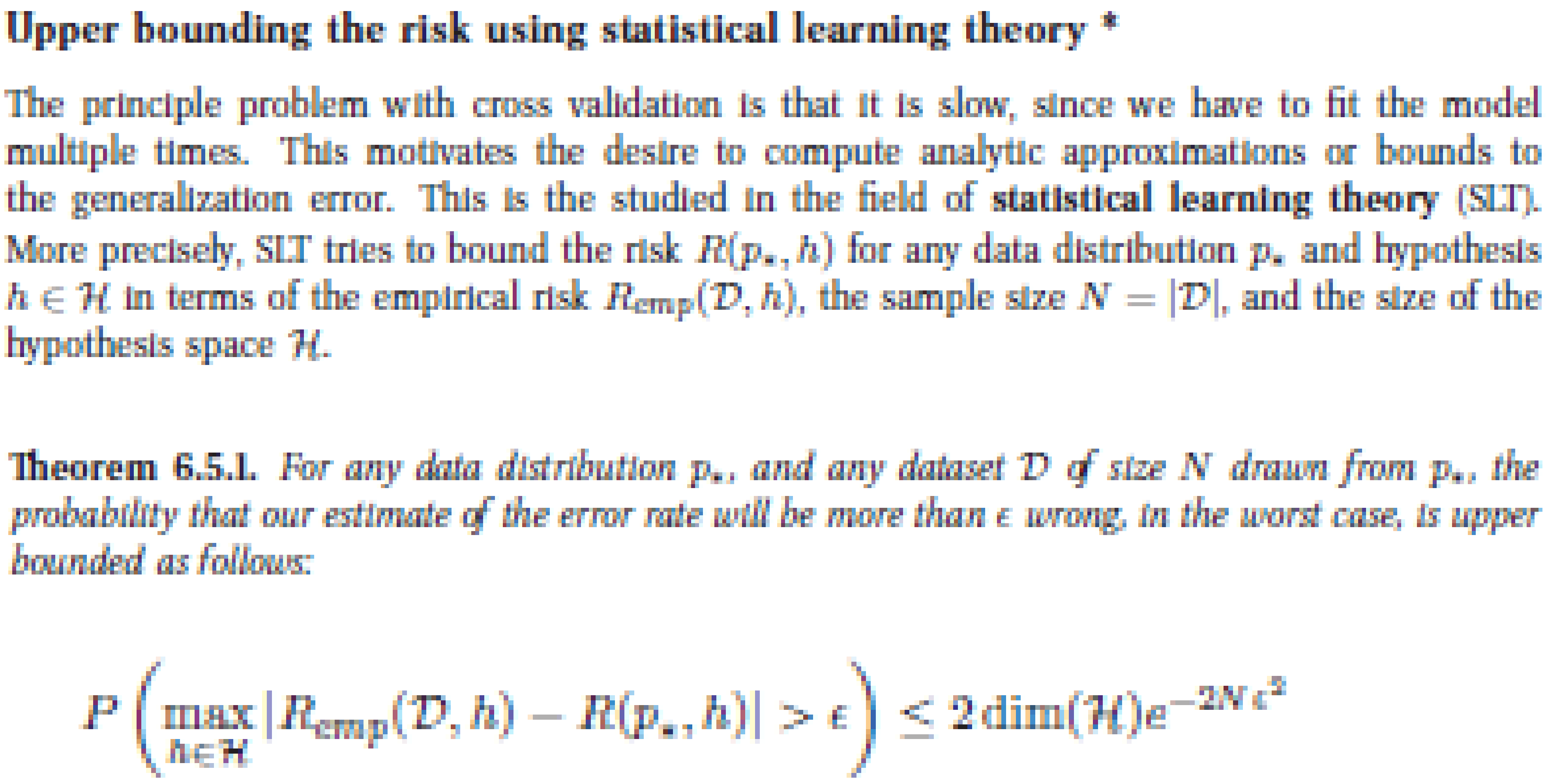
n_{11} = the number of tests examples correctly classified by h_1 and h_2.

Let be the statistics

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

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.



Upper bounding the risk using statistical learning theory *

The principle problem with cross validation is that it is slow, since we have to fit the model multiple times. This motivates the desire to compute analytic approximations or bounds to the generalization error. This is the studied in the field of **statistical learning theory** (SLT). More precisely, SLT tries to bound the risk $R(p_*, h)$ for any data distribution p_* and hypothesis $h \in \mathcal{H}$ in terms of the empirical risk $R_{emp}(\mathcal{D}, h)$, the sample size $N = |\mathcal{D}|$, and the size of the hypothesis space $\mathcal{H}$.

Theorem 6.5.1. *For any data distribution p_*, and any dataset $\mathcal{D}$ of size N drawn from p_*, the probability that our estimate of the error rate will be more than ϵ wrong, in the worst case, is upper bounded as follows:*

$$P\left(\max_{h\in\mathcal{H}} |R_{emp}(\mathcal{D}, h) - R(p_*, h)| > \epsilon\right) \leq 2\,\dim(\mathcal{H})e^{-2N\epsilon^2}$$