A scaling law for the validation-set training-set size ratio

Isabelle Guyon
AT&T Bell Laboratories, Berkeley, California
isabelle@research.att.com

Abstract

We address the problem of determining what fraction of the training set should be reserved as *development test set* or *validation set*. We determine that the ratio of the validation set size over the training set size scales like the square root of two complexity parameters: the complexity of the second level of inference (minimizing the validation error) over the complexity of the first level of inference (minimizing the error rate on the training set).
Keywords: Cross-validation; Learning Theory; Statistics; Machine Learning; Pattern Recognition; Training Set; Validation Set; Test Set; Experiment Design.

Introduction

The problem often arises when organizing benchmarks in pattern recognition to determine what size test set will give statistically significant results.

In a companion paper [1], we tackled the problem from the point of view of the *benchmark organizer*: From a corpus of available data, how much data should be reserved for the benchmark test set?

In this paper, we tackle the problem from the point of view of *benchmark participants*. The benchmark participants do not have access to the benchmark test set until the final test. During the development period, it is common practice that benchmark participants reserve part of the training data to test and compare various systems. Such a subset of the training data is usually called *development test set* or *validation set*. The problem we address here is: *how much data should be reserved for the validation set?* The optimum tradeoff between having more data to train and more data to validate must be found.

Cross-validation is a method of "model selection" which has be widely studied and criticized. Foundational papers include [2, 3, 4] and recent contributions include [5, 6]. In this paper, our emphasis is on exhibiting a simple and general scaling law which can guide experimentalists in pattern recognition: the ratio of the validation set size over the training set size scales like the square root of the complexity of the second level of inference (minimizing the validation error) over the complexity of the first level of inference (minimizing the error rate on the training set).

Our result is easy to remember, it does not require refering a complicated formula or worse to an abacus. It does not contain parameters that are impossible to calculate (we shall explain how to empirically obtain the complexity parameters). We make only a few simplifying hypotheses (large

number of examples, small error rates) and we discuss how the solution is altered if alternative hypotheses are made. Our hypotheses do not include assumptions on the nature of the target function, noise level, nature and structure of hypothesis space and learning algorithm.

The problem of determining the size of the benchmark test set can be solved with classical statistics. In contrast, the problem of determining the size of the validation set involves the complexity of the learning process and the theory of uniform convergence [7]. Our derivation method follows similar lines as found in reference [5]: we bound the probability of error of the recognizer selected by cross-validation using both classical bounds and VC-bounds [7]; we optimize the resulting bound for the training-validation split. Similarly also, we exhibit two "tradeoff terms" the balance of which decides of the optimum. In spite of the similarity of the method, the difference in the set of hypotheses made and in the definition of the "tradeoff terms" yield a different framework to describe the problem and a different solution.

In reference [6], the authors seek the best training/validation split for a specific problem: preventing overtraining of neural networks. They find that the fraction of patterns reserved for the validation set should be inversely proportional to the square root of the number of free adjustable parameters. Our result generalizes and confirms their result.

1 Problem Statement and Notations

We call t the total size of the training database. We call g the fraction of t actually used for training during the development period, referred to as "training set". We call $f = 1 - g$ the remaining fraction of t used as "validation set".

We adopt the learning statement proposed in [7]: Training and test set patterns x_k are drawn randomly and independently from a source of patterns according to a fixed but unknown probability distribution. Patterns are labeled into class categories y_k according to another fixed but unknown probability distribution $P(y_k|x_k)$. By "training" recognizer i, we mean selecting, in a family of recognizers H_i, the recognizer which gives the smallest number of errors on the training set. Each family of recognizer is characterized by its complexity h_i which may or may not be related to the VC-dimension [7], the description length [8], the number of adjustable parameters, or other measures of complexity.

Consider a recognizer i trained with l examples. We call $p_i(l)$ its probability of error on patterns distributed similarly to the training, validation and test patterns. We call $\hat{p}_i(l)$ its empirical error rate calculated on the ft examples of the validation set. We call *opt* the "true" best recognizer, that is the recognizer having the smallest probability of error $p_{opt}(t)$ when the recognizers are trained on all t examples. We call *val* the recognizer selected by cross-validation (see Figure 1).

We adopt the statement of cross-validation proposed [2]: Cross-validation consists in: (i) identifying among N families of recognizers H_i, the family H_{val} whose recognizer *val* has smallest number of errors on the ft examples of the validation set, when all recognizers are trained on the remaining gt examples, and (ii) training a new recognizer with all t examples (training set plus validation set) in the family H_{val}. Step (ii) of our statement is often omitted by other others, but it has important implications in our derivation and it makes sense from the practical standpoint of benchmark participants.

Note that *opt* is fixed once the training data is given, whereas *val* is a function of the training/validation split. Our notation is elliptic since it does not precise $val(f)$. We omit the f for

notation simplicity, but we want to emphasize that if figure 1-a, the dashed curve is the learning curve of $val(1-g_a)$ and in figure 1-b, it is the learning curve of $val(1-g_b)$.

Testing consists of calculating the number of errors on an independent test set, distinct from the training set and the validation set. To lift any remaining ambiguity, we refer to such set as "benchmark test set".

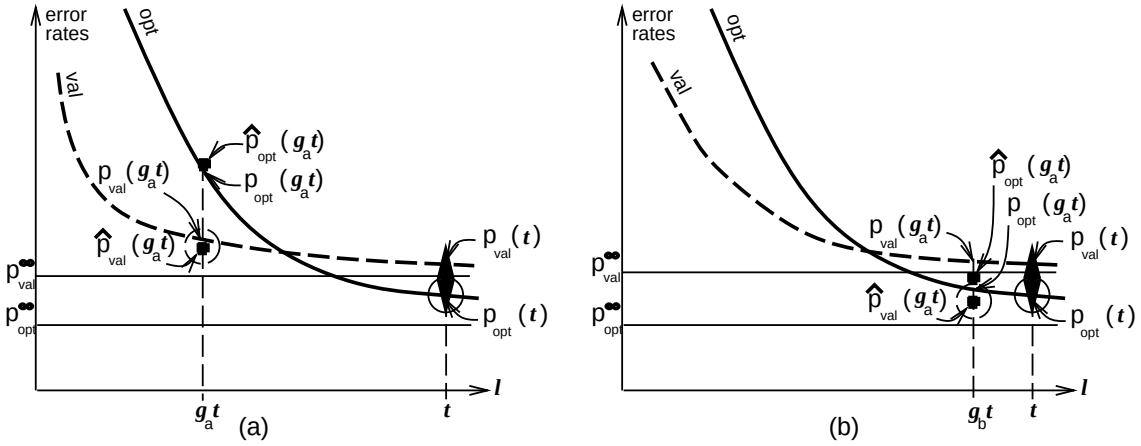

Figure 1. **Learning curves.** Recognizer *val* is the recognizer chosen by cross-validation by using gt training examples and $(1-g)t$ validation examples. It has the smallest error rate on the validation set (point circled by a dashed line). Recognizer *opt* has the smallest probability of error when trained on all t examples (point circled by a full line). We want to avoid the situation $\hat{p}_{val}(gt) < \hat{p}_{opt}(gt)$ and yet $p_{val}(t) > p_{opt}(t)$. This can arise in two limit case: (a) If g is too small, $g = g_a$, the validation set is large enough that the empirical error rates are very close to their mathematical expectations ($\hat{p}_{val}(g_a t) \simeq p_{val}(g_a t)$ and $\hat{p}_{opt}(g_a t) \simeq p_{opt}(g_a t)$), but the learning curves may cross and it is possible that $p_{val}(g_a t) < p_{opt}(g_a t)$ while $p_{val}(t) > p_{opt}(t)$. The "learning curve" term dominates. (2) If g is too large, $g = g_b$, the validation set is too small and $p_{val}(g_b t)$ and $p_{opt}(g_b t)$ are not estimated with enough precision from their empirical values. It is possible that $\hat{p}_{val}(g_b t) < \hat{p}_{opt}(g_b t)$ while $p_{val}(g_b t) > p_{opt}(g_b t)$. The "uncertainty term" dominates.

2 Result and proof sketch

We are going to show that the ratio f^*/g^* of the optimum number of examples reserved for validation over the corresponding number of examples reserved for training is proportional to the square root of the ratio of two "complexity" parameters: the logarithm of the number of families of recognizers considered, $\ln N$, and the largest complexity parameter of all those families, h_{max}. The denominator h_{max} is the complexity of the "first level" of inference, based on selecting the recognizer with smallest training error. The numerator $\ln N$ is the complexity of the "second level" of inference, consisting in choosing among all recognizers trained at the first level the one with smallest validation error.

To reach this result, we shall make a number of simplifying hypotheses, including that the size of the training database t is very large and that the error rate is very small. A discussion of the hypotheses is reported in section 4.

To obtain f^* we want to minimize the difference in performance between the recognizer *val* selected by cross-validation and the recognizer *opt*, that is we want to minimize $p_{val}(t) - p_{opt}(t)$.

We rewrite this difference as the sum of 3 terms:

$$\begin{aligned} & & \textbf{term 1} & & \textbf{term 2} & & \textbf{term 3} \\ p_{val}(t) - p_{opt}(t) & = [p_{val}(t) - p_{val}(gt)] & + & [p_{val}(gt) - p_{opt}(gt)] & + & [p_{opt}(gt) - p_{opt}(t)] \quad . \end{aligned} \tag{1}$$

Similarly as in reference [5], we will find upper bounds for each term; we obtain our prediction for f^* by minimizing the resulting upper bound. The tradeoff arises from the competition of two terms: term 3, the "learning curve" term, and term 2, the "uncertainty" term (see Figure 1 and 2). Term 1 is always negative and will be dropped.

3 Proof

3.1 Learning curve term

First we address the problem of finding bounds for $p_{val}(t) - p_{val}(gt)$ and $p_{opt}(gt) - p_{opt}(t)$. We obtain these differences from "learning curves" which predict the probability of error as a function of the number of training examples (see Figure 1). Several authors [9, 7, 10, 5] have proposed and justified theoretically and experimentally learning curves of the type:

$$p_i(l) = p_i^{\infty} + \left(\frac{h_i}{l}\right)^{\lambda} \quad , \tag{2}$$

where l is the number of training examples and where $0.5 \leq \lambda \leq 1$. The complexity parameter h_i can be determined experimentally by curve fitting [10].

These learning curves are assymptotically valid for large values of the training set size l. They predict the expected value of the error rate for all samples of size l. For small samples, there is some variance for the particular set of patterns that was drawn. We neglect that variance in this analysis by assuming that l is sufficiently large in the region of interest and we assume that the learning curve describes $p_i(l)$ for a particular sequence of patterns. We make the additional simplifying assumption that $\lambda = 1$. This last assumption is valid if the training error rate is vanishingly small ("learnable rule"). See section 4 for a discussion of these hypotheses.

Let us consider the particular recognizer $val(1-g)$ obtained with a training set of gt examples and a validation set of $ft = (1-g)t$ examples. Let us follow the learning curve of that particular recognizer as a function of l. From Equation 2, we have:

$$\textbf{term 1:} \quad p_{val}(t) - p_{val}(gt) = h_{val}\left(\frac{1}{t} - \frac{1}{gt}\right) = \frac{-f}{g}\frac{h_{val}}{t} \leq 0 \quad , \tag{3}$$

Term 1 is always negative or null. Therefore, we bound it by zero and drop it.

From Equation 2, we also have:

$$\textbf{term 3:} \quad p_{opt}(gt) - p_{opt}(t) = h_{opt}\left(\frac{1}{gt} - \frac{1}{t}\right) = \frac{f}{g}\frac{h_{opt}}{t} \quad . \tag{4}$$

Term 3 is a function of h_{opt}. Since recognizer opt is unknown, we bound h_{opt} by h_{max}, the maximum complexity of the family of classifiers considered here:

$$\textbf{term 3:} \quad p_{opt}(gt) - p_{opt}(t) \leq \frac{f}{g}\frac{h_{max}}{t} \quad . \tag{5}$$

3.2 Uncertainty term

Second, we address the problem of finding a bound for $p_{val}(gt) - p_{opt}(gt)$. We can decompose it into:

$$\underset{\textbf{term 2}}{p_{val}(gt) - p_{opt}(gt)} = \underset{\textbf{term 2a}}{[p_{val}(gt) - \hat{p}_{val}(gt)]} + \underset{\textbf{term 2b}}{[\hat{p}_{val}(gt) - \hat{p}_{opt}(gt)]} + \underset{\textbf{term 2c}}{[\hat{p}_{opt}(gt) - p_{opt}(gt)]} \quad , \tag{6}$$

where the hat designates the empirical error rate calculated on the ft examples of the validation set.

By definition of the cross-validation procedure, $\hat{p}_{val}(gt) \leq \hat{p}_{opt}(gt)$ and therefore term 2b is a negative term. Thus we can write:

$$\underset{\textbf{term 2}}{p_{val}(gt) - p_{opt}(gt)} \leq \underset{\textbf{term 2a}}{[p_{val}(gt) - \hat{p}_{val}(gt)]} + \underset{\textbf{term 2c}}{[\hat{p}_{opt}(gt) - p_{opt}(gt)]} \quad . \tag{7}$$

The error rate $\hat{p}_i$ calculated on a test set of size n, for a particular recognizer i, converges to the probability of error p_i, according to the law of large numbers. With probability $(1-\alpha)$:

$$|p_i - \hat{p}_i| \leq \varepsilon(n, \alpha) \quad . \tag{8}$$

Since *opt* is a particular unknown but "fixed" recognizer (not determined from data), inequality (8) applies to *opt*:

$$\textbf{term2c}: \quad \hat{p}_{opt} - p_{opt} \leq \varepsilon(n, \alpha) \quad . \tag{9}$$

Recognizer *val* is determined from the validation set itself and therefore inequality (8) is not directly applicable to it. We shall bound the deviation $|p_{val} - \hat{p}_{val}|$ by the largest deviation $\sup_i |p_i - \hat{p}_i|$.

From inequality (8):

$$Proba\{|p_i - \hat{p}_i| > \varepsilon(n, \alpha)\} < \alpha \quad . \tag{10}$$

Therefore, for N recognizers:

$$Proba\{\sup_i |p_i - \hat{p}_i| > \varepsilon(n, \alpha)\} < \sum_{i=1}^{N} Proba\{|p_i - \hat{p}_i| > \varepsilon(n, \alpha)\} < N\alpha \quad . \tag{11}$$

Substituting α by α/N, we obtain:

$$Proba\{\sup_i |p_i - \hat{p}_i| > \varepsilon(n, \alpha/N)\} < \alpha \quad . \tag{12}$$

Therefore, with probability $(1-\alpha)$:

$$\sup_i |p_i - \hat{p}_i| \leq \varepsilon(n, \alpha/N) \quad , \tag{13}$$

hence:

$$\textbf{term2a}: \quad p_{val} - \hat{p}_{val} \leq \varepsilon(n, \alpha/N) \quad . \tag{14}$$

Note that this derivation is typical of the VC theory [7].

From inequalities (7), (9) and (14), and replacing the number of test examples n by the size of the validation set ft, we obtain:

$$\underset{\textbf{term 2}}{p_{val}(gt) - p_{opt}(gt)} \leq \underset{\textbf{term 2a}}{\varepsilon(ft, \alpha/N)} + \underset{\textbf{term 2c}}{\varepsilon(ft, \alpha)} \quad . \tag{15}$$

Various values of $\varepsilon(n, \alpha)$ have been proposed in the literature. According to Chernoff's bound [11], and using the hypothesis that the error rate $\hat{p}_i$ is small, the following value of $\varepsilon(n, \alpha)$ is valid:

$$\varepsilon(n, \alpha) = C \frac{\ln(1/\alpha)}{n} \tag{16}$$

where C is a small constant, $C \leq 1.5$. For a discussion of this bound, see section 4.

With this value of $\varepsilon(n, \alpha)$, we obtain from Equation (15):

$$\textbf{term2}: \quad p_{val}(gt) - p_{opt}(gt) \leq \frac{C \ln(N/\alpha^2)}{ft} \tag{17}$$

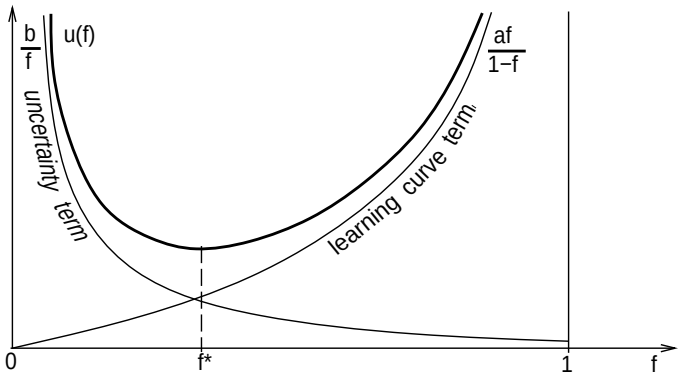

Figure 2. **Validation set size tradeoff.** Two terms are competing: the "learning curve" term tends to increase the training set size (at the expense of the validation set size), whereas the "uncertainty" term tends to increase the validation set size.

3.3 Result

We have bounded all the terms of Equation (1). By using the bounds of term 1 (zero), term 2 (inequality 17) and term 3 (inequality 5), we obtain:

$$p_{val}(t) - p_{opt}(t) \leq \frac{h_{max}}{t} \frac{f}{g} + \frac{C \ln(N/\alpha^2)}{t} \frac{1}{f} \quad . \tag{18}$$

The bound is a function $u(f)$ of the form (Figure 2):

$$u(f) = \underset{\textbf{"learning curve" term}}{a \frac{f}{1-f}} + \underset{\textbf{"uncertainty" term}}{\frac{b}{f}} \quad , \tag{19}$$

N:	Number of families of recognizers considered (e.g. $N = 10$).	
α:	Risk of being wrong (e.g. $\alpha = 0.05$).	
C:	Constant of Chernoff bound (e.g. $C = 1.5$).	
h_{max}:	Largest complexity of the families of recognizers considered (e.g. $h_{max} = 100$).	(a)
t:	Total size of the training database (e.g. $t = 100,000$ examples).	
f^*:	Fraction of t that should be reserved for validation.	
g^*:	Fraction of t that should be reserved for training.	

$$\boxed{\frac{f^*}{g^*} = \sqrt{\frac{C \ln(N/\alpha^2)}{h_{max}}}} \quad \text{(b)}$$

Table 1. **Summary of the steps taken to determine the validation set size:** (a) Notations and typical values of the parameters. (b) Ratio of the number validation examples over the number of training examples that minimizes a bound on difference $p_{val} - p_{opt}$.

$$\text{where} a = \frac{h_{max}}{t} \tag{20}$$

$$b = \frac{C \ln(N/\alpha^2)}{t} \tag{21}$$

$$\tag{22}$$

If the proposed bound is tight, the minimum of the curve $u(f)$ informs us on the optimum training-validation split. The derivative of $u(f)$ with respect to f is:

$$u'(f) = \frac{a}{(1-f)^2} - \frac{b}{f^2} \quad . \tag{23}$$

The derivative is null for particular values f^* and g^* of f and g such that:

$$\frac{f^*}{g^*} = \sqrt{\frac{b}{a}} \quad . \tag{24}$$

By replacing a and b by their values, we obtain the result of Table 1.

Alternatively:

$$f^* = \frac{A}{1+A} \quad , \tag{25}$$

where $A = f^*/g^*$ given by Table 1. Note that fo $A \ll 1$, $f^* \simeq A$.

Numerical application

Let us consider a few typical values of the parameters: 95% confidence level ($\alpha = 0.05$) and $C = 1.5$.

In Table 2, we give the values of f^* obtained for various values of h_{max} and the number N of recognizers involved in the cross-validation procedure.

h_{max} N	1	5	10	50	100
1	0.75	0.77	0.78	0.79	0.80
5	0.57	0.60	0.61	0.63	0.64
10	0.49	0.52	0.53	0.55	0.56
50	0.30	0.32	0.33	0.35	0.36
100	0.23	0.25	0.26	0.28	0.29
500	0.12	0.13	0.14	0.15	0.15
1000	0.09	0.10	0.10	0.11	0.11
5000	0.04	0.05	0.05	0.05	0.05

Table 2. **Values of** f^* when h_{max} and N vary, for $C = 1.5$ and $\alpha = 0.05$.

We notice that in the degenerate case $N = 1$, our method does not predict $f^* = 0$, which would be the logical answer. In this case $p_{val}(t) - p_{opt}(t) \equiv 0$ for all f, because there is only a single recognizer. Any value of f^* works. If the term $-\ln \alpha^2$ is ignored, we satisfy $f^* = 0$ when $N = 1$. Doing so does not dramatically change the other values and yet simplifies further the formula.

The value of h_{max} may be estimated from previous experiments, by fitting empirical learning curves with Equation (2), according to the method proposed in Reference [10]. $h_{max} \simeq F/3$, where F is the number of free parameters, is a rule of thumb for neural networks trained with "back-propagation" using "early stopping".

4 Discussion of the hypotheses

4.1 Learning curve term

Sample average and confidence interval.

Throughout the paper, we have considered only one particular split into training and validation set for each value of f. The Equality in Equation (2), which implies that $p_i(l)$ represents an average over all samples of size l, should be replaced by and inequality involving an error bar (or confidence interval).

The VC-theory [7] provides us with such confidence intervals. With probability $(1 - \eta)$,

$$|p_i(l) - \tilde{p}_i(l)| < 2\varepsilon(l, h_i, \eta) \quad , \tag{26}$$

where $\tilde{p}_i(l)$ is the training error rate calculated on l examples and h_i is the VC dimension, a particular measure of complexity of the family of recognizers H_i. A complete formula for $\varepsilon(l, h_i, \eta)$ is given in Reference [7], page 157. Let us call $\varepsilon_0(l, h_i, \eta)$ the quantity:

$$\varepsilon_0(l, h_i, \eta) = \frac{h_i[\ln(2l/h_i) + 1] - \ln(\eta/10)}{l} \quad . \tag{27}$$

For small values of the training error, $\tilde{p}_i(l)$, $\varepsilon \simeq \varepsilon_0$ whereas, for large values of $\tilde{p}_i(l)$, $\varepsilon \simeq \sqrt{\varepsilon_0}$. For the purpose of making qualitative statements, the behaviour of the bound in between these two linit case can be approximated by a power law: $\varepsilon \simeq (\varepsilon_0)^\lambda$, with $0.5 \leq \lambda \leq 1$. The learning curves

of Equation (2) can be connected to the VC-bound by making the following simplifications:
(i) For large values of l, $p_i(l)$ and $\tilde{p}_i(l)$ reach the assymptote p_i^∞ symmetrically. The bound is split into: $p_i^\infty - \tilde{p}_i(l) < \varepsilon(l, h_i, \eta)$ and $p_i(l) - p_i^\infty < \varepsilon(l, h_i, \eta)$. This last bound is the learning curve of interest to us.
(ii) $\ln(2l/h_i) + 1 \simeq 1$. Keeping the log factor is a refinement that would considerably complicate the solution but would not change the result qualitatively.
(iii) $\ln(\eta/10) \ll h_i$. Dropping the term $\ln(\eta/10)$ is clearly justified for typical values of h_i ($h_i > 100$) and η ($\eta = 0.05$).

Value of the exponent.

Solving for f^* by keeping the general exponent λ ($0.5 \leq \lambda \leq 1$) in Equation (2) does not yield a simple and elegant solution. The exponent $\lambda = 1$ chosen in our calculations corresponds to the "learnable rule" case, that is the case of a vanishingly small training error. A large fraction of pattern recognition problems closely fulfill this requirement. When our hypothesis is violated, we can qualitatively understand the effect of a smaller exponent λ by looking at Figure 2. For typical values of $a = h_{max}/t \leq 1/30$, $af/(1-f) < 1$ for the most part of the curve. Therefore, $(af/(1-f))^\lambda$ will be above $af/(1-f)$ near the minimum. This will tend to decrease the value of f^*. Therefore, if our "learnable rule" hypothesis is violated, the value of f^* proposed in Table 1 is an over-estimate.

4.2 Uncertainty term

Chernoff bounds.

According to Chernoff [11], The following bounds are valid with probability $(1-\alpha)$:

$$p_i - \hat{p}_i \leq \sqrt{-2\ln\alpha}\sqrt{\frac{p}{n}} \quad , \tag{28}$$

$$p_i - \hat{p}_i \geq -\sqrt{-3\ln\alpha}\sqrt{\frac{p}{n}} \quad . \tag{29}$$

where $\hat{p}_i$ is a test error rate calculated on n examples.

Let us call $\varepsilon_0(n, \alpha)$ the quantity:

$$\varepsilon_0(n, \alpha) = \frac{\ln(1/\alpha)}{n} \quad . \tag{30}$$

For large values of p_i, $p_i \simeq 1$, the bounds (29) simplify to: $p_i - \hat{p}_i \leq \sqrt{2\varepsilon_0}$ and $p_i - \hat{p}_i \geq -\sqrt{3\varepsilon_0}$.

More refined bounds can be obtained from (29) by solving a second degree equation (as explained in Reference [7], page 148). For instance, the right side bound obtained is:

$$p_i - \hat{p}_i \leq \frac{\ln\alpha}{n}\left(1 + \sqrt{1 + \frac{2n\hat{p}_i}{-\ln\alpha}}\right) \quad . \tag{31}$$

For small values of $\hat{p}_i$, these bounds simplify to: $p_i - \hat{p}_i \leq \varepsilon_0$ and $p_i - \hat{p}_i \geq -\frac{3}{2}\varepsilon_0$.

For the purpose of making qualitative statements, we replace all these bounds by a simple power law $|p_i - \hat{p}_i| \leq (C\varepsilon_0)^\mu$, where C is a constant and $0.5 \leq \mu \leq 1$.

Value of the exponent.

In equation (16), we chose $\mu = 1$ and $C \leq 1.5$ to simplify our calculations. This hypothesis coresponds to a $\hat{p}_i$ vanishingly small. Applied to $i = opt$ and $i = val$, this means that the error on

the validation set of the recognizer *opt* and of the recognizer *val* should be close to zero ("learnable rule" at the second level of inference). This simplifying hypothesis may be quite often violated.

The effect of a smaller exponent than $\mu = 1$ on the "uncertainty" term can be qualitatively understood from Figure 2. For typical values of b ($b = C \ln(N/\alpha^2)/t \ll 1/100$), $b/f < 1$ for the most part of the curve. Therefore, $(b/f)^\mu$ will be above (b/f) near the minimum. This will tend to increase the value of f^*. Therefore, if our hypothesis of small error rate on the validation set is violated, the value of f^* proposed in Table 1 is an under-estimate.

5 Conclusion

We derived a formula for splitting the training database into training set and validation set valid for large training databases and small error rates.

If we call N is the number families of recognizers, h_{max} the largest complexity of those families, f the validation set size and g the training set size, the ratio f/g scales like $\sqrt{\ln N/h_{max}}$. For instance, for $N = 10$, if $h_{max} = 100$, 25% of the training data should be reserved for validation whereas if $h_{max} = 1000$, only 10% is sufficient.

Although this framework is not perfect and makes some simplifying assumptions, it sheds some light on the tradeoff between training set size and validation set size. The training set serves to determine the recognizers parameters (first level of inference) and the validation set serves to select which family of recognizers performes best (second level of inference). We find that optimum training-validation split is monitored by the ratio of the complexity of these two levels of inference.

This work was originally motivated by the organization of the UNIPEN benchmark [12]. The simplicity of our result should appeal to the experimentalists involved in the benchmark and to others.

Acknowledgements

We are very grateful to Vladimir Vapnik for his guidance and to Michael Kearns for communicating to us his work prior to publication and providing us with helpful comments on our draft.

References

[1] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik. What size test set gives good error rate estimates? *AT&T Bell Labs memorandum BL0115540-951206-07, submitted to PAMI*, 1995.

[2] W. H. Highleyman. The design and analysis of pattern recognition experiments. *The Bell systems technical journal*, pages 723–744, March 1962.

[3] M. Stone. Cross-validatory choice ans assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.

[4] M. Stone. Asymptotics for and against cross-validation. *Biometrica*, 64(1):29–35, 1977.

[5] M. Kearns. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. In *Advances in Neural Information Processing Systems 7 (NIPS 95)*, 1996, to appear.

[6] S. Amari, N. Murata, K.-R. Müller, M. Finke, and H. Yang. Asymptotic statistical theory of overtraining and cross-validation. *Submitted to IEEE Trans. on Neural Networks*, 1995.

[7] V.N. Vapnik. *Estimation of dependences based on empirical data.* Springer, New York, 1982.

[8] J. Rissanen. *Stochastic complexity in statistical inquiry*, volume 15. Wold Scientific, Series in Computer Science, 1989.

[9] Akaike. A new look at statistical model identification. *IEEE Trans.*, AC-19:716–723, 1974.

[10] C. Cortes, L. D. Jackel, S. A. Solla, V. Vapnik, and J. S. Denker. Learning curves: Asymptotic values of rate of convergence. Technical Report 11359-931111-11TM, AT&T Bell Labs, Holmdel, New Jersey, 1993.

[11] H. Chernoff. A measure of assymptotic efficientcy for tests of a hypothesis based on the sums of observations. *Ann. Math. Stat.*, 23:493–509, 1952.

[12] I. Guyon and L. Schomaker. UNIPEN project of data exchange and database benchmark. Technical Report TM-11359-921111-10, AT&T Bell Laboratories, Holmdel, New Jersey, USA, 1993.