A Generalization Model for Learning Systems

Yaser S. Abu-Mostafa
yaser@caltech.edu
Department of Electrical Engineering
and Department of Computer Science
California Institute of Technology
136-93 Pasadena, CA, 91125

Xubo Song *
xubosong@fire.work.caltech.edu
Department of Electrical Engineering
California Institute of Technology
136-93 Pasadena, CA, 91125

Alexander Nicholson
zander@foot.work.caltech.edu
Department of Computer Science
California Institute of Technology
136-93 Pasadena, CA, 91125

Malik Magdon-Ismail
malik@wild.work.caltech.edu
Department of Electrical Engineering
California Institute of Technology
136-93 Pasadena, CA, 91125

Abstract

We propose a novel theoretical framework, called the Bin Model, for learning and generalization. Using the Bin Model, we derive a closed form for generalization that estimates the out of sample performance in terms of the in sample performance. We show that the expected generalization for a given learning problem can be characterized by a single probability distribution. We address the problems of over-fitting and over-learning, and characterize conditions under which they do not appear. The Bin Model is a very powerful model which is simple enough for mathematical manipulation and general enough to address and give insights to many important issues in learning.

1 Introduction

In machine learning, the task of interest is to discover an underlying mapping from a set of input-output data. In addition, one one would like to generalize from the training data set to never before seen inputs. It is extremely desirable to have a tool that can estimate the out-of-sample performance in terms of the in-sample performance.

*To whom correspondence should be addressed.

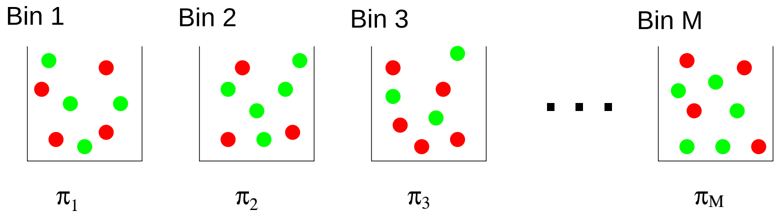


Figure 1: *The Bin Model: We have a set of M bins, each containing red and green marbles. The frequency of red marbles in bin i is denoted by π_i.*

For regression problems, Akaike's FPE (Final Prediction Error) [2] and Moody's GPE (Generalized Prediction Error) [3] indicate that the expected out-of-sample error equals the expected in-sample error plus a term which depends on the noise variance, model complexity and the size of training set. These results are all asymptotic and are approximations. For classification problems, the VC analysis [4], bounds the probability that the worst case deviation of the in-sample error from the out-of-sample error exceeds a fixed quantity.

It would be nice if we could model the generalization behavior for arbitrary nonlinear models, nonlinear target functions and noisy data sets, which are the most challenging theoretically and the most relevant practically. We derive an analytic solution for such a scenario for the expected out-of-sample error in terms of the in-sample error. The key is to represent to learning problem using a simple model, which we call the **Bin Model**, that can handle the general nonlinear, noisy case.

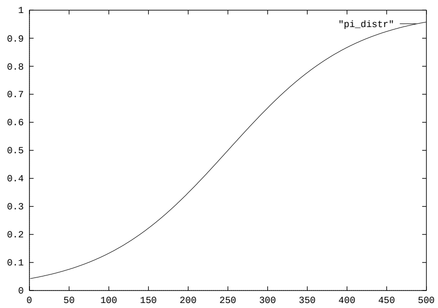
This paper is organized as follows: In section 2, we present the framework of the Bin Model. Section 3 provides the formulation of generalization using the Bin Model. We derive a closed form estimation of the out-of-sample error in terms of the in-sample error. In section 4, we discuss the important conclusion derived from the Bin Model that characterizes the conditions for over-fitting and over-learning.

2 The Bin Model

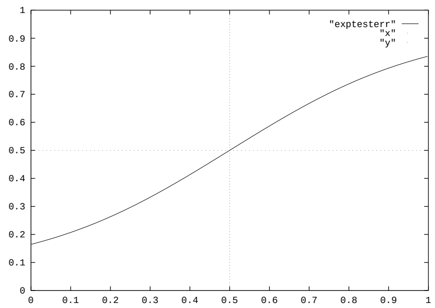
We consider target functions of the form $f : X \to \{0, 1\}$. A candidate function $g : X \to \{0, 1\}$ is characterized by a parameter π, which is the probability that $g(x)$ disagrees with $f(x)$ on a point $\mathbf{x}$ picked from the input space $\mathbf{X}$ according to the possibly unknown input distribution $\Omega(\mathbf{x})$.

$$\pi \triangleq Pr\left[g \neq f\right] = \int \left[g(x) - f(x)\right]^2 \Omega(x) dx \tag{1}$$

A single hypothesis function can be visualized as one bin that contains red and green marbles, with red marbles representing input points on which the hypothesis disagrees with the target, and green marbles representing those on which they agree. Then the fraction of red marbles in the bin is π. When we draw an example x from the input space and compare $f(x)$ with $g(x)$, we are virtually drawing a marble from the bin and checking its color. We can treat a sample as a Bernoulli trial with probability π of error (red marble) and probability $1 - \pi$ of success (green marble). When we have a sample of size N, we have N i.i.d. Bernoulli trials. We denote by ν the fraction of red marbles (frequency of error) in the sample. The Law of Large Numbers tells us that as $N \to \infty$, random variable ν approaches its mean π. That



(a)



(b)

Figure 2: (a)*An example π-distribution for $M = 500$ (note that the x-axis indicates bin index). In this case the hypothesis that best approximates the target function makes approximately 5% error.* (b) *The expected test error as a function of training error ν for the π-distribution in (a).*

is to say, the frequency of error will be a very good estimate of the probability of error for large sample size N.

In reality, instead of one single candidate function, we are actually looking at a set G of candidate functions (the learning model). Each single function has its own probability of error. We have an array of $\pi_1, \pi_2, ..., \pi_M$, where $M = |G|$ is the number of functions in G. This array of π's leads to a π-distribution. The π-distribution indicates the suitability of the learning model for approximating the target function. The smaller the probability of error π and the more the functions that have small π, the better suited this learning model is for the approximation of f.

Figure 2 (a) is an example of π-distribution in which we have some "best" functions with error probability around 0.05, some "worst" ones around 0.95, and a lot of mediocre ones that lie in between. (The π's are ordered in an increasing way for illustrative purposes.)

It is worth noticing that the π-distribution and M are all the information that we need to specify a bin model. We do not need the detailed information about the specific forms the target and the candidate functions assume. Nor do we need to know anything about the inputs and input distribution. The bin model abstracts relevant quantities from the target function and the learning model. It can be applied to arbitrary, nonlinear target function and arbitrary, nonlinear learning model. This makes it possible to model learning processes in a very simple way and without loss of generality.

3 Generalization

In learning, what we have is a finite and often noisy data set. We choose a hypothesis from the learning model that fits the data well according to some criterion. We have access only to the training error, and would like to know if the in sample performance generalizes to out of sample points. This is the matter of generalization. We define the expected out of sample error given the in sample error as the measure of generalization, whose analytic expression can be derived from the bin model.

The learning game of interest is the following. We have a learning model and a training set. One candidate is randomly picked from the learning model. and its in sample error is measured. This is equivalent to drawing a sample of size N from this bin. An error frequency is obtained. If it matches some predetermined level of training error ν, we keep this function as our approximation to the target function f, otherwise we put it back. This reminds us of what we do in learning – we stop at a certain training error, and keep the function that achieves this training error as our hypothesis. Among the ones that achieve the same training error ν, a candidate is randomly chosen as the final candidate. We would like to know the expected out of sample error of the final candidate, that is $E[\pi|\nu]$.[1]

The set $\{\pi_i\}$ induces a probability distribution p on π. Randomly choosing a candidate function from G gives us a g with probability of error which can be assumed to be randomly chosen from $p(\pi)$. In this case, we have

$$\pi(\nu) \triangleq E[\pi \mid \nu] = \int \pi p(\pi \mid \nu) d\pi \tag{2}$$

$$= \frac{\int \pi p(\pi, \nu) d\pi}{\int p(\pi, \nu) d\pi} \tag{3}$$

$$= \frac{\int \pi p(\pi) \pi^{N\nu} (1-\pi)^{N(1-\nu)} d\pi}{\int p(\pi) \pi^{N\nu} (1-\pi)^{N(1-\nu)} d\pi} \tag{4}$$

where

$$p(\pi, \nu) = p(\pi) \binom{N}{N\nu} \pi^{N\nu} (1-\pi)^{N(1-\nu)}$$

Figure 2 (b) is the generalization curve based on the π-distribution given in 2 (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 a more optimistic estimation of the corresponding out of sample error.

In the No-Free-Lunch theorems [5], Wolpert has shown that, in general, *some* assumption must be made about the target function to make any claims about generalization. With the Bin Model analysis, we have shown that the information of interest (enough to give us expected generalization error) is contained in a single π-distribution. While the details of the π-distribution are still dependendent on the target function, we can gain insights from invariant properties of the analysis. For example, the addition of noise in a classification problem results in a simple transformation of $p(\pi)$ [1], and so the comparative effects of noise can be studied relatively simply.

4 Unbiased Optimization

When attempting to learn a function from a data set by minimizing the in sample error, it is not uncommon to observe over-fitting – the hypothesis approximates the training data set well, but is far from the target function on other inputs. The learning algorithm may select more and more complex functions that fit the training data better, but that also fit any idiosyncrasies or noise in the training set. As a result, the hypothesis approximates the training data set well, but can't generalize on out of sample data.

[1] We use $E[\cdot]$ to denote expectations.

With the idea that over-fitting is usually accompanied by using overly complex models, and that the data captures the main structure of the function, one might expect that a simpler model that performs well must be close in general form to the target and therefore would have good generalization. This notion is embodied in *Occam's Razor*, which prescribes a preference for simpler explanations. Thus efforts to improve generalization have generally involved simplifying the class of functions in the learning model, often at the "cost" of an increased raw error on the training set. Among these methods are early stopping, regularization, and Structural Risk Minimization.

In the Bin Model framework, however, the over-fitting phenomenon does not appear. It can be shown that $\pi(\nu)$ is monotonically increasing with respect to ν, that is $\frac{d\pi(\nu)}{d\nu} > 0$ for all ν. This implies that smaller in sample error always corresponds to smaller out of sample error in expectation. Thus there will be no over-fitting and we can always benefit from getting smaller in sample error. Furthermore, no explicit measure of the model complexity appears in the calculation of $\pi(\nu)$. The generalization ability is completely captured by $p(\pi)$, which may be independent of traditional complexity measures.

Fundamental to the Bin Model analysis is the random selection of a hypothesis – we assume that candidate functions are chosen randomly from G. The choice is not subject to a controlled tour of parameter space. We use the term **unbiased optimization** to refer to this learning algorithm. For continuously parameterized learning models, we can only approximately implement unbiased optimization in practice by Monte Carlo simulations.

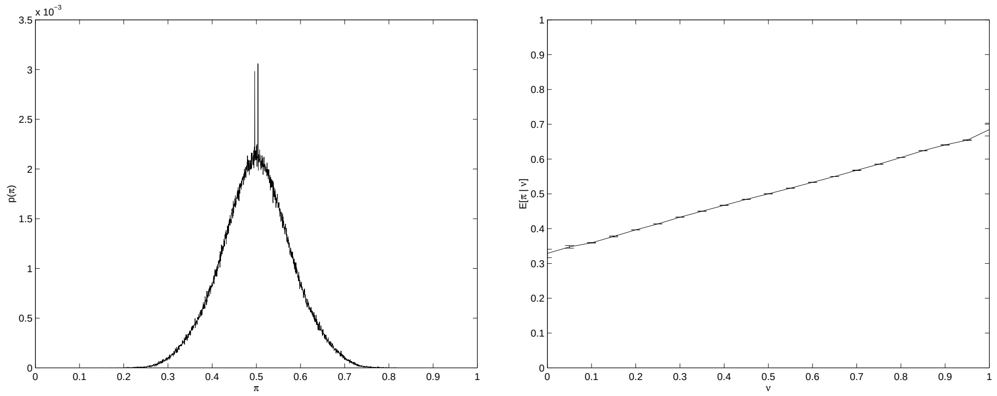
Figure 3 is an example of the generalization behavior under unbiased optimization. A histogram approximating $p(\pi)$ is shown, along with the empirical generalization function $\pi(\nu)$. $p(\pi)$ is estimated by randomly sampling hypotheses from the learning model and estimating π by the test error on a large data set. We can see that the expected generalization is poor for such a small data set (even for $\nu = 0$ we can only expect $\pi \sim 0.3$), and that $\pi(\nu)$ is indeed monotonic – we obtain the best out-of-sample error by finding the minimum training error (over-fitting is not observed).

5 Conclusion

In this paper we present the Bin Model, which provides a framework for analyzing generalization in a general classification learning problem. Within this framework, we can derive properties that The Bin Model abstracts the relevant quantities from a learning problem by parameterizing each hypothesis function with a single value, π. The expected generalization error is shown to depend only on the π-distribution for the learning problem. Within the Bin Model framework, we can derive properties that hold for any $p(\pi)$ (for example, the lack of overfitting with unbiased optimization), and therefore that hold for any learning model, target function and input distribution. Thus the Bin Model can provide a tool for further study of distribution independent properties. Work in progress includes the extension of the model to noisy problems, regression problems and the identification of relative features of $p(\pi)$ that characterize commonly encountered learning problems.

Acknowledgements

We would like to thank Dr. Amir Atiya, Dr. Zehra Cataltepe and Dr. Joe Sill for numerous helpful discussions.



(a) (b)

Figure 3: (a) *The* π *distribution for a noisy learning problem. The target function is* $f(\mathbf{x}) = sign(sin(\frac{x_1}{2})sin(\frac{x_2}{4}))$ *and the learning model is 2-10-1 neural network with hyperbolic tangent transfer function and linear output. The training set is of size 20. The noise in the training set is realized by randomly flipping the sign of the output with probability 0.1. The input distribution is uniform in [-2* π*, 2* π*].* (b) *The expected test error as a function of training error* ν *for the* π *distribution in (a) when using unbiased optimization.*

References

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

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

[3] 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.

[4] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer–Verlag, 1995.

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