The Bin Model

Yaser Abu-Mostafa
Learning Systems Group
California Institute of Technology
136-93 Pasadena, CA, 91125

Xubo Song
Department of Electrical Engineering
Oregon Graduate Institute
20000 N.W. Walker Road
Beaverton, OR, 97006

Alexander Nicholson
Learning Systems Group
California Institute of Technology
136-93 Pasadena, CA, 91125

Malik Magdon-Ismail
Learning Systems Group
California Institute of Technology
136-93 Pasadena, CA, 91125

Abstract

We propose a novel theoretical framework for understanding learning and generalization which we call *the bin model*. Using the bin model, a closed form is derived for the generalization error that estimates the out-of-sample performance in terms of the in-sample performance. We address the problem of overfitting, and show that using a simple exhaustive learning algorithm it does not arise. This is independent of the target function, input distribution and learning model, and remains true even with noisy data sets. We apply our analysis to both classification and regression problems and give an example of how it may be used efficiently in practice.

Keywords:Learning theory, generalization, noisy data, VC dimension, out of sample error, overfitting

1 Introduction

Learning from examples is one of the standard techniques for dealing with unstructured or mathematically ill-defined problems (see for instance [7][4][8][16] for an introduction). The task at hand is to extract relevant information from a finite set of examples to develop predictive models. If only a small amount of data is available (as is often the case in real applications), a sufficiently complex model could effectively store the data in a lookup table. This results in perfect performance on one data set, but no predictive ability or generalization. In contrast, a simple model may capture the underlying characteristics of the data without being able to replicate it exactly. In order to select an appropriate learning model, we need to have some way of predicting or evaluating the generalization performance. Indeed, the main theoretical question in learning from examples is that of generalization.

†This work was supported by the NSF Engineering Research Center for Neuromorphic Systems Engineering at Caltech.

‡This manuscript was prepared in December 1999 with minor revisions made in April 2002 and remains unchanged except for this footnote. Xubo Song is currently with the Department of Computer Science and Engineering, OGI School of Science and Engineering, Oregon Health and Science University, Alexander Nicholson is currently with SAC Capital Management, LLC, and Malik Magdon-Ismail is currently with Rensselaer Polytechnic Institute.

Here we present a mathematical paradigm for generalization [2] in which each hypothesis is modeled as a *bin* containing colored marbles indicating agreement or disagreement with the target function. This 'Bin Model' relates a classification problem to a physical experiment that is easy to visualize, and which lends itself to probabilistic analysis. The most significant result gives the expected generalization error for a given level of training error (equation 13). Thus, when we formalize a learning process in terms of the bin model, we can directly find the quantity of interest – the expected performance on new data. Even in the cases where the quantity cannot be expressed in closed form, the equation still enables us to prove properties of generalization, such as overfitting, without the need for a closed-form solution. The simplicity of the bin model enables us to accommodate noisy examples using a binary symmetric channel. Further analysis extends the model to regression problems as well as input-dependent noise.

We begin with a discussion of approaches to addressing the problem of generalization in learning theory. We then set up the learning problem and introduce notation for the rest of the paper. The bin model and main generalization results are presented in section 2 in the context of classification problems. The effect of noise on learning and generalization is discussed in section 3 and in section 4, we extend the analysis to regression problems. A method for using the bin model analysis with practical learning algorithms is given in section 5. Finally, we conclude with a discussion of application areas and future directions for research.

1.1 The Learning Problem

In a learning problem, we are presented with a data set $\mathcal{D}$, the *training set*, which consists of N input-output pairs $\{x_i, y_i\}_{i=1}^{N}$. The input-output pairs are generated according to an *unknown* underlying function $f : \mathcal{X} \to \mathcal{Y}$, which we call the *target function*, so that $y_i = f(x_i)$. Each $x_i \in \mathcal{X}$ is drawn from some input probability measure $p_X(x)$.[1] The training set is our sole knowledge pertaining to the target function, and our goal is to infer the target function based on the training set. Learning entails choosing a hypothesis function $g : \mathcal{X} \to \mathcal{Y}$ from a collection of candidate functions $\mathcal{G}$ as our inference of the target function. The set $\mathcal{G}$ is called the *learning model* because it reflects how we choose to model the target function. The hypothesis function is chosen by a *learning algorithm*, $\mathcal{A}$. The learning algorithm takes as inputs the training set $\mathcal{D}$ and the learning model $\mathcal{G}$, and outputs a hypothesis $g \in \mathcal{G}$, usually based on some performance criterion on the training set. For example, a typical learning algorithm might be to choose from the learning model the hypothesis which minimizes an error measure on the training set.

We measure how well the hypothesis g matches the target function f based on some error measure $e : \mathcal{G} \times \mathcal{X} \to \mathbf{R}$. Error occurs when a hypothesis deviates from the target function on any point in the input space. The error a hypothesis function commits on the training set is referred to as the in-sample or training error, and the error on points out of the training set is termed the out-of-sample or test error. A good hypothesis should give low out-of-sample error. A "clever" learning algorithm might be able to find a hypothesis that fits the training set well and has a small in-sample error. However, in-sample error can have no bearing on the out-of-sample error[18].

Typically, two possible scenarios may occur in a learning process as illustrated in Figure 2. In the first scenario, the out-of-sample error decreases as the in-sample error decreases, all the way until the in-sample error reaches its minimum. In this case, it is good for a learning algorithm to find a hypothesis that achieves the minimum in-sample error, since the corresponding out-of-sample error is also minimal. In the second scenario, the out-of-sample error decreases at the beginning together with the in-sample error as the learning algorithm is getting a grasp of the general properties of the target function contained in the training set.

[1] In general we will denote by $p_R(\cdot)$ the probability distribution function of random variable R. We use $\Pr[\cdot]$ to denote probabilities of events and $E_R[\cdot]$ for expectations with respect to R.

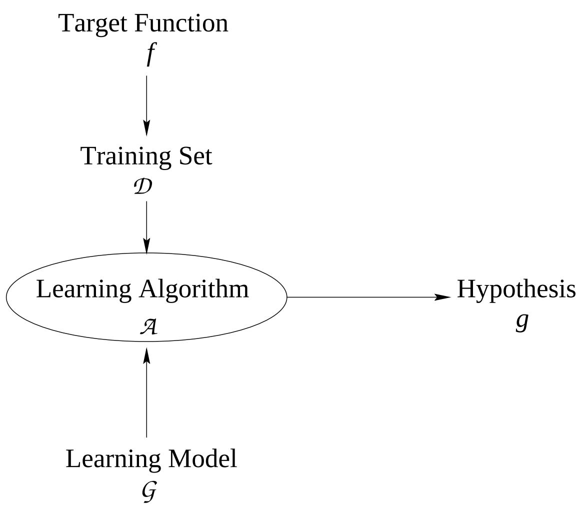


Figure 1: The learning process.

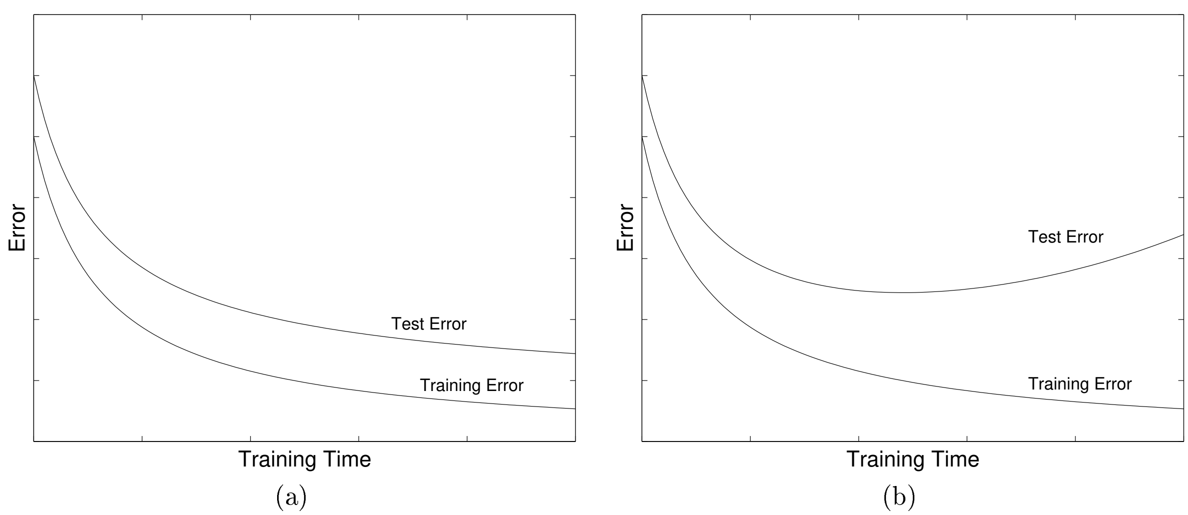


Figure 2: Two learning scenarios: overfitting occurs in the second one.

Then at certain point, the out-of-sample error starts to rise as the in-sample error is further reduced. The learning algorithm is finding hypotheses that fit the training data better, but that also fit any idiosyncrasies in the training set. As a result, the hypothesis approximates the training data set well, but fails to generalize to out-of-sample data. This phenomenon is usually referred to as *overfitting*. Overfitting occurs when the learning model is overly complex and is especially prominent when the training set is noisy.

This raises some important questions: When can we be confident that our model has truly learned and not simply memorized the examples it is given? How much does the in-sample error tell us about the out-of-sample error? Does the in-sample performance generalize to unseen data points? This is the issue of *generalization*. It is an important issue when learning from examples because the training error is the only quantity we have access to, and the whole merit of learning lies in the hope that training error, to some extent, provides some information about the test error.

In general, both the target function and the learning model are nonlinear, and the training set is noisy, thus making technical analysis difficult. It would be nice to have a general framework that, for arbitrary target functions, arbitrary learning models, arbitrary input distributions and noisy data sets, can estimate the out-of-sample performance in terms of the in-sample performance, characterize the conditions for overfitting, quantify the effect of noise on generalization, and eventually, lead to rules for selecting learning models and learning algorithms that can achieve better generalization performance. This is the motivation for our proposition and design of the Bin Model. The Bin Model is a general yet manageable mathematical framework from which we extract the essential property p_π of a learning process that combines the properties of the learning model, input distribution and target function. Given some knowledge of p_π, a closed form relationship between in- and out-of-sample performance can be derived. In this framework, we can handle the general nonlinear and noisy cases of learning. Important issues in learning, such as generalization, overfitting, the role of model complexity and the impact of noise are addressed.

1.2 Generalization

Once we have selected a hypothesis g to model our target function, we would like to know how well it will perform on data not necessarily in the training set (out-of-sample data). That is, we would like to know if the hypothesis can *generalize* from the training set to the entire space $\mathcal{X}$.

Many attempts have been made to assess the generalization performance of a learning process. The Prediction Error [3][11], VC analysis [17][1], and Exhaustive Learning paradigm [14][13] are some of the significant results along these lines.

The Prediction Error approach gives a general form for the out-of-sample error that consists of the sum of two terms:

$$\text{out-of-sample error} \approx \text{in-sample error} + \frac{2C}{N}\sigma^2 \tag{1}$$

where C embeds model complexity and therefore the second term can be viewed as a penalty for more complex models. σ^2 is the variance of noise in the data. This criterion has the nice property of formulating the impact of model complexity and data noise on the gap between in- and out-of-sample error. The prediction error criterion is an asymptotic result, and for the nonlinear case [11], it is assumed that the input density is discrete with support only at input points. These are required for mathematical feasibility, but may limit the accuracy of the approximation in practice.

Some insights into generalization are gained by bounding the deviation between training and test error in the worst-case scenario. VC theory provides exactly such a bound:

$$P[\sup_{g\in\mathcal{G}}|\pi_g-\nu_g|>\varepsilon]<\delta(\varepsilon,N,d_{VC})$$

where ε is a tolerance level for the deviation between in-sample error ν_g and out-of-sample error π_g for a hypothesis g, N is the number of training examples available and d_{VC} is a parameter related to the model complexity (the VC-dimension). This criterion highlights the impact of number of training examples N and model complexity on generalization. Because it is a worst-case analysis, the VC bound can be applied universally. This is not without drawbacks, though, since in practical scenarios the bound is often found to be weak. Also, the confidence level δ depends on the *growth function* [1], the calculation of which can be formidable for general learning models.

Schwartz et al. [13] suggested a framework with a somewhat similar formulation as our Bin Model which they call *exhaustive learning*. Their work leads to the following results:

$$\begin{aligned} G_N &= \int_0^1 g\rho_N(g)dg && (2)\\ &= \frac{\int_0^1 g^{N+1}\rho_0(g)dg}{\int_0^1 g^N\rho_0(g)dg} && (3)\end{aligned}$$

G_N is termed the 'mean generalization ability', and we will see that it effectively corresponds to $E[\pi]$ in our analysis. Similarly, $1-g$ is equivalent to our definition of π, and $\rho_0(1-g)$ corresponds to $p_\pi(\pi)$. $\rho_N(g)$ corresponds to $p(\pi|\nu=0)$ with N examples in the bin model setting. In other words, the analysis of [13] addresses the expected test error given zero training error, in our notation $E[\pi|\nu=0]$.

1.3 Learning Algorithms

In many typical learning algorithms, a starting point is chosen, and a small set of hypotheses are explored according to some sequential rule. Gradient based methods [12], for example, typically explore only a few hypotheses that lie on a path of descent in parameter space. This results in a great speed advantage, but is susceptible to getting stuck in local minima. Global optimization techniques like simulated annealing [9] explore a much greater number of hypotheses. The error spaces corresponding to familiar parameterized learning models like artificial neural networks tend to have many symmetries [4], so there remains the question of selecting from hypotheses with the same in-sample performance.

In the bin model analysis, we consider hypotheses to be selected from the learning model in a way similar to that of the exhaustive learning paradigm discussed above. Hence we refer to this as the *exhaustive learning algorithm*.

We assume there is a probability distribution $p_{\mathcal{G}}$ over the set of hypotheses in the learning model. We have some performance criterion that we wish to satisfy, and which is satisfied by a subset of the learning model, $\mathcal{G}^*$. The output of the learning algorithm is one hypothesis from $\mathcal{G}^*$ randomly selected according to the underlying distribution.

In some simple cases, we can implement exhaustive learning by finding $\mathcal{G}^*$ explicitly and making a random selection. It is often more practical, however, to repeatedly make random selections (with replacement) of g from $\mathcal{G}$, stopping when we find $g\in\mathcal{G}^*$. The two approaches are equivalent, with the selected hypothesis distributed as

$$p(\mathcal{A}(\mathcal{D}_N)=g)=\frac{p_{\mathcal{G}}(g)}{\int_{\mathcal{G}^*}p_{\mathcal{G}}(g)} \quad (4)$$

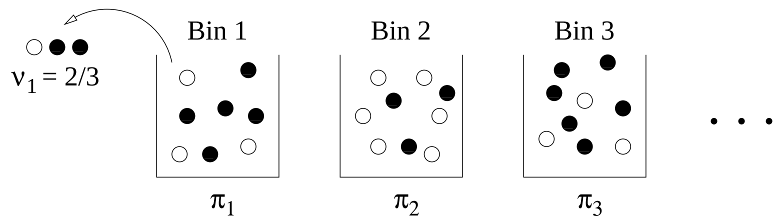


Figure 3: We have a set of bins, each containing red and green marbles. The fraction of red marbles in a given bin is denoted by π. A sample from the first bin has two red marbles and one green marble, giving $\nu = 2/3$.

in either case.

2 The Bin Model for Classification

We introduce the bin model in the context of a binary classification problem, that is $f : \mathcal{X} \to \{0, 1\}$ and $g : \mathcal{X} \to \{0, 1\}$. We use the error function

$$
\begin{aligned}
e(g, x) &= (g(x) - f(x))^2 && (5)\\
&= \begin{cases} 0 & g(x) = f(x) \\ 1 & g(x) \neq f(x) \end{cases} && (6)
\end{aligned}
$$

We can envision a learning model as a collection of bins. Each bin ($g \in \mathcal{G}$) contains a number of marbles, each colored red or green. Each marble corresponds to a point $x \in \mathcal{X}$, and is colored green if $g(x) = f(x)$ and red otherwise. Then observing the behavior of a hypothesis g on a data set $\mathcal{D}$ corresponds to sampling marbles from the appropriate bin and observing their colors.

Mathematically, green marbles correspond to correct classifications (0 error), and red marbles to incorrect classifications (1 error). Each hypothesis g has an inherent probability of error, which we denote by $\pi(g)$. $\pi(g)$ represents the fraction of red marbles in the corresponding bin.

$$
\begin{aligned}
\pi(g) &= \Pr[g(x) \neq f(x)] && (7)\\
&= E_x[e(g, x)] && (8)
\end{aligned}
$$

We denote the in-sample (training) error on the data set $\mathcal{D}$ by $\nu_{\mathcal{D}}$.

$$\nu_{\mathcal{D}}(g) = \frac{1}{N} \sum_{i=1}^{N} e(g, x_i) \qquad (9)$$

For a given hypothesis with error probability π_0, the probability for an in-sample error ν is given by the binomial distribution,

$$\Pr[\nu | \pi_0] = \binom{N}{N\nu} \pi_0^{N\nu} (1 - \pi_0)^{N(1-\nu)} \qquad (10)$$

We assume there is a probability distribution p_G on the set of hypotheses. p_G induces a π-distribution p_π. We use the exhaustive learning algorithm of section 1.3 to select a hypothesis with in-sample error ν_0. We can then explicitly compute the expected out-of-sample error, $\pi(\nu_0) = E[\pi|\nu = \nu_0]$.

$$\begin{aligned}
\pi(\nu_0) &= \int_0^1 s\ p_{\pi|\nu}(s|\nu_0)ds & (11)\\
&= \frac{\int_0^1 s\ p_{\pi,\nu}(s,\nu_0)ds}{\int_0^1 p_{\pi,\nu}(s,\nu_0)ds} & (12)\\
&= \frac{\int_0^1 s\ p_\pi(s)s^{N\nu_0}(1-s)^{N(1-\nu_0)}ds}{\int_0^1 p_\pi(s)s^{N\nu_0}(1-s)^{N(1-\nu_0)}ds} & (13)
\end{aligned}$$

We refer to the relationship between ν and $\pi(\nu)$ in (13) as the *generalization curve.* A perfect generalization curve should be $\pi(\nu) = \nu$, which implies that in-sample error ν is a perfect indication of out-of-sample error in expectation. It is not the absolute value of ν and $\pi(\nu)$, but the deviation between them that characterizes the generalization behavior of a learning process. The further a generalization curve is from $\pi(\nu) = \nu$ at any point, the worse the generalization is at that point. It is important to point out that no assumption is made about the dependence among the hypotheses. The statistical dependence among the hypotheses in the learning model does not enter the π-distribution.

Figure 4 (b) is the generalization curve based on the π-distribution given in 4 (a). According to this model, when the training error ν is zero, the expected out-of-sample error is actually 0.17, indicating this sample error is an optimistic estimation of the corresponding out-of-sample error.

It should be noted that in the special case that $\Pr[\pi \in \{0,1\}] = 1$, $\pi(\nu)$ will be undefined for $\nu \notin \{0,1\}$. We call such a learning model *strictly degenerate.* These cases are not of practical interest, since any hypothesis can trivially be made to indicate the target function perfectly. We therefore make the assumption that the learning model is not strictly degenerate (and hence that $\pi(\nu)$ is defined for $\nu = \frac{i}{N}, i = 0, 1, \ldots, N$).

We say that a function $F(x)$ is *monotonically increasing (decreasing)* in x iff $x_1 \leq x_2 \Rightarrow F(x_1) \leq F(x_2)$ ($x_1 \leq x_2 \Rightarrow F(x_1) \geq F(x_2)$). The following theorem gives a qualitative relationship between in-sample and out-of-sample errors.

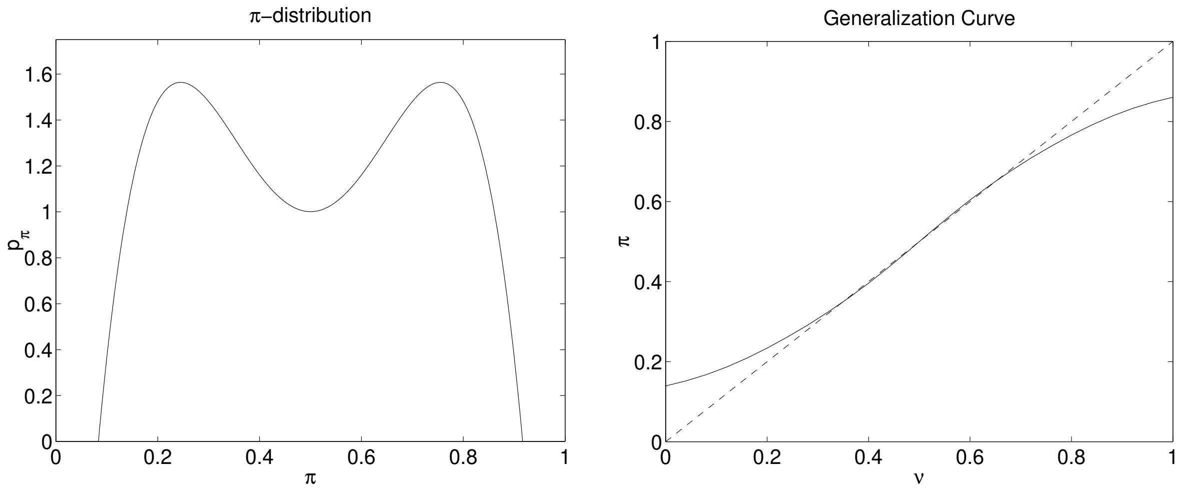
Theorem 2.1 *The expected test error $\pi(\nu)$ is monotonically increasing in the empirical error ν.*

See appendix A for the proof. Theorem 2.1 tells us that there is no overfitting with the exhaustive learning algorithm – to improve the out-of-sample error we should always find a hypothesis that does better on the training set.

For certain π-distributions, we can obtain $\pi(\nu)$ explicitly. For illustrative purposes we consider the case $p_\pi(\pi) = (d+1)\pi^d$, for $d \in \mathbf{Z}^+$ (nonnegative integers). The generalization curve is

$$\pi(\nu) = \frac{N\nu + d + 1}{N + d + 2}, \tag{14}$$

Of particular interest is the case $d = 0$. Then $p(\pi) = 1$ is the uniform distribution, which implies that we have no bias or preference over the distribution of π. $\pi(\nu)$ is linear in ν, but is skewed towards $\pi = \frac{1}{2}$ depending on the amount of data (see figure 5). If there is only one training example ($N = 1$), then $\pi(\nu) = \frac{1}{3}\nu + \frac{1}{3}$. As the number of examples grows ($N \to \infty$), $\pi(\nu) \to \nu$ indicating that the training error is a perfect indication of the generalization error, as is expected.



(a) (b)

Figure 4: (a)An example π-distribution. In this case the hypothesis that best approximates the target function makes approximately 8% error. (b) The expected test error as a function of training error ν (on 25 examples) for the π-distribution in (a). The dashed curve illustrates perfect generalization, $\pi(\nu) = \nu$.

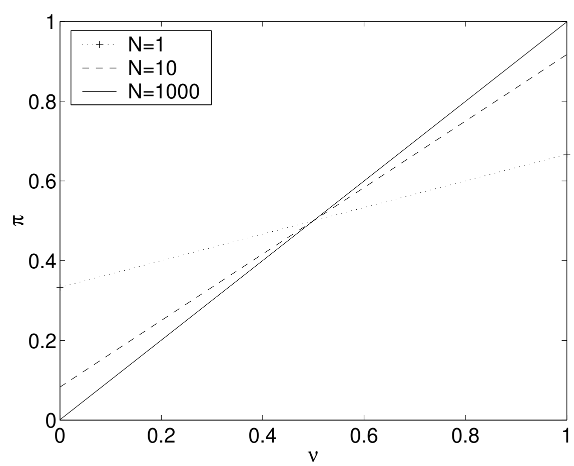


Figure 5: Generalization curves for a uniform π-distribution and varying training set size. As the amount of data increases, $\pi(\nu)$ approaches the perfect generalization curve.

3 The Effect of Noise

3.1 Uniform Noise

In the case of a binary classification problem, we can consider noise in the output to be a random flip of the classification with some probability. We define the input-independent noisy version $\tilde{f}$ of the target function f as

$$\tilde{f}(x) = \begin{cases} f(x) & \text{with probability } 1-\varepsilon_0 \\ 1-f(x) & \text{with probability } \varepsilon_0 \end{cases} \tag{15}$$

This is the same type of noise present in a Binary Symmetric Channel [6]. This BSC noise is easily incorporated into the Bin Model. Let $\tilde{\pi}(g)$ denote the error of a hypothesis g with the noisy target function $\tilde{f}$. We can write $\tilde{\pi}$ in terms of π.

$$\begin{aligned} \tilde{\pi}(g) &= \Pr_x[g(x) \neq \tilde{f}(x)] && (16)\\ &= (1-\varepsilon_0)\Pr_x[g(x) \neq f(x)] + \varepsilon_0 \Pr_x[g(x) = f(x)] && (17)\\ &= (1-\varepsilon_0)\pi(g) + \varepsilon_0(1-\pi(g)) && (18)\\ &= \pi(g)(1-2\varepsilon_0) + \varepsilon_0 && (19)\\ & && (20) \end{aligned}$$

If we know the distribution p_π, we can find the distribution of $\tilde{\pi}$ by a change of variables.

$$p_{\tilde{\pi}}(\tilde{\pi}) = \frac{p_\pi((\tilde{\pi}-\varepsilon_0)/(1-2\varepsilon_0))}{1-2\varepsilon_0} \tag{21}$$

Thus the result of adding BSC noise results in a linear transformation of π and a compression of the π-distribution. This effect is illustrated in figure 6. The π-distribution is squeezed towards $\pi = \frac{1}{2}$, indicating that the performance of every hypothesis is a bit closer to that of random guessing.

Now we can study $\tilde{\pi}(\nu) = E[\tilde{\pi}|\nu]$. Theorem 2.1 immediately applies – $\tilde{\pi}(\nu)$ is monotonically increasing in ν, thus for $\varepsilon_0 < \frac{1}{2}$, $\pi(\nu)$ is also monotonic. Thus there is no overfitting, even in the case of a noisy data set.

3.2 Input Dependent Noise

The noise in a data set may be input dependent. For example, on an experimental measurement, the uncertainty can depend on the characteristics of the particular instrument used, and may change for different measurement scales.

We model this by writing the noise $\varepsilon = \varepsilon(x)$. We are now concerned with the probability of the hypothesis agreeing with the *noisy* target value. Once again we define a noisy target function $\tilde{f}$.

$$\tilde{f}(x) = \begin{cases} f(x) & \text{with probability } 1-\varepsilon(x) \\ 1-f(x) & \text{with probability } \varepsilon(x) \end{cases} \tag{22}$$

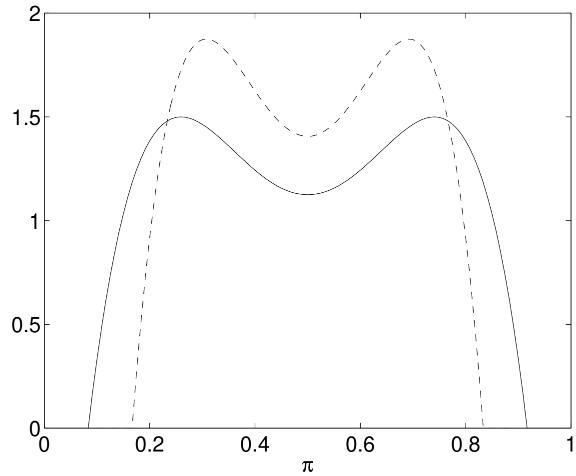


Figure 6: A π-distribution and the resulting $p_{\tilde{\pi}}$ under BSC noise with $\varepsilon_0 = 0.1$. The solid line shows the original p_π and the dashed line shows the noisy version.

$$
\begin{aligned}
\Pr[g(x) \neq \tilde{f}(x)|x] &= e(g,x)(1-\varepsilon(x)) + (1-e(g,x))\varepsilon(x) && (23)\\
\tilde{\pi}(g) &= \Pr[g(x) \neq \tilde{f}(x)] && (24)\\
&= E_x[e(g,x)(1-\varepsilon(x)) + (1-e(g,x))\varepsilon(x)] && (25)\\
&= E_x[e(g,x)] + E_x[\varepsilon(x)] - 2E_x[e(g,x)\varepsilon(x)] && (26)\\
&= \pi(g) + E_x[\varepsilon(x)] - 2E_x[e(g,x)\varepsilon(x)] && (27)
\end{aligned}
$$

Of particular interest are noise models which allow $\tilde{\pi}(g)$ to be determined given only $\pi(g)$.

Definition 3.1 *We say a noise model $\varepsilon(x)$ is* **regular** *iff $\tilde{\pi}(g) = \tilde{\pi}(\pi(g))$.*[2]

In the case of constant noise $\varepsilon(x) = \varepsilon_0 \ \forall x$, (27) reduces to the result of section 3.1, $\tilde{\pi} = \pi(1-2\varepsilon_0) + \varepsilon_0$.

We illustrate the case of input dependent noise with a simple example. Consider the learning model that consists of simple threshold functions, $g_w(x) = sgn(x-w), w \in [0,1]$. Furthermore, assume that the target function is in the learning model, $f(x) = g_\alpha(x)$, and for simplicity that $\frac{1}{2} < \alpha < 1$. Assume that the input distribution, is uniform in $(0,1)$, and that exhaustive learning is used with w chosen uniformly in $(0,1)$.

For this simple learning problem, an error occurs ($e(g_w, x) = 1$) when either $w < x < \alpha$ or $\alpha < x < w$. It follows that $\pi(g_w) = |\alpha - w|$. Also for any noise function $\varepsilon(x)$ we can compute $\tilde{\pi}(g_w)$.

$$
\tilde{\pi}(g_w) = \begin{cases} \pi(g_w) + \int_0^1 \varepsilon(x)dx - 2\int_w^\alpha \varepsilon(x)dx & w < \alpha \\ \pi(g_w) + \int_0^1 \varepsilon(x)dx - 2\int_\alpha^w \varepsilon(x)dx & w > \alpha \end{cases} \tag{28}
$$

We notice that for $p < (1-\alpha)$ there are two hypotheses, $g_{\alpha-p}$ and $g_{\alpha+p}$ that have $\pi(g) = p$. For $(1-\alpha) < p < \alpha$ there is only one such hypothesis, and none has $\pi > \alpha$. Thus the noise model is regular if

[2]Equivalently, we need only say that $E_x[e(g,x)\varepsilon(x)]$ is a function of $\pi(g)$.

it satisfies $\tilde{\pi}(g_{\alpha-p}) = \tilde{\pi}(g_{\alpha+p})$ for all $p < (1-\alpha)$. For regular noise distributions, we can rewrite the noisy error $\tilde{\pi}$ as a function of the noiseless error π.

$$\begin{aligned}\tilde{\pi}(\pi) &= \tilde{\pi}(g_{\alpha-\pi}) && (29)\\ &= \pi(g_w) + \int_0^1 \varepsilon(x)dx - 2\int_{\alpha-\pi}^{\alpha} \varepsilon(x)dx && (30)\end{aligned}$$

Given a training error ν on N points, we are interested in computing the noisy and noiseless expected test errors, $\tilde{\pi}(\nu)$ and $\pi(\nu)$ respectively.

$$\begin{aligned}P_{\nu|g}[\nu|g] &= P_{\nu|\tilde{\pi}}[\nu|\tilde{\pi}(g)] && (31)\\ &= \binom{N}{N\nu}\tilde{\pi}(g)^{N\nu}(1-\tilde{\pi}(g))^{N(1-\nu)} && (32)\\ \tilde{\pi}(\nu) &= E_{g,x^N}[\tilde{\pi}|\nu] && (33)\\ &= \int_0^1 s\ p_{\tilde{\pi}|\nu}[s|\nu]ds && (34)\\ &= \frac{\int_0^1 s\ P_{\nu|\tilde{\pi}}[\nu|s]p_{\tilde{\pi}}(s)ds}{\int_0^1 P_{\nu|\tilde{\pi}}[\nu|s]p_{\tilde{\pi}}(s)ds} && (35)\end{aligned}$$

$p_{\tilde{\pi}}(\cdot)$ is the distribution of $\tilde{\pi}(g)$ induced by $p_{\mathcal{G}}(g)$. When the noise distribution is regular, we can write $\tilde{\pi}(\nu)$ and $\pi(\nu)$ in terms of the π-distribution, $p_{\pi}(\cdot)$.

$$\begin{aligned}P_{\nu|\pi}[\nu|\pi] &= P_{\nu|\tilde{\pi}}[\nu|\tilde{\pi}(\pi)] && (36)\\ &= \binom{N}{N\nu}\tilde{\pi}(\pi)^{N\nu}(1-\tilde{\pi}(\pi))^{N(1-\nu)} && (37)\\ \pi(\nu) &= E_{g,x^N}[\pi|\nu] && (38)\\ &= \int_0^1 s\ p_{\pi|\nu}[s|\nu]ds && (39)\\ &= \frac{\int_0^1 s\ P_{\nu|\pi}[\nu|s]p_{\pi}(s)ds}{\int_0^1 P_{\nu|\pi}[\nu|s]p_{\pi}(s)ds} && (40)\\ \tilde{\pi}(\nu) &= \frac{\int_0^1 \tilde{\pi}(s)P_{\nu|\pi}[\nu|s]p_{\pi}(s)ds}{\int_0^1 P_{\nu|\pi}[\nu|s]p_{\pi}(s)ds} && (41)\end{aligned}$$

In general, computing exact values for $\pi(\nu)$ and $\tilde{\pi}(\nu)$ is intractable. However, some general overfitting results can be obtained, namely, noisy test error is a monotonically increasing function of the noisy training error. Furthermore, when $\tilde{\pi}$ is a monotonically increasing function of π, the noiseless test error is also a monotonically increasing function of the *noisy* training error.

Theorem 3.1 *For any $\tilde{\pi}(g)$, $p_{\mathcal{G}}(\cdot)$ and noise $\varepsilon(x)$, $\tilde{\pi}(\nu)$ is monotonically increasing in ν.*

Theorem 3.2 *If $\varepsilon(x)$ is regular and $\tilde{\pi}(\pi)$ is monotonically increasing (decreasing) in π then for any $p_\pi(\cdot)$, $\pi(\nu)$ is monotonically increasing (decreasing) in ν.*

Theorem 3.1 tells us that overfitting will never be observed in the noisy out-of-sample error, and is a generalization of Theorem 2.1. Theorem 3.2 gives conditions on the noise under which overfitting will or will not occur in the noiseless out-of-sample error. For proofs see appendix A.

Figure 7 shows several interesting quantities for the example above with $\alpha = 0.7$ and $\varepsilon(x) = 16(x-\alpha)^4 - 8(x-\alpha)^2 + 1$. This choice of $\varepsilon(x)$ is regular for this learning problem, and we see that $\tilde{\pi}(\pi)$ is not monotonically increasing in π, and for the given $p(\pi)$, $\pi(\nu)$ is not monotonically increasing in ν.

3.3 Value of Noisy Examples

In [10], Magdon-Ismail et al. studied how noise affects generalization for a general class of learning systems. They showed that the generalization error for a learning problem with noise can be bounded by a function of the generalization error with noiseless data. Explicitly, they found that

$$\mathcal{E}_N(\sigma) \leq \mathcal{E}_N(0) + \frac{C_1\sigma^2 + C_2}{N} + o\left(\frac{1}{N}\right) \tag{42}$$

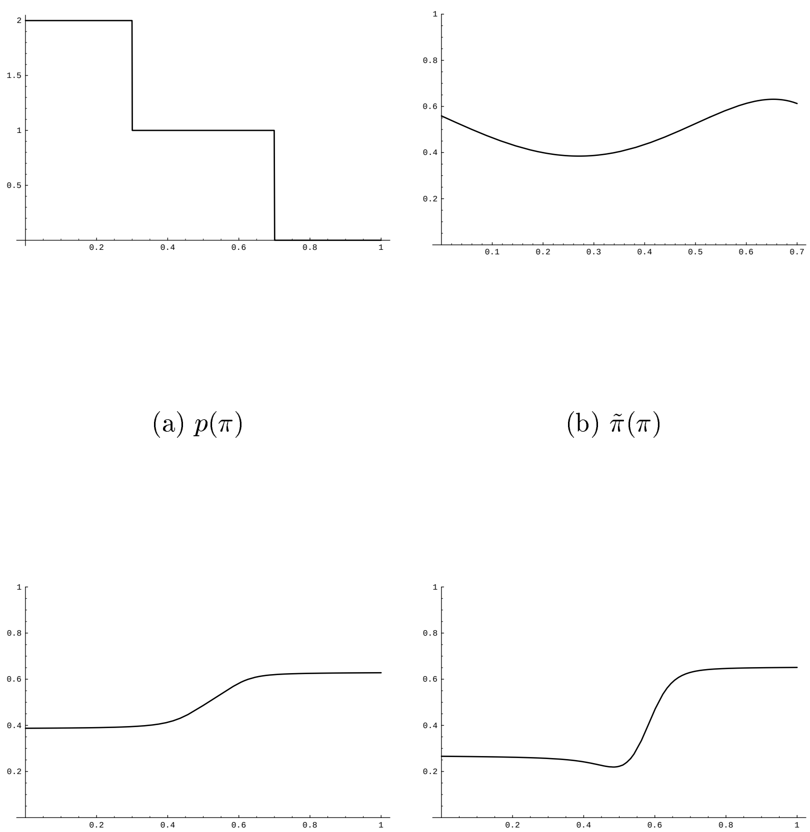
where $\mathcal{E}_N(\sigma)$ is the expected test error with N training examples and noise variance σ^2, and C_1 and C_2 are constants depending on the learning problem.[3]

We can derive a similar result for the bin model with exhaustive learning. For simplicity we consider the case that $\nu = 0$ and a uniform π-distribution, $p_\pi(\pi) = 1, \pi \in [0,1]$. We assume the noise is of the BSC type with probability of flip ε_0.

$$\begin{aligned}
E[\pi|\nu=0] &= \frac{\int_0^1 \pi(1-\pi)^N p_\pi(\pi)d\pi}{\int_0^1 (1-\pi)^N p_\pi(\pi)d\pi} && (43)\\
&= \frac{1}{N+2} && (44)\\
E[\pi|\nu=0,\varepsilon_0] &= \frac{\int_0^1 \pi(1-\tilde{\pi})^N p_\pi(\pi)d\pi}{\int_0^1 (1-\tilde{\pi})^N p_\pi(\pi)d\pi} && (45)\\
&= \frac{\int_0^1 \pi(1-(1-2\varepsilon_0)\pi - \varepsilon_0)^N d\pi}{\int_0^1 (1-(1-2\varepsilon_0)\pi-\varepsilon_0)^N d\pi} && (46)\\
&= \frac{1}{N+2}\left(1 + \frac{\varepsilon_0}{1-2\varepsilon_0} - \frac{\varepsilon_0^{N+1}}{\varepsilon_0^{N+1} - (1-\varepsilon_0)^{N+1}}\right) + \frac{\varepsilon_0^{N+1}}{\varepsilon_0^{N+1} - (1-\varepsilon_0)^{N+1}} && (47)\\
&= E[\pi|\nu=0] + \frac{1}{N+2}\left(\frac{\varepsilon_0}{1-2\varepsilon_0}\right) - \Theta(e^{-\alpha N}) && (48)
\end{aligned}$$

for some $\alpha > 0$, where we have assumed that $\varepsilon_0 < \frac{1}{2}$ for convenience.[4]

[3] We write $g(x) = o(f(x))$ iff $\lim_{x\to\infty} g(x)/f(x) = 0$.
[4] We write $g(x) = \Theta(f(x))$ iff $\lim_{x\to\infty} g(x)/f(x) \in (0,\infty)$.



(a) $p(\pi)$ (b) $\tilde{\pi}(\pi)$

(c) $\tilde{\pi}(\nu)$ (d) $\pi(\nu)$

Figure 7: Several interesting quantities for the input-dependent noise example of section 3.2. (a) shows the π-distribution, (b) the relation $\tilde{\pi}(\pi)$ between noisy and noiseless errors, (c) the noisy generalization curve $\tilde{\pi}(\nu)$ and (d) the noiseless generalization $\pi(\nu)$. In this case we have used $\alpha = 0.7$ and $\varepsilon(x) = 16(x-\alpha)^4 - 8(x-\alpha)^2 + 1$. This noise function is regular, but $\tilde{\pi}(\pi)$ is not monotonically increasing in π and $\pi(\nu)$ fails to be monotonically increasing in ν.

This result tells us what (noiseless) generalization error we can expect using exhaustive learning if we can estimate the noise in the training data and we are guaranteed to find $\nu = 0$. The explicit result is only for $p_\pi(\pi) = 1$, but similar results can be derived for other π-distributions.

To compare this result with (42), we need to write (48) in terms of σ^2. Since $\sigma^2 = \varepsilon_0(1-\varepsilon_0) \in [0, \frac{1}{4})$ for the BSC noise model, we can write

$$E[\pi|\nu = 0, \varepsilon_0] \quad = \quad E[\pi|\nu = 0] + \frac{1}{N+2}\left(\frac{1}{2\sqrt{1-4\sigma^2}} - \frac{1}{2}\right) - \Theta(e^{-\alpha N}) \tag{49}$$

We see that the number of training examples plays a similar role in (42) and (49), but that the effect of noise is somewhat different.

4 The Bin Model for Regression

So far, the bin model analysis has dealt only with classification problems. In order to incorporate continuous error measures and regression problems, we need to generalize the concept of π. We now redefine π in terms of an arbitrary error function $e : \mathcal{G} \times \mathcal{X} \to \mathcal{E}$ where $\mathcal{E} = [e_{min}, e_{max}]$, $e_{min} = inf(e)$ and $e_{max} = sup(e)$.

For any hypothesis g, the error values $e(g, x)$ are distributed according to some distribution $p_{e|g}(e|g)$. Again we denote the expected generalization error by $\pi(g)$.

$$\begin{aligned}
\pi(g) &= E_x[e(g,x)] &(50)\\
&= \int_{\mathcal{X}} e(g,x) p_{\mathcal{X}}(x) dx &(51)\\
&= \int_{e_{min}}^{e_{max}} s \; p_{e|g}(s|g) ds &(52)\\
& &(53)
\end{aligned}$$

where $p_{e|g}$ is the distribution induced on e by $p_{\mathcal{X}}$.

For the regression analysis, we will also need to find the variance of the errors for a given hypothesis, which we denote by $\sigma^2(g)$.

$$\begin{aligned}
\sigma^2(g) &= E_x[(e(g,x) - \pi(g))^2] &(54)\\
&= \int_{e_{min}}^{e_{max}} (s - \pi(g))^2 p_{e|g}(s|g) ds &(55)
\end{aligned}$$

For a general error function, π and ν now can take any values in $\mathcal{E}$. Once again, the quantity of interest is the expected generalization error $\pi(\nu)$. In the binary classification problem, we found that $\pi(\nu)$ was a monotonically increasing function of ν. For regression problems, this is not necessarily true. We would like to determine the conditions under which $\pi(\nu)$ is always monotonically increasing in ν, that is, under which $\frac{d}{d\nu}\pi(\nu) \geq 0$ for all ν.

4.1 Monotonicity Conditions

As with classification problems, we would like to abstract the characteristics of the learning problem into the π-distribution. Given p_π, we can evaluate $\pi(\nu)$ if we assume that, given $\pi(g)$, there is some conditional distribution $p_{\nu|\pi}$ for the in-sample error. We can determine conditions on $p_{\nu|\pi}$ that guarantee monotonicity of the generalization curve.

Theorem 4.1 *If $p_{\nu|\pi}$ is non-zero, continuous and differentiable on $(\nu, \pi) \in \mathcal{E}^2$, then $\pi(\nu)$ is monotonically increasing in ν for any p_π iff $(\frac{\partial}{\partial\nu} p_{\nu|\pi})/p_{\nu|\pi}$ is monotonically increasing in π.*

See appendix A for the proof. Thus, if we can specify how the in-sample errors arise as a function of the out-of-sample error, we may be able to rule out overfitting under exhaustive learning. In general, however, specifying such a $p_{\nu|\pi}$ is very difficult. Something can be said, though, in the large data limit.

The in-sample error is

$$\nu = \frac{1}{N}\sum_{i=1}^{N} e(g, x_i) \in \mathcal{E} \tag{56}$$

Since each x_i is i.i.d. according to the input distribution, each $e(g, x_i)$ is i.i.d. according to some distribution $p_{e|g}$. Usually we don't know $p_{e|g}$, but we do know that it has mean $\pi(g)$ and variance $\sigma^2(g)$. The central limit theorem tells us that as $N \to \infty$,

$$\nu \rightsquigarrow \mathcal{N}(\pi(g), \frac{\sigma^2(g)}{N}) \tag{57}$$

That is, $p_{\nu|g}$ converges in distribution to a Gaussian with mean $\pi(g)$ and variance $\sigma^2(g)/N$.

$$p_{\nu|\pi,\sigma}(\nu|\pi, \sigma) \propto \exp(-\frac{N(\nu - \pi(g))^2}{2\sigma^2(g)}) \tag{58}$$

If for any g, the error variance is a function of the mean,[5] that is, $\sigma(g) = \sigma(\pi(g))$, then we can rewrite (58) and apply Theorem 4.1 to find necessary and sufficient conditions for monotonicity of $\pi(\nu)$.

$$p_{\nu|\pi}(\nu|\pi) \quad \propto \quad \exp(-\frac{N(\nu-\pi)^2}{2\sigma^2(\pi)}) \tag{59}$$

$$\frac{d}{d\nu}p_{\nu|\pi}(\nu|\pi) \quad = \quad \frac{N(\pi-\nu)}{\sigma^2(\pi)} p_{\nu|\pi}(\nu|\pi) \tag{60}$$

Theorem 4.2 *If $\sigma(g) = \sigma(\pi(g)) \quad \forall g \in \mathcal{G}$, then in the large data limit $N \to \infty$, $\pi(\nu)$ is monotonically increasing in ν for all p_π iff $\frac{e_{max}-\pi_1}{e_{max}-\pi_2} \leq \frac{\sigma^2(\pi_1)}{\sigma^2(\pi_2)} \leq \frac{\pi_1 - e_{min}}{\pi_2 - e_{min}}$ whenever $\pi_1 > \pi_2$.*

Theorem 4.2 gives a condition on the learning problem such that under exhaustive learning, asymptotically there will be no overfitting. The form of $\sigma(\pi)$ will generally depend on the learning model, target function and input distribution. We can find common distributions for which the condition either satisfied or violated. For the Bernoulli distribution (with $\mathcal{E} = \{0, 1\}$), $\sigma^2(\pi) = \pi(1-\pi)$ and the condition holds, reiterating the results of section 2. For a uniform distribution over $\mathcal{E} = \{0, e_{max}(g)\}$, however, $\pi(g) = e_{max}(g)/2$ and $\sigma^2(\pi) = \pi^2/3$, so the condition is violated.

[5] This is true for many familiar distributions, for example, when $p_{e|g}$ is binomial, exponential or χ^2.

4.2 Noise in Regression Problems

For the purposes of analyzing the effects of noise in the case of regression problems, it is useful to work in the frequency domain. Letting $\hat{p}$ denote the Fourier transform of p, we have

$$\begin{aligned} p_{\nu|g}(\nu|g) &= \prod^{\star} p_{e|g}(e(g,x_i)|g) && (61) \\ \widehat{p_{\nu|g}}(\omega) &= \widehat{p_{e|g}}^{N}(\omega) && (62) \end{aligned}$$

That is, $p_{\nu|g}$ is the N-fold convolution of $p_{e|g}$ with itself. We are interested in quantifying the effects of noise on $p_{e|g}$. This allows us to find $p_{\nu|g}$ for any $g \in \mathcal{G}$, allowing the computation of $p_{\nu|\pi}$ and hence the expected out-of-sample error $\pi(\nu)$. We consider the familiar case of a squared error measure and additive zero-mean Gaussian noise.

$$\begin{aligned} e(g,x) &= (g(x) - f(x))^2 && (63) \\ \tilde{f}(x) &= f(x) + \eta && (64) \\ \eta &\sim \mathcal{N}(0, \sigma_\eta^2) && (65) \\ \tilde{e}(g,x) &= (g(x) - \tilde{f}(x))^2 && (66) \end{aligned}$$

In this case we can exactly determine the relationship in the frequency domain between the noiseless error distribution $p_{e|g}$ and the noisy version $p_{\tilde{e}|g}$.

$$\widehat{p_{\tilde{e}|g}}(\omega) = \frac{\widehat{p_{e|g}}\left(\frac{\omega}{1+2i\sigma_\eta^2\omega}\right)}{\sqrt{1+2i\sigma_\eta^2\omega}} \qquad (67)$$

A derivation is given in Proposition A.1 in the appendix. Equation (67) gives us a transformation that is useful if the entire distribution $p_{e|g}$ is known. In the noiseless case, however, we could guarantee monotonicity of $\pi(\nu)$ knowing only $\pi(g)$ and $\sigma^2(g)$. With a squared error measure it is straightforward to determine these parameters for the noisy case in terms of the noiseless versions and the moments of the noise distribution p_η. For zero-mean, zero-skew input-independent noise

$$\begin{aligned} \tilde{\pi}(g) &= E_{x,\eta}[(g(x) - \tilde{f}(x))^2] && (68) \\ &= E_x[(g(x)-f(x))^2] - 2E_x[(g(x)-f(x))]E_\eta[\eta] + E_\eta[\eta^2] && (69) \\ &= \pi(g) + \sigma_\eta^2 && (70) \\ \tilde{\sigma}^2(g) &= E_{x,\eta}[(g(x) - \tilde{f}(x))^4] - \tilde{\pi}(g)^2 && (71) \\ &= E_x[(g(x)-f(x))^4] - 4E_x[(g(x)-f(x))^3]E_\eta[\eta] + 6E_x[(g(x)-f(x))^2]E_\eta[\eta^2] \\ &\quad -4E_x[g(x)-f(x)]E_\eta[\eta^3] + E_\eta[\eta^4] - (\pi(g)+\sigma_\eta^2)^2 && (72) \\ &= \sigma^2(g) + 4\pi(g)\sigma_\eta^2 + E_\eta[\eta^4] - \sigma_\eta^4 && (73) \end{aligned}$$

In the special case of Gaussian noise, we can simplify (73) to get an expression for $\tilde{\sigma}^2(g)$ that depends only on $\pi(g)$, $\sigma^2(g)$ and the noise variance σ_η^2.

$$\tilde{\sigma}^2(g) \quad = \quad \sigma^2(g) + 4\pi(g)\sigma_\eta^2 + 2\sigma_\eta^4 \tag{74}$$

Hence, as in section 3.1, the addition of input-independent noise results in a simple transformation of the essential parameters $\pi(g)$ and $\sigma^2(g)$. We can then apply Theorem 4.2 with the noisy $\tilde{\pi}(g)$ and $\tilde{\sigma}^2(g)$ to determine whether the monotonicity conditions are satisfied. For example, we consider the case of a Bernoulli distribution for $e(g,x) \in \{0,1\}$ so that $\sigma^2 = \pi(1-\pi)$. If we assume Gaussian noise ($e_{max} = \infty$) and use the transformations above, we have monotonicity of $\tilde{\pi}(\nu)$ when the following inequalities are true for all $\pi_1 > \pi_2$.

$$1 \leq \frac{\pi_1(1-\pi_1) + 4\pi_1\sigma_\eta^2 + 2\sigma_\eta^4}{\pi_2(1-\pi_2) + 4\pi_2\sigma_\eta^2 + 2\sigma_\eta^4} \leq \frac{\pi_1 + \sigma_\eta^2}{\pi_2 + \sigma_\eta^2} \tag{75}$$

The left inequality holds if $\sigma_\eta^2 > 1/4$, but if $p_\pi(\pi) > 0$ in some neighborhood of 0, then the right inequality is only satisfied for $\sigma_\eta^2 = 0$. In other words, the addition of Gaussian noise makes overfitting a possibility for this problem (which has monotonic generalization in the noiseless case).

5 Learning Algorithms

One of the major limitations of the bin model thus far is that it requires an exhaustive learning algorithm. In real applications, the exhaustive learning algorithm is impractical. Results similar to Theorem 2.1 can be obtained in special cases [5], but in general the selected hypotheses will depend on the training set in ways specific to the individual learning algorithm. Nevertheless, there is a way that we can use an arbitrary learning algorithm and still take advantage of the bin model analysis through use of a validation set[15].

Given a data set $\mathcal{D}$, we divide it into a training set $\mathcal{D}_T$ and a validation set $\mathcal{D}_V$. The training set will be used to select a hypothesis using a learning algorithm $\mathcal{A}$. We have altered the black-box that produces the hypotheses. $p_\mathcal{G}(g)$ is now the distribution induced by $\mathcal{A}$ on the distribution of possible training sets. The new process is illustrated in figure 8.

There is now a training set underlying the selection process, but for the purpose of generalization analysis, the learning problem is the same. The hypotheses are produced according to some distribution $p_\mathcal{G}$, and we can find the generalization curve with respect to p_π, this time using $\mathcal{D}_V$ to compute the in-sample error. We now have a smaller number of examples, but presumably the hypotheses produced are better.

As an example of the improvement we can expect by using this two step learning, we consider a two dimensional classification problem. The target function f is a linear classifier, and $\mathcal{G}$ is a linear perceptron learning model. Figure 9 compares the π-distributions and generalization curves for exhaustive learning and the perceptron learning rule. While the π-distribution for the original $p_\mathcal{G}$ is relatively flat, using 10 examples to train with the perceptron skews the distribution to favor hypotheses with $\pi < 0.2$. As a result, if we have 20 data points, then for any observed ν (on the validation set) we expect a much lower generalization error using the perceptron learning algorithm.

Although in general we might expect better generalization with more data, we can sacrifice some of it to incorporate a learning algorithm that alters the π-distribution. This allows us to overcome the restriction to the inefficient exhaustive learning algorithm.

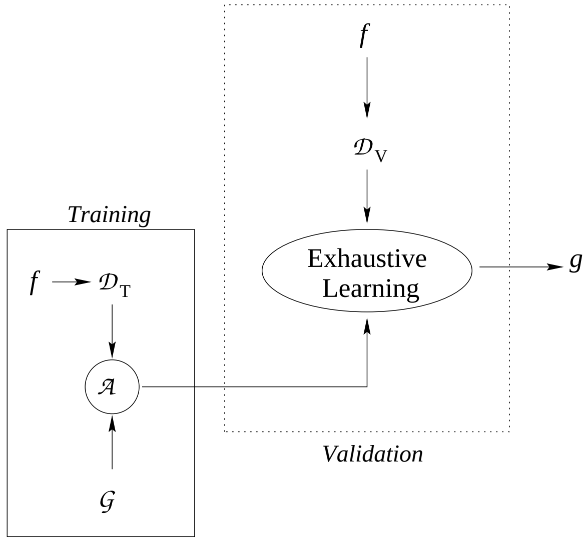


Figure 8: Incorporating a learning algorithm into the bin model framework. The validation step is the learning process used in the bin model analysis. The hypotheses available for the exhaustive learning are now produced by a training step.

6 Conclusion

We have presented a theoretical framework for studying the generalization behavior of learning systems. The bin model analysis parameterizes the generalization curve for a learning process in terms of its π-distribution. We have shown how this can be done for classification problems and certain classes of regression problems. The effects of noise on learning and generalization have been discussed in the context of the bin model. A method for incorporating the analysis with a given learning algorithm has been presented, and it was shown how this can lead to improved generalization performance.

The bin model is intended to be applicable to a very general learning problem, while addressing important issues and providing useful insights into the problem of generalization. While in general the π-distribution is unknown, certain invariants hold true. Using the exhaustive learning algorithm, overfitting is certain not to occur, even in the presence of regular noise. The same holds true for other algorithms in the validation scenario of section 5. Future work directions include further analysis of the effects of noise and characterization of π-distributions for certain common learning models.

A Proofs of Results

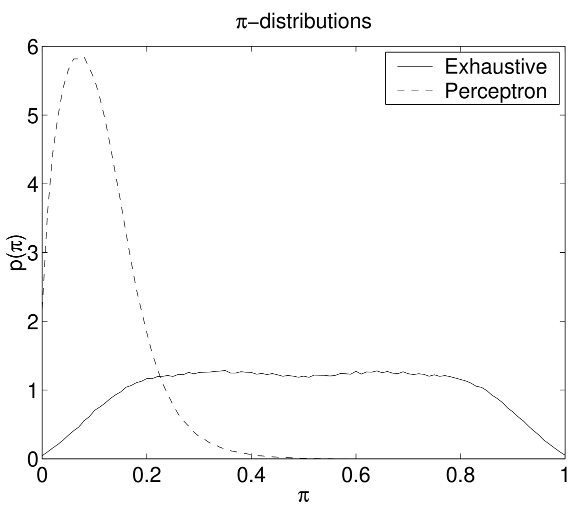
Theorems 2.1, 3.1 and 3.2 assume that the learning model is not strictly degenerate.

Theorem 2.1 *The expected test error $\pi(\nu)$ is monotonically increasing in the empirical error ν.*

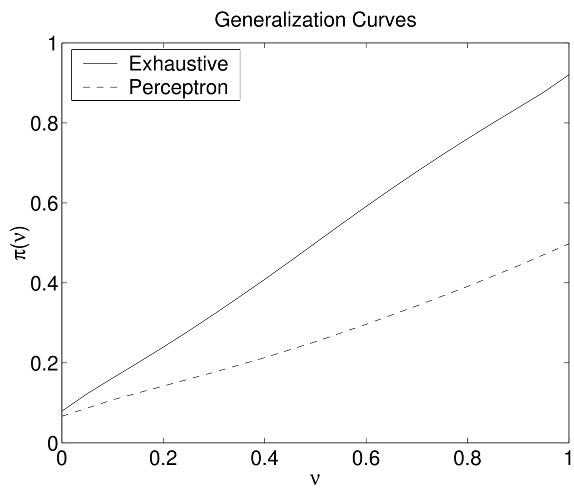
Proof of Theorem 2.1:

This is a special case of Theorem 3.1 with $\tilde{\pi}(g) = \pi(g) \forall g$.

$\blacksquare$



(a)



(b)

Figure 9: (a) Empirically measured π-distributions for a linear perceptron learning model under exhaustive learning, and using the perceptron learning rule with 10 training examples. (b) Generalization curves for a linear perceptron learning model with exhaustive learning and the perceptron learning rule. The exhaustive learning result assumes 20 examples. The generalization error is significantly lower for all in-sample errors for the perceptron rule, even though we spend 10 examples for training, leaving only 10 for validation.

Theorem 3.1 *For any $\tilde{\pi}(g)$, $p_{\mathcal{G}}(\cdot)$ and noise $\varepsilon(x)$, $\tilde{\pi}(\nu)$ is monotonically increasing in ν.*

Proof of Theorem 3.1:

Let $0 \leq k < N$. Then we compare $\tilde{\pi}(\frac{k}{N})$ and $\tilde{\pi}(\frac{k+1}{N})$.

$$\begin{aligned}
&\tilde{\pi}\left(\frac{k+1}{N}\right) - \tilde{\pi}\left(\frac{k}{N}\right) && (76)\\
&= \frac{\int_0^1 s\ \Pr_{\nu|\tilde{\pi}}[\frac{k+1}{N}|s]p_{\tilde{\pi}}(s)ds}{\int_0^1 \Pr_{\nu|\tilde{\pi}}[\frac{k+1}{N}|s]p_{\tilde{\pi}}(s)ds} - \frac{\int_0^1 t\ \Pr_{\nu|\tilde{\pi}}[\frac{k}{N}|t]p_{\tilde{\pi}}(t)dt}{\int_0^1 \Pr_{\nu|\tilde{\pi}}[\frac{k}{N}|t]p_{\tilde{\pi}}(t)dt} && (77)\\
&= \frac{\int s^{k+2}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)ds}{\int s^{k+1}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)ds} - \frac{\int t^{k+1}(1-t)^{N-k}p_{\tilde{\pi}}(t)dt}{\int t^{k}(1-t)^{N-k}p_{\tilde{\pi}}(t)dt} && (78)\\
&= A_0(k)\left[\iint s^{k+2}(1-s)^{N-k-1}t^k(1-t)^{N-k}p_{\tilde{\pi}}(t)p_{\tilde{\pi}}(s)ds\ dt\right.\\
&\quad \left. - \iint t^{k+1}(1-t)^{N-k}s^{k+1}(1-s)^{N-k-1}p_{\tilde{\pi}}(t)p_{\tilde{\pi}}(s)ds\ dt\right] && (79)\\
&= A_0(k)\iint s(s-t)(1-t)s^k t^k(1-s)^{N-k-1}(1-t)^{N-k-1}p_{\tilde{\pi}}(t)p_{\tilde{\pi}}(s)ds\ dt && (80)\\
&= A_0(k)\iint t(t-s)(1-s)t^k s^k(1-t)^{N-k-1}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)p_{\tilde{\pi}}(t)dt\ ds && (81)\\
&= \frac{A_0(k)}{2}\iint (s(s-t)(1-t)+t(t-s)(1-s))t^k s^k(1-t)^{N-k-1}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)p_{\tilde{\pi}}(t)dt\ ds && (82)\\
&= \frac{A_0(k)}{2}\iint (s-t)^2 t^k s^k(1-t)^{N-k-1}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)p_{\tilde{\pi}}(t)dt\ ds && (83)\\
&\geq 0 && (84)
\end{aligned}$$

We use the shorthand $\Pr_{\nu|\tilde{\pi}}[a|b]$ for the conditional probability $\Pr[\nu = a|\tilde{\pi} = b]$. In (79), the factor

$$A_0(k) = \left(\int s^{k+1}(1-s)^{N-k-1}p_{\tilde{\pi}}(s)ds \int t^k(1-t)^{N-k}p_{\tilde{\pi}}(t)dt\right)^{-1}$$

is positive (since the integrands are nonnegative) and finite (since $\mathcal{G}$ is not strictly degenerate, hence $\exists\ 0 < \tilde{\pi}_0 < 1$ with $p_{\tilde{\pi}}(\tilde{\pi}_0) > 0$) for all k. (81) is (80) rewritten with a change of variables, and (82) is obtained by summing (81) and (80). Thus $\tilde{\pi}(0) \leq \tilde{\pi}(\frac{1}{N}) \leq \tilde{\pi}(\frac{2}{N}) \leq \ldots \leq \tilde{\pi}(1)$ completing the proof.

■

Theorem 3.2 *If $\varepsilon(x)$ is regular and $\tilde{\pi}(\pi)$ is monotonically increasing (decreasing) in π then for any $p_\pi(\cdot)$, $\pi(\nu)$ is monotonically increasing (decreasing) in ν.*

Proof of Theorem 3.2:

The proof is similar to that of Theorem 3.1. Let $0 \leq k < N$. We look at $\pi\left(\frac{k+1}{N}\right) - \pi\left(\frac{k}{N}\right)$.

$$\begin{aligned}
&\pi(\frac{k+1}{N}) - \pi(\frac{k}{N}) && (85)\\
&= \frac{\int_0^1 s\ \Pr_{\nu|\pi}[\frac{k+1}{N}|s]p_\pi(s)ds}{\int_0^1 \Pr_{\nu|\pi}[\frac{k+1}{N}|s]p_\pi(s)ds} - \frac{\int_0^1 t\ \Pr_{\nu|\pi}[\frac{k}{N}|t]p_\pi(t)dt}{\int_0^1 \Pr_{\nu|\pi}[\frac{k}{N}|t]p_\pi(t)dt} && (86)
\end{aligned}$$

$$= \frac{\int s\ \tilde{\pi}(s)^{k+1}(1-\tilde{\pi}(s))^{N-k-1}p_\pi(s)ds}{\int \tilde{\pi}(s)^{k+1}(1-\tilde{\pi}(s))^{N-k-1}p_\pi(s)ds} - \frac{\int t\ \tilde{\pi}(t)^{k}(1-\tilde{\pi}(t))^{N-k}p_\pi(t)dt}{\int \tilde{\pi}(t)^{k}(1-\tilde{\pi}(t))^{N-k}p_\pi(t)dt} \tag{87}$$

$$\begin{aligned} = & A_1(k)\left[\int\int s\ \tilde{\pi}(s)^{k+1}(1-\tilde{\pi}(s))^{N-k-1}\tilde{\pi}(t)^k(1-\tilde{\pi}(t))^{N-k}p_\pi(t)p_\pi(s)ds\ dt \right. \\ & \left. -\int\int t\ \tilde{\pi}(t)^k(1-\tilde{\pi}(t))^{N-k}\tilde{\pi}(s)^{k+1}(1-\tilde{\pi}(s))^{N-k-1}p_\pi(t)p_\pi(s)ds\ dt\right] \end{aligned} \tag{88}$$

$$= A_1(k)\int\int (s-t)\tilde{\pi}(s)(1-\tilde{\pi}(t))\tilde{\pi}(s)^k\tilde{\pi}(t)^k(1-\tilde{\pi}(s))^{N-k-1}(1-\tilde{\pi}(t))^{N-k-1}p_\pi(t)p_\pi(s)ds\ dt \tag{89}$$

$$= A_1(k)\int\int (t-s)\tilde{\pi}(t)(1-\tilde{\pi}(s))\tilde{\pi}(s)^k\tilde{\pi}(t)^k(1-\tilde{\pi}(s))^{N-k-1}(1-\tilde{\pi}(t))^{N-k-1}p_\pi(t)p_\pi(s)ds\ dt \tag{90}$$

$$= \frac{A_1(k)}{2}\int\int (s-t)(\tilde{\pi}(s)-\tilde{\pi}(t))\tilde{\pi}(t)^k\tilde{\pi}(s)^k(1-\tilde{\pi}(t))^{N-k-1}(1-\tilde{\pi}(s))^{N-k-1}p_\pi(s)p_\pi(t)dt\ ds \tag{91}$$

In (88) the factor

$$A_1(k) = \left(\int \tilde{\pi}(s)^{k+1}(1-\tilde{\pi}(s))^{N-k-1}p_\pi(s)ds\int \tilde{\pi}(t)^{k}(1-\tilde{\pi}(t))^{N-k}p_\pi(t)dt\right)^{-1}$$

is positive and finite for all k, and the steps are essentially the same as those in the proof of Theorem 3.1.

If $\tilde{\pi}(\pi)$ is monotonically increasing in π, then $(s-t)(\tilde{\pi}(s)-\tilde{\pi}(t)) \geq 0\ \forall s,t$, the integrand in (91) is always nonnegative, and hence $\pi(\nu)$ is monotonically increasing in ν. Likewise, if $\tilde{\pi}(\pi)$ is monotonically decreasing in π, then the integrand in (91) is always nonpositive and $\pi(\nu)$ is monotonically decreasing in ν.

■

Theorem 4.1 *If $p_{\nu|\pi}$ is non-zero, continuous and differentiable on $(\nu,\pi)\in\mathcal{E}^2$ then $\pi(\nu)$ is monotonically increasing in ν for any p_π iff $(\frac{\partial}{\partial\nu}p_{\nu|\pi})/p_{\nu|\pi}$ is monotonically increasing in π.*

Proof of Theorem 4.1:

$$\pi(\nu) = \frac{\int sp_{\nu|\pi}(\nu|s)p_\pi(s)ds}{\int p_{\nu|\pi}(\nu|s)p_\pi(s)ds} \tag{92}$$

$$\frac{d\pi(\nu)}{d\nu} = \frac{\int s\frac{d}{d\nu}p_{\nu|\pi}(\nu|s)p_\pi(s)ds\int p_{\nu|\pi}(\nu|t)p_\pi(t)dt - \int sp_{\nu|\pi}(\nu|s)p_\pi(s)ds\int\frac{d}{d\nu}p_{\nu|\pi}(\nu|t)p_\pi(t)ds}{\left(\int p_{\nu|\pi}(\nu|s)p_\pi(s)ds\right)^2} \tag{93}$$

$$\begin{aligned} = & A_2(\nu)\left[\int\int s\frac{d}{d\nu}p_{\nu|\pi}(\nu|s)p_{\nu|\pi}(\nu|t)p_\pi(s)p_\pi(t)ds\ dt\right. \\ & \left. -\int\int sp_{\nu|\pi}(\nu|s)\frac{d}{d\nu}p_{\nu|\pi}(\nu|t)p_\pi(s)p_\pi(t)ds\ dt\right] \end{aligned} \tag{94}$$

$$= A_2(\nu)\int\int s\left(\frac{d}{d\nu}p_{\nu|\pi}(\nu|s)p_{\nu|\pi}(\nu|t) - p_{\nu|\pi}(\nu|s)\frac{d}{d\nu}p_{\nu|\pi}(\nu|t)\right)p_\pi(s)p_\pi(t)ds\ dt \tag{95}$$

$$= A_2(\nu)\int\int t\left(\frac{d}{d\nu}p_{\nu|\pi}(\nu|t)p_{\nu|\pi}(\nu|s) - p_{\nu|\pi}(\nu|t)\frac{d}{d\nu}p_{\nu|\pi}(\nu|s)\right)p_\pi(s)p_\pi(t)ds\ dt \tag{96}$$

$$= \frac{A_2(\nu)}{2}\int\int (s-t)\left(\frac{d}{d\nu}p_{\nu|\pi}(\nu|s)p_{\nu|\pi}(\nu|t) - p_{\nu|\pi}(\nu|s)\frac{d}{d\nu}p_{\nu|\pi}(\nu|t)\right)p_\pi(s)p_\pi(t)ds\ dt \tag{97}$$

In (94) the factor

$$A_2(\nu) = \left(\int p_{\nu|\pi} p_\pi(x) ds \right)^{-2}$$

is positive and finite for all ν and (96) is (95) rewritten with a change of variables. Combining (95) and (96) gives (97).

First we show sufficiency. For any $s > t$, the integrand in (97) is nonnegative if

$$\frac{d}{d\nu} p_{\nu|\pi}(\nu|s) p_{\nu|\pi}(\nu|t) \geq p_{\nu|\pi}(\nu|s) \frac{d}{d\nu} p_{\nu|\pi}(\nu|t) \tag{98}$$

$$\frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu|s)}{p_{\nu|\pi}(\nu|s)} \geq \frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu|t)}{p_{\nu|\pi}(\nu|t)} \tag{99}$$

$$\tag{100}$$

Since $p_{\nu|\pi}$ is assumed to be continuous, this is equivalent to increasing monotonicity of $(\frac{d}{d\nu} p_{\nu|\pi})/p_{\nu|\pi}$ in π.

To show necessity, suppose that $(\frac{d}{d\nu} p_{\nu|\pi})/p_{\nu|\pi}$ is not monotonically increasing in π. Then $\exists \pi_1, \pi_2, \nu_0$ such that

$$\pi_1 > \pi_2 \tag{101}$$

$$\frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu_0|\pi_1)}{p_{\nu|\pi}(\nu_0|\pi_1)} < \frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu_0|\pi_2)}{p_{\nu|\pi}(\nu_0|\pi_2)} \tag{102}$$

$$\tag{103}$$

Then letting $p_\pi(\pi) = \frac{1}{2}(\delta(\pi - \pi_1) + \delta(\pi - \pi_2))$, we get

$$\frac{d\pi(\nu)}{d\nu} \propto \frac{1}{2} \int\int (s-t)(\frac{d}{d\nu} p_{\nu|\pi}(\nu|s) p_{\nu|\pi}(\nu|t) - p_{\nu|\pi}(\nu|s) \frac{d}{d\nu} p_{\nu|\pi}(\nu|t)) p_\pi(s) p_\pi(t) ds\ dt \tag{104}$$

$$= \frac{1}{4}(\pi_1 - \pi_2)(\frac{d}{d\nu} p_{\nu|\pi}(\nu|\pi_1) p_{\nu|\pi}(\nu|\pi_2) - \frac{d}{d\nu} p_{\nu|\pi}(\nu|\pi_2) p_{\nu|\pi}(\nu|\pi_1)) \tag{105}$$

$$\tag{106}$$

$$= \frac{1}{4}(\pi_1 - \pi_2) \left(\frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu|\pi_1)}{p_{\nu|\pi}(\nu|\pi_1)} - \frac{\frac{d}{d\nu} p_{\nu|\pi}(\nu|\pi_2)}{p_{\nu|\pi}(\nu|\pi_2)} \right) p_{\nu|\pi}(\nu|\pi_1) p_{\nu|\pi}(\nu|\pi_2) \tag{107}$$

$$\left. \frac{d\pi}{d\nu} \right|_{\nu=\nu_0} < 0 \tag{108}$$

Thus $\pi(\nu)$ fails to be monotonically increasing at ν_0 and the proof is complete.

■

Theorem 4.2 *If $\sigma(g) = \sigma(\pi(g))$ $\forall g \in \mathcal{G}$, then in the large data limit $N \to \infty$, $\pi(\nu)$ is monotonically increasing in ν for all p_π iff $\frac{e_{max} - \pi_1}{e_{max} - \pi_2} \leq \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \leq \frac{\pi_1 - e_{min}}{\pi_2 - e_{min}}$ whenever $\pi_1 > \pi_2$.*

Proof of Theorem 4.2:

In the large data limit

$$p_{\nu|\pi}(\nu|\pi) \quad \propto \quad \exp(-\frac{N(\nu-\pi)^2}{2\sigma(\pi)^2}) \tag{109}$$

$$\frac{d}{d\nu}p_{\nu|\pi}(\nu|\pi) \quad = \quad \frac{N(\pi-\nu)}{\sigma(\pi)^2}p_{\nu|\pi}(\nu|\pi) \tag{110}$$

Theorem 4.1 tells us that increasing monotonicity of $\pi(\nu)$ in ν for any p_π is equivalent to increasing monotonicity of $(\frac{d}{d\nu}p_{\nu|\pi})/p_{\nu|\pi}$ in π.

$$\frac{\frac{d}{d\nu}p_{\nu|\pi}(\nu|\pi)}{p_{\nu|\pi}(\nu|\pi)} = \frac{N(\pi-\nu)}{\sigma(\pi)^2} \tag{111}$$

Thus we want, $\forall \nu \in \mathcal{E}$, if $\pi_1 > \pi_2$, then

$$\frac{N(\pi_1-\nu)}{\sigma(\pi_1)^2} \geq \frac{N(\pi_2-\nu)}{\sigma(\pi_2)^2} \tag{112}$$

We consider three possible cases. If $e_{min} \leq \nu < \pi_2$, then we have monotonicity iff

$$\frac{N(\pi_1-\nu)}{\sigma(\pi_1)^2} \quad \geq \quad \frac{N(\pi_2-\nu)}{\sigma(\pi_2)^2} \tag{113}$$

$$\frac{\pi_1-\nu}{\pi_2-\nu} \quad \geq \quad \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \tag{114}$$

The worst case is $\nu = e_{min}$, so it is sufficient that

$$\frac{\pi_1-e_{min}}{\pi_2-e_{min}} \geq \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \tag{115}$$

If $\pi_2 \leq \nu < \pi_1$, then

$$\frac{N(\pi_1-\nu)}{\sigma(\pi_1)^2} > 0 \geq \frac{N(\pi_2-\nu)}{\sigma(\pi_2)^2} \tag{116}$$

and we always have monotonicity.

If $\pi_1 \leq \nu \leq e_{max}$, then we have monotonicity iff

$$\frac{N(\pi_1-\nu)}{\sigma(\pi_1)^2} \quad \geq \quad \frac{N(\pi_2-\nu)}{\sigma(\pi_2)^2} \tag{117}$$

$$\frac{\nu-\pi_1}{\nu-\pi_2} \quad \leq \quad \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \tag{118}$$

The worst case is $\nu = e_{max}$, so it is sufficient that

$$\frac{e_{max}-\pi_1}{e_{max}-\pi_2} \leq \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \tag{119}$$

(115) and (119) tell us that $(\frac{d}{d\nu}p_{\nu|\pi})/p_{\nu|\pi}$ is monotonically increasing in π for all ν iff $\frac{e_{max}-\pi_1}{e_{max}-\pi_2} \leq \frac{\sigma(\pi_1)^2}{\sigma(\pi_2)^2} \leq \frac{\pi_1-e_{min}}{\pi_2-e_{min}}$ whenever $\pi_1 > \pi_2$, proving the theorem.

■

Proposition A.1 *For a regression problem, assume a squared error measure and assume additive zero-mean Gaussian noise.*

$$\begin{aligned}
e(g,x) &= (g(x)-f(x))^2 && (120)\\
\tilde{f}(x) &= f(x)+\eta && (121)\\
\eta &\sim \mathcal{N}(0,\sigma_\eta^2) && (122)\\
\tilde{e}(g,x) &= (g(x)-\tilde{f}(x))^2 && (123)
\end{aligned}$$

If the noiseless errors e are distributed like $p_{e|g}$ and the noisy errors $\tilde{e}$ are distributed like $p_{\tilde{e}|g}$, then $p_{e|g}$ and $p_{\tilde{e}|g}$ are related by

$$\widehat{p_{\tilde{e}|g}}(\omega) = \frac{\widehat{p_{e|g}}\left(\frac{\omega}{1+2i\sigma_\eta^2\omega}\right)}{\sqrt{1+2i\sigma_\eta^2\omega}} \qquad (124)$$

Proof of Proposition A.1:

Let $s(x)=g(x)-f(x)$.

$$\begin{aligned}
e(g,x) &= s^2(x) && (125)\\
\tilde{e}(g,x) &= (s(x)-\eta)^2 && (126)\\
\widehat{p_{\tilde{e}|g}}(\omega) &= E_{\tilde{e}}[\exp(-i\omega\tilde{e})|g] && (127)\\
&= E_{x,\eta}[\exp(-i\omega\tilde{e}(x,g))] && (128)\\
&= \iint \exp\left(-i\omega\,(s(x)-\eta)^2\right) p_{e|g}(s(x)^2)p_\eta(\eta)d\eta\ ds^2(x) && (129)\\
&= \frac{1}{\sqrt{2\pi}\sigma_\eta}\int \exp\left(-i\omega s(x)^2\right) p_{e|g}(s(x)^2)\cdot\\
&\quad \int \exp\left(-i\omega\left(\eta^2-2s(x)\eta\right)\right)\exp\left(-\frac{\eta^2}{2\sigma_\eta^2}\right)d\eta\ ds^2(x) && (130)\\
&= \frac{1}{\sqrt{2\pi}\sigma_\eta}\int \exp\left(-i\omega s(x)^2\right) p_{e|g}(s(x)^2)\cdot\\
&\quad \int \exp\left(-\left(i\omega+\frac{1}{2\sigma_\eta^2}\right)\left(\eta^2-\frac{4i\omega\sigma_\eta^2 s(x)\eta}{2i\omega\sigma_\eta^2+1}\right)\right)d\eta\ ds^2(x) && (131)\\
&= \frac{1}{\sqrt{2\pi}\sigma_\eta}\int \exp\left(-i\omega s(x)^2\right) p_{e|g}(s(x)^2)\exp\left(-\frac{2\omega^2\sigma_\eta^2 s(x)^2}{2i\omega\sigma_\eta^2+1}\right)\cdot\\
&\quad \int \exp\left(-\left(i\omega+\frac{1}{2\sigma_\eta^2}\right)\left(\eta-\frac{2i\omega\sigma_\eta^2 s(x)}{2i\omega\sigma_\eta^2+1}\right)^2\right)d\eta\ ds^2(x) && (132)\\
&= \frac{1}{\sqrt{2\pi}\sigma_\eta}\int \exp\left(-is(x)^2\left(\omega-\frac{2i\omega^2\sigma_\eta^2}{2i\omega\sigma_\eta^2+1}\right)\right)p_{e|g}(s(x)^2)\sqrt{\frac{2\pi\sigma_\eta^2}{2i\omega\sigma_\eta^2+1}}ds^2(x) && (133)\\
&= \frac{1}{\sqrt{2i\omega\sigma_\eta^2+1}}\int \exp\left(-i\left(\frac{\omega}{2i\omega\sigma_\eta^2+1}\right)s(x)^2\right)p_{e|g}(s(x)^2)ds^2(x) && (134)
\end{aligned}$$

$$= \frac{E_x[\exp\left(-i\left(\frac{\omega}{1+2i\sigma_\eta^2\omega}\right)e(g|x)\right)]}{\sqrt{1+2i\sigma_\eta^2\omega}} \quad (135)$$

$$= \frac{\widehat{p_{e|g}}\left(\frac{\omega}{1+2i\sigma_\eta^2\omega}\right)}{\sqrt{1+2i\sigma_\eta^2\omega}} \quad (136)$$

■

References

[1] Y. Abu-Mostafa. The vapnik-chervonenkis dimension: Information versus complexity in learning. *Neural Computation*, 1:312–317, 1989.

[2] Y. Abu-Mostafa and X. Song. Bin model for neural networks. In *Proceedings of ICONIP'96*, pages 169–173, Hong Kong, 1996.

[3] H. Akaike. Statistical predictor identification. *Annals of the Institute of Statistical Mathematics*, 22:203, 1970.

[4] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[5] Z. Cataltepe et al. No free lunch for early stopping. *Neural Computation*, 11:995–1009, 1999.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecomunications. Wiley, 1991.

[7] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. J. Wiley & Sons, 1973.

[8] S. Haykin. *Neural Networks*. Macmillan College Publishing Company, New York, 1994.

[9] S. Kirkpatrick et al. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[10] M. Magdon-Ismail et al. Financial markets: very noisy information processing. *Proceedings of the IEEE*, 86(11):2184–2195, 1998.

[11] J. E. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 4, pages 847–854, 1991.

[12] D. Rumelhart et al. Learning internal representations by error propagation. In D. Rumelhart and J. McCleland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362. MIT Press, Cambridge, MA, 1986.

[13] D. Schwartz et al. Exhaustive learning. *Neural Computation*, 2(2):374–385, 1990.

[14] S. Solla. Supervised learning: A theoretical framework. In M. Casdagli and S. Eubank, editors, *Nonlinear Modelling and Forecasting*, pages 25–38, 1992.

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

[16] V. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc., New York, 1998.

[17] V. N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[18] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.