Author and Co-author Contact Information

Ambuj Tewari
439 West Hall
1085 South University
Ann Arbor, MI 48109-1107
USA
tewaria@umich.edu
734-763-3519

Peter L. Bartlett
387 Soda Hall #1776
Berkeley, CA 94720-1776
USA
bartlett@cs.berkeley.edu
510-642-7780

Abstract

We present an overview of learning theory including its statistical and computational aspects. We start by giving a probabilistic formulation of learning problems including classification and regression where the learner knows nothing about the probability distribution generating the data. We then consider the principle of empirical risk minimization (ERM) that chooses a function from a given class based on its performance on the observed data. Learning guarantees for ERM are shown to be intimately connected with the uniform law of large numbers. A uniform law of large numbers ensures that empirical means converge to true expectations uniformly over a function class. Tools such as Rademacher complexity, covering numbers, and combinatorial dimensions known as the Vapnik-Chervonenkis (VC) and fat shattering dimensions are introduced. After considering the case of learning using a fixed function class, we turn to the problem of model selection: how to choose a function class from a family based on available data? We also survey alternative techniques for studying generalization ability of learning algorithms including sample compression, algorithmic stability, and the PAC-Bayesian theorem. After dealing with statistical issues, we study computational models of learning such the basic and agnostic PAC learning models, the statistical query, and the mistake bound model. Finally, we point out extensions of the basic theory beyond the probabilistic setting of a learner passively learning a single task from independent and identically distributed samples.

Glossary

AdaBoost Acronym for Adaptive Boosting, a popular boosting algorithm.

Bayes classifier In classification problems with 0-1 loss, a classifier with the least risk.

boosting The process of converting a weak learning algorithm (that is, one that outputs classifiers whose performance is just a little bit better than random guessing) into a strong one.

concentration inequality A probabilistic inequality stating that some random variable will not deviate too far from its mean or median.

excess risk The difference between the risk of a given function and the minimum possible risk over a function class.

empirical risk Same as risk except that the expectation under the unknown data distribution is replaced with an empirical average over the samples observed.

empirical risk minimization A learning rule that minimizes the empirical risk to choose a prediction function.

ERM Empirical Risk Minimization

fat shattering dimension A scale sensitive generalization of the VC dimension.

generalization error bound An upper bound stating that the risk of a learning rule is not going to differ too much from its empirical risk.

kernel A symmetric positive semidefinite function.

Littlestone dimension A combinatorial quantity associated with a class of binary valued functions. Plays a fundamental role in determining worst case bounds in the online mistake bound model.

loss function A function used to measure the quality of a prediction given the true label.

mistake bound model A model of learning that, unlike the PAC model, does not assume anything about the way data is generated.

model selection The task of selecting an appropriate statistical model from a family based on available data.

PAC model A computational model for analyzing the resources (time, memory, samples) required for performing learning tasks. Originally defined for the task of binary classification using the 0-1 loss. The acronym PAC stands for "Probably Approximately Correct".

Rademacher complexity A quantity associated with a function class that measures its capacity to overfit by measuring how well the function class can fit randomly generated ± 1 signs.

regression function In regression problems with the squares loss, the prediction function with the least risk.

reproducing kernel Hilbert space A Hilbert space of functions for which there exists a kernel with the reproducing property: point evaluations of functions can be written as linear functionals.

risk The expected loss of a prediction function under the underlying unknown distribution generating input, output pair.

sample complexity The number of samples requires to achieve a given level of accuracy (usually with high probability).

sample compression An approach to obtaining generalization error bounds for algorithms whose output can be reconstructed from a small number of examples belonging to the original sample.

stability The property of a learning algorithm that ensures that its output does not change much if the training data set is changed slightly.

supervised learning The task of learning a functional dependence between an input space and an output or label space using given examples of input, output pairs.

universal consistency The property of a learning rule to converge to the minimum possible risk as the number of samples grows to infinity.

Vapnik-Chervonenkis dimension A combinatorial quantity associated with a binary valued function class that serves as a measure of the capacity of the function class to overfit the data.

VC dimension Vapnik-Chervonenkis dimension

Contents

1 Introduction **5**

2 Probabilistic Formulation of Learning Problems **6**
- 2.1 Loss Function and Risk . . . 7
- 2.2 Universal Consistency . . . 8
- 2.3 Learnability with Respect to a Fixed Class . . . 9
- 2.4 ERM and Uniform Convergence . . . 11
- 2.5 Bibliographic Note . . . 12

3 Uniform Convergence of Empirical Means **13**
- 3.1 Concentration Inequalities . . . 13
- 3.2 Rademacher Complexity . . . 14
- 3.3 Covering Numbers . . . 17
- 3.4 Binary Valued Functions: VC Dimension . . . 18
- 3.5 Real Valued Functions: Fat Shattering Dimension . . . 20

4 Model Selection **21**
- 4.1 Structural Risk Minimization . . . 21
- 4.2 Model Selection via Penalization . . . 22
- 4.3 Data Driven Penalties Using Rademacher Averages . . . 24
- 4.4 Hold-out Estimates . . . 24

5 Alternatives to Uniform Convergence **25**
- 5.1 Sample Compression . . . 25
- 5.2 Stability . . . 26
- 5.3 PAC-Bayesian Analysis . . . 28

6 Computational Aspects **29**
- 6.1 The PAC Model . . . 29
- 6.2 Weak and Strong Learning . . . 30
- 6.3 Random Classification Noise and the SQ Model . . . 31
- 6.4 The Agnostic PAC Model . . . 32
- 6.5 The Mistake Bound Model . . . 33
  - 6.5.1 Halving and Weighted Majority . . . 34
  - 6.5.2 Perceptron and Winnow . . . 37

7 Beyond the Basic Probabilistic Framework **42**

8 Conclusions and Future Trends **43**

1 Introduction

In a section on *Learning Machines* in one of his most famous papers (Turing, 1950), Turing wrote:

> Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child.

The year was 1950. Earlier, in 1943, McCulloch and Pitts (McCulloch and Pitts, 1943) had already hit upon the idea that mind-like machines could be built by studying how biological nerve cells behave. This directly lead to neural networks. By the late 1950s, Samuels (1959) had programmed a machine to play checkers and Rosenblatt (1959) had proposed one of earliest models of a learning machine, the perceptron. The importance of the perceptron in the history of learning theory cannot be overemphasized. Vapnik says that the appearance of the perceptron was "when the mathematical analysis of learning processes truly began". Novikoff's perceptron theorem (we will prove it in Section 6.5.2) was also proved in 1962 (Novikoff, 1963).

Vapnik (2000) points out the interesting fact that the 1960s saw four major developments that were to have lasting influence on learning theory. First, Tikhonov and others developed regularization theory for the solution of ill posed inverse problems. Regularization theory has had and continues to have tremendous impact on learning theory. Second, Rosenblatt, Parzen, and Chentsov pioneered nonparametric methods for density estimation. These beginnings were crucial for a distribution free theory of non-parametric classification and regression to emerge later. Third, Vapnik and Chervonenkis proved the uniform law of large numbers for indicators of sets. Fourth, Solomonoff, Kolmogorov, and Chaitin all independently discovered algorithmic complexity: the idea that the complexity of a string of 0's and 1's can be defined by the length of the shortest program that can generate that string. This later led Rissanen to propose his Minimum Description Length (MDL) principle.

In 1986, a major technique to learn weights in neural networks was discovered: backpropagation (Rumelhart et al., 1986). Around the same time, Valiant (1984) published his paper introducing a computational model of learning that was to be called the PAC (probably approximately correct) model later. The 1990s saw the appearance of support vector machines (Cortes and Vapnik, 1995) and AdaBoost (Freund and Schapire, 1995). This brings us to the brink of the second millenium which is where we end our short history tour.

As we can see, learning theory has absorbed influences from a variety of sources: cybernetics, neuroscience, nonparametric statistics, regularization theory of inverse problems, and the theory of computation and algorithms. Ongoing research is not only trying to overcome known limitations of classic models but is also grappling with new issues such as dealing with privacy and web-scale data.

Learning theory is a formal mathematical theory. Assumptions are made, beautiful lemmas are proved, and deep theorems are discovered. But developments in learning theory also lead to practical algorithms. SVMs and AdaBoost are prime examples of high impact learning algorithms evolving out of a principled and well-founded approach to learning. We remain hopeful that learning theory will help us discover even more exciting learning algorithms in the future.

Finally, a word about the mathematical background needed to read this overview of learning theory. Previous exposure to mathematical reasoning and proofs is a must. So is familiarity with probability theory. But measure theory is not needed as we will avoid all measure-theoretic details. It will also be helpful, but not essential, to be aware of the basics of computational complexity such as the notion of polynomial time computation and NP-completeness. Similarly, knowing a little bit of functional analysis will help.

2 Probabilistic Formulation of Learning Problems

A basic problem in learning theory is that of learning a functional dependence $f : \mathcal{X} \to \mathcal{Y}$ from some *input space* $\mathcal{X}$ to some *output* or *label* space $\mathcal{Y}$. Some special choices for the output space deserve particular attention since they arise often in applications. In *multiclass classification*, we have $Y = \{1, \ldots, K\}$ and the goal is to classify any input $x \in \mathcal{X}$ into one of K given classes. A special case of this problem is when $K = 2$. Then, the problem is called *binary classification* and it is mathematically convenient to think of the output space as $\mathcal{Y} = \{-1, +1\}$ rather than $\mathcal{Y} = \{0, 1\}$. In *regression*, the output space $\mathcal{Y}$ is some subset of the set $\mathbb{R}$ of real numbers. Usually $\mathcal{Y}$ is assumed to be a bounded interval, say $[-1, +1]$.

We assume that there is an underlying distribution P over $\mathcal{X} \times \mathcal{Y}$ and that the learner has access to n samples

$$(X_1, Y_1), \ldots, (X_n, Y_n)$$

where the (X_i, Y_i) are independent and identically distributed (or *iid*) random variables with the same distribution P. Note that P itself is assumed to be unknown to the learner. A *learning rule* or *learning algorithm* $\hat{f}_n$ is simply a mapping

$$\hat{f}_n : (\mathcal{X} \times \mathcal{Y})^n \times \mathcal{X} \to \mathcal{Y} \ .$$

That is, a learning rule maps the n samples to a function from $\mathcal{X}$ to $\mathcal{Y}$. When the sample is clear from context, we will denote the function returned by a

learning rule itself by $\hat{f}_n$.

2.1 Loss Function and Risk

In order to measure the predictive performance of a function $f : \mathcal{X} \to \mathcal{Y}$, we use a *loss function*. A loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a non-negative function that quantifies how bad the prediction $f(x)$ is if the true label is y. We say that $\ell(f(x), y)$ is the loss incurred by f on the pair (x, y). In the classification case, binary or otherwise, a natural loss function is the 0-1 loss:

$$\ell(y', y) = \mathbf{1}\left[y' \neq y\right] \ .$$

The 0-1 loss function, as the name suggests, is 1 if a misclassification happens and is 0 otherwise. For regression problems, some natural choices for the loss function are the *squared loss*:

$$\ell(y', y) = (y' - y)^2 \ ,$$

or the *absolute loss*:

$$\ell(y', y) = |y' - y| \ .$$

Given a loss function, we can define the *risk* of a function $f : \mathcal{X} \to \mathcal{Y}$ as its expected loss under the true underlying distribution:

$$R(f) = \mathbb{E}_{(X,Y)\sim P}\left[\ell(f(X), Y)\right] \ .$$

Note that the risk of any function f is not directly accessible to the learner who only sees the samples. But the samples can be used to calculate the *empirical risk* of f:

$$\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), Y_i) \ .$$

Minimizing the empirical risk over a fixed class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ of functions leads to a very important learning rule, namely *empirical risk minimization* (ERM):

$$\hat{f}_n = \operatorname*{argmin}_{f\in\mathcal{F}} \ \widehat{R}(f) \ .$$

Under appropriate conditions on $\mathcal{X}, \mathcal{Y}$ and $\mathcal{F}$, an empirical risk minimizer is guaranteed to exist though it will not be unique in general. If we knew the distribution P then the best function from $\mathcal{F}$ would be

$$f^\star_{\mathcal{F}} = \operatorname*{argmin}_{f\in\mathcal{F}} \ R(f) \ ,$$

as $f^\star_{\mathcal{F}}$ minimizes the expected loss on a random (X, Y) pair drawn from P. Without restricting ourselves to the class $\mathcal{F}$, the best possible function to use is

$$f^\star = \operatorname*{argmin}_{f} \ R(f) \ .$$

where the infimum is with respect to all[1] functions.

For classification with 0-1 loss, $f^\star$ is called the *Bayes classifier* and is given simply by

$$f^\star(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \; P(Y = y | X = x) \; .$$

Note that $f^\star$ depends on the underlying distribution P. If we knew P, we could achieve minimum misclassification probability by always predicting, for a given x, the label y with the highest conditional probability (ties can be broken arbitrarily). The following result bounds the *excess risk* $R(f) - R(f^\star)$ of any function f in the binary classification case.

Theorem 1. *For binary classification with* 0-1 *loss, we have*

$$R(f) - R(f^\star) = \mathbb{E}\left[\mathbf{1}\left[f(X) \neq f^\star(X)\right] |2\eta(X) - 1|\right]$$

where $\eta(x) = P(Y = +1 | X = x)$.

For the regression case with squared loss, $f^\star$ is called the *regression function* and is given simply by the conditional expectation of the label given x:

$$f^\star(x) = \mathbb{E}\left[Y | X = x\right] \; .$$

In this case, the excess risk takes a particularly nice form.

Theorem 2. *For regression with squared loss, we have*

$$R(f) - R(f^\star) = \mathbb{E}\left[|f(X) - f^\star(X)|^2\right] \; .$$

2.2 Universal Consistency

A particularly nice property for a learning rule $\hat{f}_n$ to have is that its (expected) excess risk converges to zero as the sample size n goes to infinity. That is,

$$\mathbb{E}\left[R(\hat{f}_n)\right] - R(f^\star) \to 0 \; .$$

Note $R(\hat{f}_n)$ is a random variable since $\hat{f}_n$ depends on a randomly drawn sample. A rule $\hat{f}_n$ is said to be *universally consistent* if the above convergence holds irrespective of the true underlying distribution P. This notion of consistency is also sometimes called *weak* universal consistency in order to distinguish it from *strong universal consistency* that demands

$$R(\hat{f}_n) - R(f^\star) \to 0$$

with probability 1.

The existence of universally consistent learning rules for classification and regression is a highly non-trivial fact which has been known since Stone's work

[1] We should say "all *measurable* functions" here. We will, however, ignore measurability issues in this overview article.

in the 1970s (Stone, 1977). Proving universal consistency typically requires addressing the *estimation error* versus *approximation error* trade-off. For instance, often the function $\hat{f}_n$ is guaranteed to lie in a function class $\mathcal{F}_n$ that does not depend on the sample. The classes $\mathcal{F}_n$ typically grow with n. Then we can decompose the excess risk of $\hat{f}_n$ as

$$R(\hat{f}_n) - R(f^\star) = \underbrace{R(\hat{f}_n) - R(f^\star_{\mathcal{F}_n})}_{\text{estimation error}} + \underbrace{R(f^\star_{\mathcal{F}_n}) - R(f^\star)}_{\text{approximation error}} .$$

This decomposition is useful as the estimation error is the only random part. The approximation error depends on how rich the function class $\mathcal{F}_n$ is but does not depend on the sample. The reason we have to trade these two errors off is that the richer the function class $\mathcal{F}_n$ the smaller the approximation error will be. However, that leads to a large estimation error. This trade-off is also known as the *bias-variance trade-off*. The bias-variance terminology comes from the regression with squared error case but tends to be used in the general loss setting as well.

2.3 Learnability with Respect to a Fixed Class

Unlike universal consistency, which requires convergence to the minimum possible risk over all functions, learnability with respect to a fixed function class $\mathcal{F}$ only demands that

$$\mathbb{E}\left[R(\hat{f}_n)\right] - R(f^\star_{\mathcal{F}}) \to 0 .$$

We can additionally require quantitative bounds on the number of samples needed to reach, with high probability, a given upper bound on excess risk relative to $f^\star_{\mathcal{F}}$. The *sample complexity* of a learning rule $\hat{f}_n$ is the minimum number $n_0(\varepsilon, \delta)$ such that for all $n \geq n_0(\varepsilon, \delta)$, we have

$$R(\hat{f}_n) - R(f^\star_{\mathcal{F}}) \leq \varepsilon$$

with probability at least $1 - \delta$. We now see how we can arrive at sample complexity bounds for ERM via *uniform convergence* of empirical means to true expectations.

We mention some examples of functions classes that arise often in practice.

Linear Functions This is one of the simplest functions classes. In any finite dimensional Euclidean space $\mathbb{R}^d$, define the class of real valued functions

$$\mathcal{F} = \{x \mapsto \langle w, x \rangle \ : \ w \in \mathcal{W} \subseteq \mathbb{R}^d\} ,$$

where $\langle w, x \rangle = \sum_{j=1}^d w_j x_j$ denotes the standard *inner product*. The *weight vector* might be additionally constrained. For instance, we may require $\|w\| \leq W$ for some norm $\|\cdot\|$. The set $\mathcal{W}$ takes care of such constraints.

Linear Threshold Functions or Halfspaces This is class of binary valued function obtained by thresholding linear functions at zero. This gives us the class

$$\mathcal{F} = \{x \mapsto \text{sign}(\langle w, x \rangle) \ : \ w \in \mathcal{W} \subseteq \mathbb{R}^d\} \ .$$

Reproducing Kernel Hilbert Spaces Linear functions can be quite restricted in their approximation ability. However, we can get a significant improvement by first mapping the inputs $x \in \mathcal{X}$ into a feature space through a *feature mapping* $\Phi : \mathcal{X} \to \mathcal{H}$ where $\mathcal{H}$ is a very high dimensional Euclidean space (or an infinite dimensional *Hilbert space*) and then considering linear functions in the feature space:

$$\mathcal{F} = \{x \mapsto \langle w, \Phi(x) \rangle_{\mathcal{H}} \ : \ w \in \mathcal{H}\} \ ,$$

where the inner prodct now carries the subscript $\mathcal{H}$ to emphasize the space where it operates. Many algorithms access the inputs only through pairwise inner products

$$\langle \Phi(x_i), \Phi(x_j) \rangle_{\mathcal{H}} \ .$$

In such cases, we do not care what the feature mapping is or how complicated it is to evaluate, provided we can compute these inner products efficiently.

This idea of using only inner products leads us to reproducing kernel Hilbert space. We say that a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric positive semidefinite (psd) if it satisfies the following two properties.

Symmetry For any $x, x' \in \mathcal{X}$, we have $K(x, x') = K(x', x)$.

Positive Semidefiniteness For any $(x_1, \ldots, x_n) \in \mathcal{X}^n$, the symmetric matrix

$$\mathbf{K} = (K(x_i, x_j))_{i,j} \in \mathbb{R}^{n \times n}$$

is psd[2]. The matrix $\mathbf{K}$ is often called the *Gram matrix*.

Note that, given a feature mapping Φ, it is trivial to verify that

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$$

is a symmetric psd function. However, the much less trivial converse is also true. Every symmetric psd function arises out of a feature mapping. This is a consequence of the *Moore-Aronszajn* theorem. To state the theorem, we first need a definition. A *reproducing kernel Hilbert space or RKHS* $\mathcal{H}$ is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ such that there exists a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying the properties:

- For any $x \in \mathcal{X}$, the function $K_x = K(x, \cdot)$ is in $\mathcal{H}$.
- For any $x \in \mathcal{X}$, $f \in \mathcal{H}$, the *reproducing property* holds: $f(x) = \langle f, K_x \rangle_{\mathcal{H}}$.

[2]Recall that a symmetric matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is psd iff for all $u \in \mathbb{R}^n$, $\langle u, \mathbf{K}u \rangle \geq 0$.

Theorem 3 (Moore-Aronszajn)**.** *Given a symmetric psd function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there is a unique RKHS $\mathcal{H}$ with reproducing kernel K.*

This theorem gives us the feature mapping easily. Starting from a symmetric psd function K, we use the Moore-Aronszajn theorem to get an RKHS $\mathcal{H}$ whose reproducing kernel is K. Now, by the reproducing property, for any $x, x' \in \mathcal{X}$,

$$K(x, x') = K_x(x') = \langle K_x, K_{x'} \rangle_{\mathcal{H}}$$

which means that $\Phi : \mathcal{X} \mapsto \mathcal{H}$ given by $\Phi(x) = K_x$ is a feature mapping we were looking for. Given this equivalence between symmetric psd functions and RKHSes, we will use the word *kernel* for a symmetric psd function itself.

Convex Hulls This example is not about a function class but rather about a means of obtaining a new function class from an old one. Say, we have class $\mathcal{F}$ of binary valued functions $f : \mathcal{X} \to \{\pm 1\}$. We can consider functions that take a *majority vote* over a subset of functions in $\mathcal{F}$. That is,

$$f'(x) = \text{sign}\left(\sum_{\ell=1}^{L} f_\ell\right), \ f_\ell \in \mathcal{F}$$

More generally, we can assign weights $w_\ell \geq 0$ to f_ℓ and consider a weighted majority

$$f'(x) = \text{sign}\left(\sum_{\ell=1}^{L} w_\ell f_\ell\right) \ .$$

Constraining the weights to sum to no more than W gives us the scaled convex hull

$$W \cdot \text{conv}(\mathcal{F} \cup \{0\}) = \left\{ \sum_{\ell=1}^{L} w_\ell f_\ell \ : \ L \in \mathbb{N}, \ w_\ell \geq 0, \sum_{\ell}^{L} w_\ell \leq W \right\} \ .$$

The class above can, of course, be thresholded to produce binary valued functions.

2.4 ERM and Uniform Convergence

The excess risk of ERM relative to $f^\star_{\mathcal{F}}$ can be decomposed as

$$R(\hat{f}_n) - R(f^\star_{\mathcal{F}}) = (R(\hat{f}_n) - \widehat{R}(\hat{f}_n)) + (\widehat{R}(\hat{f}_n) - \widehat{R}(f^\star_{\mathcal{F}})) + (\widehat{R}(f^\star_{\mathcal{F}}) - R(f^\star_{\mathcal{F}})) \ .$$

By the definition of ERM, the difference $\widehat{R}(\hat{f}_n) - \widehat{R}(f^\star_{\mathcal{F}})$ is non-positive. The difference $\widehat{R}(f^\star_{\mathcal{F}}) - R(f^\star_{\mathcal{F}})$ is also easy to deal with since it deals with a fixed (i.e. non-random) function $f^\star_{\mathcal{F}}$. Using Hoeffding's inequality (see Section 3.1 below), we have, with probability at least $1 - \delta$,

$$\widehat{R}(f^\star_{\mathcal{F}}) - R(f^\star_{\mathcal{F}}) \leq \sqrt{\frac{\log(1/\delta)}{2n}} \ .$$

This simply reflects the fact that as the sample size n grows, we expect the empirical average of $f^\star_{\mathcal{F}}$ to converge to its true expectation.

The matter with the difference $R(\hat{f}_n) - \widehat{R}(\hat{f}_n)$ is not so simple since $\hat{f}_n$ is a random function. But we do know $\hat{f}_n$ lies in $\mathcal{F}$. So we can clearly bound

$$R(\hat{f}_n) - \widehat{R}(\hat{f}_n) \leq \sup_{f \in \mathcal{F}} \left(R(f) - \widehat{R}(f) \right) . \tag{1}$$

This provides a *generalization bound* (or, more properly, a *generalization error bound*) for ERM. A generalization bound is an upper bound on the true expectation $R(\hat{f}_n)$ of a learning rule in terms of its empirical performance $\widehat{R}(\hat{f}_n)$ (or some other performance measure computed from data). A generalization bound typically involves addtional terms that measure the statistical complexity of the class of functions that the learning rule uses. Note the important fact that the bound above is valid not just for ERM but for any learning rule that outputs a function $\hat{f}_n \in \mathcal{F}$.

The measure of complexity in the bound (1) above is the maximum deviation between true expectation and empirical average over functions in $\mathcal{F}$. For each *fixed* $f \in \mathcal{F}$, we clearly have

$$R(f) - \widehat{R}(f) \to 0$$

both in probability (weak law of large numbers) and almost surely (strong law of large numbers). But in order for

$$\sup_{f \in \mathcal{F}} R(f) - \widehat{R}(f) \to 0$$

the law of large numbers has to hold *uniformly* over the function class $\mathcal{F}$.

To summarize, we have shown in this section that, with probability at least $1 - \delta$,

$$R(\hat{f}_n) - R(f^\star_{\mathcal{F}}) \leq \sup_{f \in \mathcal{F}} \left(R(f) - \widehat{R}(f) \right) + \sqrt{\frac{\log(1/\delta)}{2n}} . \tag{2}$$

Thus, we can bound the excess risk of ERM if we can prove uniform convergence of empirical means to true expectations for the function class $\mathcal{F}$. This excess risk bound, unlike the generalization error bound (1), is applicable only to ERM.

2.5 Bibliographic Note

There are several book length treatments of learning theory that emphasize different aspects of the subject. The reader may wish to consult Anthony and Biggs (1997), Anthony and Bartlett (1999), Devroye et al. (1996), Vapnik (2006), Vapnik (2000), Vapnik (1998), Vidyasagar (2003), Natarajan (1991), Cucker and Zhou (2007), Kearns and Vazirani (1994), Cesa-Bianchi and Lugosi (2006), Györfi et al. (2002), Hastie et al. (2009).

3 Uniform Convergence of Empirical Means

Let us consider the simplest case of a finite function class $\mathcal{F}$. In this case, Hoeffding's inequality (see Theorem 5) gives us for any $f \in F$, with probability at least $1 - \delta/|\mathcal{F}|$,

$$R(f) - \widehat{R}(f) \leq c\sqrt{\frac{\log(|\mathcal{F}|/\delta)}{n}} \ .$$

Therefore, denoting the event that the above inequality fails to hold for function f by A_f, we have

$$P(\exists f \in \mathcal{F},\ A_f \text{ holds}) \leq \sum_{f \in \mathcal{F}} P(A_f) \leq |\mathcal{F}| \cdot \frac{\delta}{|\mathcal{F}|} = \delta \ .$$

The first inequality is called a *union bound* in probability theory. It is exact if and only if all the events it is applied to are pairwise disjoint, i.e. $A_f \cap A_{f'} = \emptyset$ for $f \neq f'$. Clearly, if there are functions that are "similar" to each other, the bound above will be loose. Nevertheless, the union bound is an important starting point for the analysis of uniform convergence of empirical means and for a finite class, it gives us the following result.

Theorem 4. *Consider a finite class $\mathcal{F}$ and a loss function bounded by 1. Then, we have, with probability at least $1 - \delta$,*

$$\sup_f \ (R(f) - \widehat{R}(f)) \leq \sqrt{\frac{\log|\mathcal{F}| + \log(1/\delta)}{2n}} \ .$$

Therefore, using (2), we have the following result for ERM. With probability at least $1 - 2\delta$,

$$R(\hat{f}_n) - R(f^\star_{\mathcal{F}}) \leq \sqrt{\frac{\log|\mathcal{F}|}{2n}} + \sqrt{\frac{2\log(1/\delta)}{n}} \ .$$

It is instructive to focus on the first term on the right hand side. First, it decreases as n increases. This means that ERM gets better as it works with more data. Second, it increases as the function class $\mathcal{F}$ grows in size. But the dependence is logarithmic and hence mild.

However, the bound above is useless if $|\mathcal{F}|$ is not finite. The goal of the theory we will build below (often called "VC theory" after its architects Vapnik and Chervonenkis) will be to replace $\log|\mathcal{F}|$ with an appropriate measure of the *capacity* of an infinite function class $\mathcal{F}$.

3.1 Concentration Inequalities

Learning theory uses *concentration inequalities* heavily. A typical concentration inequality will state that a certain random variable is tightly concentrated around its mean (or in some cases median). That is, the probability that they deviate far enough from the mean is small.

The following concentration inequality, called *Hoeffding's inequality*, applies to sums of iid random variables.

Theorem 5. *Let $Z_i \in [0,1]$ be bounded iid random variables with common expectation μ. Then we have, for $\varepsilon > 0$*

$$\mathbb{P}\left[\mu - \frac{1}{n}\sum_{i=1}^{n} Z_i > \varepsilon\right] \leq \exp(-2n\varepsilon^2) \ .$$

This can be restated as follows. For any $\delta \in (0,1)$,

$$\mathbb{P}\left[\mu - \frac{1}{n}\sum_{i=1}^{n} Z_i \leq \sqrt{\frac{\log(1/\delta)}{2n}}\right] > 1 - \delta \ .$$

A couple of remarks are in order. First, even though we have stated Hoeffding's inequality as applying to bounded iid random variables, only the boundedness and independence are really needed. It is possible to let the Z_i's have different distributions. Second, the above result provides probability upper bounds for devations below the mean μ but a similar result holds for devations of $\frac{1}{n}\sum_{i=1}^{n} Z_i$ above μ as well.

In fact, Hoeffding's inequality can be considered a special case of the following inequality that is often called *McDiarmid's inequality* or the *bounded differences inequality*.

Theorem 6. *Let $F : \mathcal{X}^n \to \mathbb{R}$ be a function that satisfies the bounded differences property: for all $i \in [n]$ and $x_{1:n} \in \mathcal{X}^n, x_i' \in \mathcal{X}$,*

$$|f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i$$

for some constants c_i. Then, for any independent $X_1, \ldots, X_n$, the random variable $Z = f(X_1, \ldots, X_n)$ satisfies,

$$\mathbb{P}\left[Z - \mathbb{E}\left[Z\right] \geq \varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n} c_i^2}\right)$$

as well as

$$\mathbb{P}\left[Z - \mathbb{E}\left[Z\right] \leq -\varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n} c_i^2}\right)$$

for any $\varepsilon > 0$.

3.2 Rademacher Complexity

Let us recall that we need to control the following quantity:

$$\sup_{f \in \mathcal{F}} \ R(f) - \widehat{R}(f) \ .$$

Since this satisfies the conditions of McDiarmid's inequality with $c_i = 1/n$, we have, with probability at least $1 - \delta$,

$$\sup_{f \in \mathcal{F}} \ R(f) - \widehat{R}(f) \leq \mathbb{E}\left[\sup_{f \in \mathcal{F}} \ R(f) - \widehat{R}(f)\right] + \sqrt{\frac{\log(1/\delta)}{2n}} \ .$$

Now, we can appeal to a powerful idea known as *symmetrization*. The basic idea is to replace

$$\sup_{f\in\mathcal{F}} \; R(f) - \widehat{R}(f)$$

with

$$\sup_{f\in\mathcal{F}} \; \frac{1}{n}\sum_{i=1}^{n}(\mathbb{E}\left[\ell(f(X_i'),Y_i')\right] - \ell(f(X_i),Y_i))$$

where $(X_1',Y_1'),\ldots,(X_n',Y_n')$ are samples drawn from the distribution P that are independent of each other and of the original sample. The new sample is often referred to as the *ghost sample*. The point of introducing the ghost sample is that, by an application of Jensen's inequality, we get

$$\begin{aligned}\mathbb{E}\left[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)\right] &= \mathbb{E}\left[\sup_{f\in\mathcal{F}} \; \frac{1}{n}\sum_{i=1}^{n}(\mathbb{E}\left[\ell(f(X_i'),Y_i')\right] - \ell(f(X_i),Y_i))\right] \\ &\leq \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(\ell(f(X_i'),Y_i') - \ell(f(X_i),Y_i))\right] \; .\end{aligned}$$

Now, the distribution of

$$\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(\ell(f(X_i'),Y_i') - \ell(f(X_i),Y_i))$$

is invariant under exchanging any X_i, Y_i with an X_i', Y_i'. That is, for any fixed choice of ± 1 signs $\epsilon_1,\ldots,\epsilon_n$, the above quantity has the same distribution as

$$\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i(\ell(f(X_i'),Y_i') - \ell(f(X_i),Y_i)) \; .$$

This allows us introduce even more randomization by making the ϵ_i's themselves random. This gives,

$$\begin{aligned}\mathbb{E}\left[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)\right] &\leq \mathbb{E}_{\epsilon,X,Y,X',Y'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i(\ell(f(X_i'),Y_i') - \ell(f(X_i),Y_i))\right] \\ &\leq \mathbb{E}_{\epsilon,X',Y'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\ell(f(X_i'),Y_i')\right] \\ &\quad + \mathbb{E}_{\epsilon,X,Y}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(-\epsilon_i)\ell(f(X_i),Y_i)\right] \\ &= 2\cdot\mathbb{E}_{\epsilon,X,Y}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\ell(f(X_i),Y_i)\right] \; .\end{aligned}$$

This directly yields a bound in terms of the *Rademacher complexity*:

$$\mathfrak{R}_n\left(\ell\circ\mathcal{F}\right) = \mathbb{E}_{\epsilon,X,Y}\left[\sup_{f\in\mathcal{F}} \; \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\ell(f(X_i),Y_i)\right] \; ,$$

where ϵ_i is a *Rademacher random variable* that is either -1 or $+1$, each with probability $\frac{1}{2}$. The subscript below the expectation serves as a reminder that the expectation is being taken with respect to the randomness in X_i, Y_i's and ϵ_i's. We have thus proved the following theorem.

Theorem 7 (Symmetrization)**.** *For a function class $\mathcal{F}$ and loss ℓ, define the loss class*

$$\ell \circ \mathcal{F} = \{(x, y) \mapsto \ell(f(x), y) \ : \ f \in \mathcal{F}\} \ .$$

We have,

$$\mathbb{E}\left[\sup_{f \in \mathcal{F}} \ R(f) - \widehat{R}(f)\right] \leq 2\,\mathfrak{R}_n\left(\ell \circ \mathcal{F}\right) \ .$$

An interesting observation is that the sample dependent quantity

$$\widehat{\mathfrak{R}}_n\left(\ell \circ \mathcal{F}\right) = \mathbb{E}_\epsilon\left[\sup_{f \in \mathcal{F}} \ \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \ell(f(X_i), Y_i)\right]$$

also satisfies the bounded differences condition with $c_i = 1/n$. This is the *empirical Rademacher average* and the expectation in its definition is only over the Rademacher random variables. Applying McDiarmid's inequality, we have,

$$\mathfrak{R}_n\left(\ell \circ \mathcal{F}\right) \leq \widehat{\mathfrak{R}}_n\left(\ell \circ \mathcal{F}\right) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

with probability at least $1 - \delta$.

To prove a bound on the Rademacher complexity of a finite class, the following lemma, due to Massart (2007), is useful.

Lemma 8 (Massart's finite class lemma)**.** *Let $\mathcal{A}$ be a finite subset of $\mathbb{R}^n$ and $\epsilon_1, \ldots, \epsilon_n$ be independent Rademacher random variables. Let $r = \max_{a \in \mathcal{A}} \|a\|_2$. Then, we have,*

$$\mathbb{E}\left[\sup_{a \in \mathcal{A}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i a_i\right] \leq \frac{r\sqrt{2\log|\mathcal{A}|}}{n} \ .$$

Therefore, for any finite class $\mathcal{F}$ consisting of functions bounded by 1,

$$\widehat{\mathfrak{R}}_n\left(\mathcal{F}\right) \leq \sqrt{\frac{2\log|\mathcal{F}|}{n}} \ .$$

Since the right hand side is not random, the same bound holds for $\mathfrak{R}_n\left(\mathcal{F}\right)$ as well.

The Rademacher complexity, and its empirical counterpart, satisfy a number of interesting structural properties.

Theorem 9 (Structural Properties of Rademacher Complexity)**.** *Let $\mathcal{F}, \mathcal{F}_1, \ldots, \mathcal{F}_k$ and $\mathcal{H}$ be classes of real valued functions. Then $\mathfrak{R}_n\left(\cdot\right)$ (as well as $\widehat{\mathfrak{R}}_n\left(\cdot\right)$) satisfies the following properties.*

Monotonicity *If* $\mathcal{F} \subseteq \mathcal{H}$, $\mathfrak{R}_n(\mathcal{F}) \leq \mathfrak{R}_n(\mathcal{H})$.

Convex Hull $\mathfrak{R}_n(\text{conv}(\mathcal{F})) = \mathfrak{R}_n(\mathcal{F})$.

Scaling $\mathfrak{R}_n(c\mathcal{F}) = |c|\mathfrak{R}_n(\mathcal{F})$ *for any* $c \in \mathbb{R}$.

Contraction *If* $\phi : \mathbb{R} \to \mathbb{R}$ *is Lipschitz with constant* L_ϕ, *then*

$$\mathfrak{R}_n(\phi \circ \mathcal{F}) \leq L_\phi \mathfrak{R}_n(\mathcal{F}) \ .$$

Translation *For any bounded function* h, $\mathfrak{R}_n(\mathcal{F} + h) = \mathfrak{R}_n(\mathcal{F})$.

Subadditivity $\mathfrak{R}_n\left(\sum_{i=1}^k \mathcal{F}_i\right) \leq \sum_{i=1}^n \mathfrak{R}_n(\mathcal{F}_i)$.

For proofs of these properties, see Bartlett and Mendelson (2002). However, note that the Rademacher complexity is defined there such that an extra absolute value is taken before the supremum over $f \in \mathcal{F}$. That is,

$$\mathfrak{R}_n^{\text{abs}}(\mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n f(X_i)\right|\right] \ .$$

Nevertheless, the proofs given there generalize quite easily to definition of Rademacher complexity that we are using. For example, see Ambroladze et al. (2007), where the Rademacher complexity as we have defined it here (without the absolute value) is called *free Rademacher complexity*.

Using these properties, it is possible to directly bound the Rademacher complexity of several interesting functions classes. See Bartlett and Mendelson (2002) for examples. The contraction property is a very useful one. It allows us to transition from the Rademacher complexity of the loss class to the Rademacher complexity of the function class itself (for Lipschitz losses).

The Rademacher complexity can also be bounded in terms of another measure of complexity of a function class: namely covering numbers.

3.3 Covering Numbers

The idea underlying covering numbers is very intuitive. Let $(\mathcal{D}, \rho)$ be any (pseudo-)metric space and fix a subset $T \subseteq \mathcal{D}$. A set $T' \subseteq \mathcal{D}$ is said to be an α-cover of T if

$$\forall f \in T, \ \exists g \in T' \text{ s.t. } \rho(f, g) \leq \alpha \ .$$

In other words "balls" of radius α placed at elements of T' "cover" the set T entirely. The *covering number* (at scale α) of T is defined as the size of the smallest cover of T (at scale α).

$$\mathcal{N}(T, \rho, \alpha) = \min \ \{|T'| \ : \ T' \text{ is an } \alpha\text{-cover of } T\} \ .$$

The logarithm of covering number is called *entropy*.

Given a sample $x_{1:n} = (x_1, \ldots, x_n)$, define the data dependent metric on $\mathcal{F}$ as

$$\hat{\rho}_p(f, g) = \left(\frac{1}{n} \sum_{i=1}^{n} |f(x_i) - g(x_i)|^p \right)^{1/p}$$

where $p \in [1, \infty)$. Taking the limit $p \to \infty$ suggests the metric

$$\hat{\rho}_\infty(f, g) = \max_{i \in [n]} |f(x_i) - g(x_i)| \ .$$

These metrics gives us the p-norm covering numbers

$$\mathcal{N}_p(\mathcal{F}, x_{1:n}, \alpha) = \mathcal{N}(\mathcal{F}, \hat{\rho}, \alpha) \ .$$

These covering numbers increase with p. That is, for $p \in [1, \infty]$,

$$\mathcal{N}_1(\mathcal{F}, x_{1:n}, \alpha) \leq \mathcal{N}_p(\mathcal{F}, x_{1:n}, \alpha) \leq \mathcal{N}_\infty(\mathcal{F}, x_{1:n}, \alpha) \ .$$

Finally, we can also define the worst case (over samples) p-norm covering number

$$\mathcal{N}_p(\mathcal{F}, n, \alpha) = \sup_{x_{1:n} \in \mathcal{X}^n} \mathcal{N}_p(\mathcal{F}, x_{1:n}, \alpha) \ .$$

A major result connecting Rademacher complexity with covering numbers is due to Dudley (1967).

Theorem 10 (Dudley's entropy integral bound)**.** *For any $\mathcal{F}$ consisting of real valued functions bounded by 1, we have*

$$\widehat{\mathfrak{R}}_n(\mathcal{F}) \leq \alpha + 12 \int_\alpha^1 \sqrt{\frac{\log \mathcal{N}_2(\mathcal{F}, X_{1:n}, \beta)}{n}} d\beta \ .$$

Therefore, using Jensen's inequality, we have

$$\mathfrak{R}_n(\mathcal{F}) \leq \alpha + 12 \int_\alpha^1 \sqrt{\frac{\log \mathbb{E}\left[\mathcal{N}_2(\mathcal{F}, X_{1:n}, \beta)\right]}{n}} d\beta \ .$$

3.4 Binary Valued Functions: VC Dimension

For binary function classes $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{X}}$, we can provide distribution free upper bounds on the Rademacher complexity or covering numbers in terms of a combinatorial parameter called the *Vapnik-Chervonenkis (VC) dimension.*

To define it, consider a sequence $x_{1:n} = (x_1, \ldots, x_n)$. We say that $x_{1:n}$ is *shattered* by $\mathcal{F}$ if each of the 2^n possible ± 1 labelings of these n points are realizable using some $f \in \mathcal{F}$. That is, $x_{1:n}$ is shattered by $\mathcal{F}$ iff

$$|\{(f(x_1), \ldots, f(x_n)) \ : \ f \in \mathcal{F}\}| = 2^n \ .$$

The *VC dimension* of $\mathcal{F}$ is the length of the longest sequence that can shattered by $\mathcal{F}$:

$$\mathrm{VCdim}(\mathcal{F}) = \max \{n \ : \ \exists x_{1:n} \in \mathcal{X}^n \text{ s.t. } x_{1:n} \text{ is shattered by } \mathcal{F}\} \ .$$

The size of the restriction of $\mathcal{F}$ to $x_{1:n}$ is called the *shatter coefficient*:

$$\mathbb{S}(\mathcal{F}, x_{1:n}) = |\{(f(x_1), \ldots, f(x_n)) \,:\, f \in \mathcal{F}\}| \,.$$

Considering the worst sequence gives us the *growth function*

$$\mathbb{S}(\mathcal{F}, n) = \max_{x_{1:n} \in \mathcal{X}^n} \mathbb{S}(\mathcal{F}, x_{1:n}) \,.$$

If $\mathrm{VCdim}(\mathcal{F}) = d$, then we know that $\mathbb{S}(\mathcal{F}, n) = 2^n$ for all $n \leq d$. One can ask: what is the behavior of the growth function for $n > d$? The following combinatorial lemma proved independently by Vapnik and Chervonenkis (1968), Sauer (1972), and Shelah (1972), gives the answer to this question.

Lemma 11. *If* $\mathrm{VCdim}(\mathcal{F}) = d$*, then we have, for any* $n \in \mathbb{N}$*,*

$$\mathbb{S}(\mathcal{F}, n) = \sum_{i=0}^{d} \binom{n}{i} \,.$$

In particular, if $d > 2$*,* $\mathbb{S}(\mathcal{F}, n) \leq n^d$*.*

Thus, for a class with finite VC dimension, the Rademacher complexity can be bounded directly using Massart's lemma

Theorem 12. *Suppose* $2 < \mathrm{VCdim}(\mathcal{F}) = d < \infty$*. Then, we have,*

$$\widehat{\mathfrak{R}}_n(\mathcal{F}) \leq \sqrt{\frac{2 \log \mathbb{S}(\mathcal{F}, n)}{n}} \leq \sqrt{\frac{2d \log n}{n}} \,.$$

It is possible to shave off the $\log n$ factor in the bound above plugging an estimate of the entropy $\log N_2(\mathcal{F}, n, \alpha)$ for VC classes due to Haussler (1995) into Theorem 10.

Theorem 13. *Suppose* $\mathrm{VCdim}(\mathcal{F}) = d < \infty$*. We have, for any* $n \in \mathbb{N}$*,*

$$\mathcal{N}_2(\mathcal{F}, n, \alpha) \leq Cd \log \frac{1}{\alpha}$$

for a universal constant C*.*

These results show that for a class $\mathcal{F}$ with finite VC dimension d, the sample complexity required by ERM to achieve ε excess risk relative to $f^\star_{\mathcal{F}}$ is, with probability at least $1 - \delta$, $O(\frac{d}{\varepsilon^2} + \log \frac{1}{\delta})$. Therefore, a finite VC dimension is *sufficient* for learnability with respect to a fixed functions class. In fact, a finite VC dimension also turns out to be necessary (see, for instance, (Devroye et al., 1996, Chapter 14)). Thus, we have the following characterization.

Theorem 14. *In the binary classification setting with* 0-1 *loss, there is a learning algorithm with bounded sample complexity for every* $\varepsilon, \delta \in (0, 1)$ *iff* $\mathrm{VCdim}(\mathcal{F}) < \infty$*.*

Note that, when $\mathrm{VCdim}(\mathcal{F}) < \infty$, the dependence of the sample complexity on $1/\varepsilon$ is polynomial and on $1/\delta$ is *logarithmic*. Moreover, such a sample complexity is achieved by ERM. On the other hand, when $\mathrm{VCdim}(\mathcal{F}) = \infty$, then no learning algorithm, including ERM, can have bounded sample complexity for arbitrarily small ε, δ.

3.5 Real Valued Functions: Fat Shattering Dimension

For real valued functions, getting distribution free upper bounds on Rademacher complexity or covering numbers involves combinatorial parameters similar to the VC dimension. The appropriate notion for determining learnability using a finite number of samples turns out to be a *scale sensitive* combinatorial dimension known as the *fat shattering dimension*.

Fix a class $\mathcal{F}$ consisting of bounded real valued functions $f : \mathcal{X} \to [0, 1]$ and a scale $\alpha > 0$. We say that a sequence $x_{1:n} = (x_1, \ldots, x_n)$ is α-shattered by $\mathcal{F}$ if there exists a witness sequence $s_{1:n} \in \mathbb{R}^n$ such that, for every $\epsilon_{1:n} \in \{\pm 1\}^n$, there is an $f \in \mathcal{F}$ such that

$$\epsilon_i(f(x_i) - s_i) \geq \alpha \ .$$

In words, this means that, for any of the 2^n possibilities for the sign pattern $\epsilon_{1:n}$, we have a function $f \in \mathcal{F}$ whose values at the points x_i is above or below s_i depending on whether ϵ_i is $+1$ or -1. Moreover, the gap between $f(x_i)$ and s_i is required to be at least α.

The fat-shattering dimension of $\mathcal{F}$ at scale α is the size of the longest sequence that can be α-shattered by $\mathcal{F}$:

$$\mathrm{fat}_\alpha(\mathcal{F}) = \max\{n \ : \ \exists x_{1:n} \in \mathcal{X}^n \text{ s.t. } x_{1:n} \text{ is } \alpha\text{-shattered by } \mathcal{F}\} \ .$$

A bound on ∞-norm covering numbers in terms of the fat shattering dimension were first given by Alon et al. (1997).

Theorem 15. *Fix a class $\mathcal{F} \subseteq [0, 1]^{\mathcal{X}}$. Then we have, for any $\alpha > 0$,*

$$\mathcal{N}_\infty(\mathcal{F}, n, \alpha) \leq 2\left(\frac{4n}{\alpha^2}\right)^{d\log(en/d\alpha)}$$

where $d = \mathrm{fat}_{\alpha/2}(\mathcal{F})$.

Using this bound, the sample complexity of learning a real valued function class of bounded range using a Lipschitz loss[3] ℓ is $O(\frac{d}{\varepsilon^2}\log\frac{1}{\varepsilon} + \log\frac{1}{\delta})$ where $d = \mathrm{fat}_{c\varepsilon}(\mathcal{F})$ for a universal constant $c > 0$. The importance of the fat shattering dimension lies in the fact that if $\mathrm{fat}_\alpha(\mathcal{F}) = \infty$ for some $\alpha > 0$, then no learning algorithm can have bounded sample complexity for learning the functions for arbitrarily small values of ε, δ.

Theorem 16. *In the regression setting with either squared loss or absolute loss, there is a learning algorithm with bounded sample complexity for every $\varepsilon, \delta \in (0, 1)$ iff $\forall \alpha > 0, \mathrm{fat}_\alpha(\mathcal{F}) < \infty$.*

Note that here, unlike the binary classification case, the dependence of the sample complexity on $1/\varepsilon$ is not fixed. It depends on how fast $\mathrm{fat}_\alpha(\mathcal{F})$ grows as $\alpha \to 0$. In particular, if $\mathrm{fat}_\alpha(\mathcal{F})$ grows as $1/\alpha^p$ then the sample complexity is polynomial in $1/\varepsilon$ (and logarithmic in $1/\delta$).

[3] By this we mean $\ell(\cdot, y)$ is Lipschitz for all y.

4 Model Selection

In general, model selection refers to the task of selecting an appropriate model from a given family based on available data. For ERM, the "model" is the class $\mathcal{F}$ of functions $f : \mathcal{X} \to \mathcal{Y}$ that are intended to capture the functional dependence between the input variable X and the output Y. If we choose too small an $\mathcal{F}$ then ERM will not have good performance if $f^\star$ is not even well approximated by any member of $\mathcal{F}$. One the other hand, if $\mathcal{F}$ is too large (say all functions) then the estimation error will be so large that ERM will not perform well even if $f^\star \in \mathcal{F}$.

4.1 Structural Risk Minimization

Consider the classification setting with 0-1 loss. Suppose we have a countable sequence of nested function classes:

$$\mathcal{F}^{(1)} \subset \mathcal{F}^{(2)} \subset \ldots \subset \mathcal{F}^{(k)} \ldots$$

where $\mathcal{F}^{(k)}$ has VC dimension d_k. The function $\hat{f}_n^{(k)}$ chosen by ERM over $\mathcal{F}_k$ satisfies the bound

$$R(\hat{f}_n^{(k)}) \leq \widehat{R}(\hat{f}_n^{(k)}) + O\left(\sqrt{\frac{d_k \log n}{n}}\right)$$

with high probability. As k increases, the classes become more complex. So, the empirical risk term will decrease with k whereas the VC dimension term will increase. The principle of *structural risk minimization (SRM)* chooses a value of k by minimizing the right hand side above.

Namely, we choose

$$\hat{k} = \operatorname*{argmin}_k \widehat{R}(\hat{f}_n^{(k)}) + C\sqrt{\frac{d_k \log n}{n}} \ , \tag{3}$$

for some universal constant C.

It can be shown (see, for example, (Devroye et al., 1996, Chapter 18)) that SRM leads to universally consistent rules provided appropriate conditions are placed on the functions classes $\mathcal{F}_k$.

Theorem 17 (Universal Consistency of SRM)**.** *Let $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots$ be a sequence of classes such that for any distribution P,*

$$\lim_{k\to\infty} R(f^\star_{\mathcal{F}^{(k)}}) = R(f^\star) \ .$$

Assume also that $\mathrm{VCdim}(\mathcal{F}^{(1)}) < \mathrm{VCdim}(\mathcal{F}^{(2)}) < \ldots$ *are all finite. Then the learning rule $\hat{f}_n^{(\hat{k})}$ with $\hat{k}$ chosen as in* (3) *is strongly universally consistent.*

The penalty term in (3) depends on the VC dimension of $\mathcal{F}^{(k)}$. It turns out that it is not essential. All we need is an upper bound on the risk of ERM in terms of the empirical risk and some measure of the complexity of the function class. One could also imagine proving results for regression problems by replacing the VC dimension with fat-shattering dimension. We can think of SRM as *penalizing* larger classes by their capacity to overfit the data. However, the penalty in SRM is fixed in advance and is not data dependent. We will now look into more general penalties.

4.2 Model Selection via Penalization

Given a (not necessarily nested) countable or finite collection of models $\{\mathcal{F}^{(k)}\}_k$ and a penalty function $\text{pen} : \mathbb{N} \to \mathbb{R}_+$, define $\hat{k}$ as the minimizer of

$$\widehat{R}(\hat{f}_n^{(k)}) + \text{pen}(k) \ .$$

Recall that $\hat{f}_n^{(k)}$ is the result of ERM over $\mathcal{F}^{(k)}$. Also note that the penalty can additionally depend on the data. Now define the *penalized estimator* as

$$\tilde{f}_n = \hat{f}_n^{(\hat{k})} \ .$$

The following theorem shows that any probabilistic upper bound $U_{n,k}$ on the risk $R(\hat{f}_n)$ can be used to design a penalty for which a performance bound can be given.

Theorem 18. *Assume that there are positive numbers c and m such that for each k we have the guarantee*

$$\forall \varepsilon > 0, \ \mathbb{P}\left[R(\hat{f}_n^{(k)}) > U_{n,k} + \varepsilon\right] \leq c \exp(-2m\varepsilon^2) \ .$$

Then the penalized estimator $\tilde{f}_n$ with

$$\text{pen}(k) = U_{n,k} - \widehat{R}(\hat{f}_n^{(k)}) + \sqrt{\frac{\log k}{m}}$$

satisfies

$$\mathbb{E}\left[R(\tilde{f}_n)\right] - R(f^\star) \leq \min_k \left\{\mathbb{E}\left[\text{pen}(k)\right] + \left(R(f^\star_{\mathcal{F}^{(k)}}) - R(f^\star)\right)\right\} + \sqrt{\frac{\log(ce)}{2m}} \ .$$

Note that the result says that the penalized estimator achieves an almost optimal trade-off between the approximation error $R(f^\star_{\mathcal{F}^{(k)}}) - R(f^\star)$ and the expected complexity $\mathbb{E}\left[\text{pen}(k)\right]$. For a good upper bound $U_{n,k}$, the expected complexity will be a good upper bound on the (expected) estimation error. Therefore, the model selection properties of the penalized estimator depend on how good an upper bound $U_{n,k}$ we can find.

Note that structural risk minimization corresponds to using a distribution free upper bound

$$U_{n,k} = \widehat{R}(\hat{f}_n^{(k)}) + O\left(\sqrt{\frac{d_k \log n}{n}}\right) \ .$$

The looseness of these distribution free upper bounds has prompted researchers to design data dependent penalties. We shall see an examples based on empirical Rademacher averages in the next section.

Proof. For any $\varepsilon > 0$,

$$\begin{aligned}
&\mathbb{P}\left[R(\tilde{f}_n) - (\widehat{R}(\tilde{f}_n) + \mathrm{pen}(\hat{k})) > \varepsilon\right] \\
&\leq \mathbb{P}\left[\exists k \text{ s.t. } R(\hat{f}_n^{(k)}) - (\widehat{R}(\hat{f}_n^{(k)}) + \mathrm{pen}(k)) > \varepsilon\right] \\
&\leq \sum_{k=1}^{\infty} \mathbb{P}\left[R(\hat{f}_n^{(k)}) - (\widehat{R}(\hat{f}_n^{(k)}) + \mathrm{pen}(k)) > \varepsilon\right] \\
&= \sum_{k=1}^{\infty} \mathbb{P}\left[R(\hat{f}_n^{(k)}) - U_{n,k} > \sqrt{\frac{\log k}{m}} + \varepsilon\right] \\
&\leq \sum_{k=1}^{\infty} c \exp\left(-2m(\varepsilon + \sqrt{\log k/m})^2\right) \\
&\leq \sum_{k=1}^{\infty} c \exp\left(-2m(\varepsilon^2 + \log k/m)\right) \\
&= c\exp(-2n\varepsilon^2) \sum_{k=1}^{\infty} \frac{1}{k^2} \leq 2c\exp(-2m\varepsilon^2) \ .
\end{aligned}$$

By integrating the final upper bound with respect to ε, we get the bound

$$\mathbb{E}\left[R(\tilde{f}_n) - (\widehat{R}(\tilde{f}_n) + \mathrm{pen}(\hat{k}))\right] \leq \sqrt{\frac{\log(ce)}{2m}} \ . \tag{4}$$

Also, for any k, we have

$$\begin{aligned}
\mathbb{E}\left[\widehat{R}(\tilde{f}_n) + \mathrm{pen}(\hat{k}) - R(f^\star_{\mathcal{F}^{(k)}})\right] &\leq \mathbb{E}\left[\widehat{R}(\hat{f}_n^{(k)}) + \mathrm{pen}(k) - R(f^\star_{\mathcal{F}^{(k)}})\right] \\
&\leq \mathbb{E}\left[\widehat{R}(f^\star_{\mathcal{F}^{(k)}}) + \mathrm{pen}(k) - R(f^\star_{\mathcal{F}^{(k)}})\right] \\
&= \mathbb{E}\left[\mathrm{pen}(k)\right] \ .
\end{aligned}$$

The first inequality above follows by definition of $\hat{k}$ and $\tilde{f}_n$. The second holds because $\hat{f}_n^{(k)}$ minimizes the empirical risk over $\mathcal{F}^{(k)}$. Summing this with (4), we get, for any k,

$$\mathbb{E}\left[R(\tilde{f}_n)\right] \leq \mathbb{E}\left[\mathrm{pen}(k)\right] + R(f^\star_{\mathcal{F}^{(k)}}) + \sqrt{\frac{\log(ce)}{2m}} \ .$$

Now subtracting $R(f^\star)$ from both sides and minimizing over k proves the theorem. $\square$

4.3 Data Driven Penalties Using Rademacher Averages

The results of Section 3.2 tell us that there exists universal constants C, c such that

$$\mathbb{P}\left[R(\hat{f}_n) > \widehat{R}(\hat{f}_n) + 2\widehat{\mathfrak{R}}_n\left(\mathcal{F}^{(k)}\right) + \varepsilon\right] < c\exp(-2Cn\varepsilon^2) \ .$$

Thus, we can use the data dependent penalty

$$\mathrm{pen}(k) = 2\widehat{\mathfrak{R}}_n\left(\mathcal{F}^{(k)}\right) + \sqrt{\frac{\log k}{Cn}}$$

in Theorem 18 to get the bound

$$\mathbb{E}\left[R(\tilde{f}_n)\right] - R(f^\star) \leq \min_k \left\{2\mathfrak{R}_n\left(\mathcal{F}^{(k)}\right) + \left(R(f^\star_{\mathcal{F}^{(k)}}) - R(f^\star)\right) + \sqrt{\frac{\log k}{Cn}}\right\} + \sqrt{\frac{\log(ce)}{2Cn}} \ .$$

Note that the penalty here is data dependent. Moreover, except for the $\sqrt{\frac{\log k}{Cn}}$ term, the penalized estimator makes the optimal trade-off between the Rademacher complexity, an upper bound on the estimation error, and the approximation error.

4.4 Hold-out Estimates

Suppose out of $m + n$ iid samples we set aside m samples where m is much smaller than n. Since the hold-out set of size m is not used in the computation of $\hat{f}_n^{(k)}$, we can estimate $R(\hat{f}_n^{(k)})$ by the hold-out estimate

$$U_{n,k} = \sum_{i=1}^{m} \ell(\hat{f}_n^{(k)}(X_{n+i}), Y_{n+i}) \ .$$

By Hoeffding's inequality, the assumption of Theorem 18 holds with $c = 1$. Moreover $\mathbb{E}\left[U_{n,k}\right] = R(\hat{f}_n^{(k)})$. Therefore, we have

$$\mathbb{E}\left[R(\tilde{f}_n)\right] - R(f^\star) \leq \min_k \left\{\mathbb{E}\left[R(\hat{f}_n^{(k)}) - \widehat{R}(\hat{f}_n^{(k)})\right] + \left(R(f^\star_{\mathcal{F}^{(k)}}) - R(f^\star)\right) + \sqrt{\frac{\log k}{m}}\right\} + \sqrt{\frac{1}{2m}} \ .$$

5 Alternatives to Uniform Convergence

The approach of deriving generalization bounds using uniform convergence arguments is not the only one possible. In this section, we review three techniques for proving generalization bounds based on sample compression, algorithmic stability, and PAC-Bayesian analysis respectively.

5.1 Sample Compression

The sample compression approach to proving generalization bounds was introduced by Warmuth (1997). It applies to algorithms whose output $\hat{f}_{n,S}$ (that is learned using some training set S) can be reconstructed from a compressed subset $S_{\mathbf{i}}$. Here $\mathbf{i}$ is a sequence of indices

$$\mathbf{i} = (i_1, i_2, \ldots, i_k),\ 1 \leq i_1 < i_2 < \ldots < i_k \leq n\ ,$$

and $S_{\mathbf{i}}$ is simply the subsequence of S indexed by $\mathbf{i}$. For the index sequence $\mathbf{i}$ above, we say that its length $|\mathbf{i}|$ is k. To formalize the idea that the algorithms output can be reconstructed from a small number of examples, we will define a *compression function*

$$\mathcal{C} : \bigcup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{I}$$

where $\mathcal{I}$ is the set of all possible index sequences of finite length. It is assumed that $|\mathcal{C}(S)| \leq n$ for an S is of size n. We also define a *reconstruction function*

$$\mathcal{R} : \bigcup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}\ .$$

We assume that the learning rule satisfies, for any S,

$$\hat{f}_{n,S} = \mathcal{R}\left(S_{\mathcal{C}(S)}\right)\ . \tag{5}$$

This formalizes our intuition that to reconstruct $\hat{f}_{n,S}$ it suffices to remember only those examples whose indices are given by $\mathcal{C}(S)$.

The following result gives a generalization bound for such an algorithm.

Theorem 19. *Let $\hat{f}_n$ be a learning rule for which there exists compression and reconstructions functions $\mathcal{C}$ and $\mathcal{R}$ respectively such that the equality* (5) *holds for all S. Then, we have, with probability at least $1-\delta$,*

$$R(\hat{f}_n) \leq \frac{n}{n-k}\widehat{R}(\hat{f}_n) + \sqrt{\frac{\log\binom{n}{k} + \log\frac{n}{\delta}}{2(n-k)}}$$

where $k = |\mathcal{C}(S)|$.

Proof. The proof uses the simple idea that for a fixed index set $\mathbf{i}$, $S_{\bar{\mathbf{i}}}$ is a sample of size $n - |\mathbf{i}|$ that is independent of $S_{\mathbf{i}}$. Here $\bar{\mathbf{i}}$ denotes the sequence $[n]\backslash\mathbf{i}$. By this independence, the risk of any function that depends only on $S_{\mathbf{i}}$ is close to its empirical average on $S_{\bar{\mathbf{i}}}$ by Hoeffding's inequality.

Denote the empirical risk of f calculated on a subset $S_{\mathbf{i}}$ of S by $\widehat{R}_{\mathbf{i}}(f)$. We have, for any f,

$$\widehat{R}_{\overline{\mathcal{C}(S)}}(f) \leq \frac{n}{n-k}\widehat{R}(f) \ ,$$

where $k = |\mathcal{C}(S)|$. Thus, all we need to show is

$$\mathbb{P}\left[R(\hat{f}_n) - \widehat{R}_{\overline{\mathcal{C}(S)}}(\hat{f}_n) \geq \varepsilon_{n,|\mathcal{C}(S)|}\right] \leq \delta$$

where

$$\varepsilon_{n,k} = \sqrt{\frac{\log\binom{n}{k} + \log\frac{n}{\delta}}{2(n-k)}} \ .$$

To show this, we proceed as follows,

$$\begin{aligned}
&\mathbb{P}\left[R(\hat{f}_n) - \widehat{R}_{\overline{\mathcal{C}(S)}}(\hat{f}_n) \geq \varepsilon_{n,|\mathcal{C}(S)|}\right] \\
&= \mathbb{P}\left[R(\mathcal{R}(S_{\mathcal{C}(S)})) - \widehat{R}_{\overline{\mathcal{C}(S)}}(\mathcal{R}(S_{\mathcal{C}(S)})) \geq \varepsilon_{n,|\mathcal{C}(S)|}\right] \\
&\leq \mathbb{P}\left[\exists \mathbf{i} \text{ s.t. } |\mathbf{i}| = k,\ R(\mathcal{R}(S_{\mathbf{i}})) - \widehat{R}_{\bar{\mathbf{i}}}(\mathcal{R}(S_{\mathbf{i}})) \geq \varepsilon_{n,k}\right] \\
&\leq \sum_{\mathbf{i}:|\mathbf{i}|=k} \mathbb{P}\left[R(\mathcal{R}(S_{\mathbf{i}}) - \widehat{R}_{\bar{\mathbf{i}}}(\mathcal{R}(S_{\mathbf{i}})) \geq \varepsilon_{n,k}\right] \\
&\leq \sum_{k=1}^{n} \sum_{\mathbf{i}:|\mathbf{i}|=k} \exp(-2(n-k)\varepsilon_{n,k}^2) \\
&= \sum_{k=1}^{n} \binom{n}{k} \exp(-2(n-k)\varepsilon_{n,k}^2) \\
&\leq \sum_{k=1}^{n} \frac{\delta}{n} = \delta \ .
\end{aligned}$$

The last step holds because, by our choice of $\varepsilon_{n,k}$,

$$\binom{n}{k} \exp(-2(n-k)\varepsilon_{n,k}^2) = \frac{\delta}{n} \ .$$

□

5.2 Stability

Stability is property of a learning algorithm that ensures that the output, or some aspect of the output, of the learning algorithm does not change much if the training data set changes very slightly. This is an intuitive notion whose roots

in learning theory go back to the work of Tikhonov on regularization of inverse problems. To see how stability can be used to give generalization bounds, let us consider a result of Bousquet and Elisseeff (2002) that avoids going through uniform convergence arguments by directly showing that any learning algorithm with *uniform stability* enjoys exponential tail generalization bounds in terms of both the training error and the leave-one-out error of the algorithm.

Let $S^{\setminus i}$ denote the sample of size $n-1$ obtained by removing a particular input (x_i, y_i) from a sample S of size n. A learning algorithm is said to have *uniform stability* β if for any $S \in (\mathcal{X} \times \mathcal{Y})^n$ and any $i \in [n]$, we have,

$$\sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |\ell(\hat{f}_{n,S}(x), y) - \ell(\hat{f}_{n-1,S^{\setminus i}}(x), y)| \leq \beta \ .$$

We have the following generalization bound for a uniformly stable learning algorithm.

Theorem 20. *Let $\hat{f}_n$ have uniform stability β. Then, with probability at least $1-\delta$,*

$$R(\hat{f}_n) \leq \widehat{R}(\hat{f}_n) + 2\beta + (4\beta n + 1)\sqrt{\frac{\log \frac{1}{\delta}}{2n}} \ .$$

Note that this theorem gives tight bound when β scales as $1/n$. As such it cannot be applied directly to the classification setting with 0-1 loss where β can only be 0 or 1. A good example of a learning algorithm that exhibits uniform stability is regularized ERM over a reproducing kernel Hilbert space using a convex loss function that has Lipschitz constant C_ℓ:

$$\forall y, y', y'' \in \mathcal{Y} \subseteq \mathbb{R}, \ |\ell(y', y) - \ell(y'', y)| \leq C_\ell \cdot |y' - y''| \ .$$

A regularized ERM using kernel K is defined by

$$\hat{f}_n = \operatorname*{argmin}_{f\in\mathcal{F}_K} \ \widehat{R}(f) + \lambda \|f\|_K^2 \ ,$$

where λ is a regularization parameter. Regularized ERM has uniform stability β for $\beta = \frac{C_\ell^2 \kappa^2}{2\lambda n}$ where

$$\kappa = \sup_{x\in\mathcal{X}} \sqrt{K(x,x)} \ .$$

The literature on stability of learning algorithms is unfortunately thick with definitions. For instance, Kutin and Niyogi (2002) alone consider over a dozen variants! We chose a simple (but quite strong) definition from Bousquet and Elisseeff above to illustrate the general point that stability can be used to derive generalization bounds. But we like to point the reader to Mukherjee et al. (2006) where a notion of *leave-one-out (LOO) stability* is defined and shown sufficient for generalization for any learning algorithm. Moreover, for ERM they show it is necessary and sufficient for both generalization and learnability with respect to a fixed class. Shalev-Shwartz et al. (2010) further examine the connections between stability, generalization and learnability in Vapnik's general learning setting that includes supervised learning (learning from example, label pairs) as a special case.

5.3 PAC-Bayesian Analysis

PAC-Bayesian analysis refers to a style of analysis of learning algorithms that output not just a single function $f \in \mathcal{F}$ but rather a *distribution* (called a "posterior distribution") over $\mathcal{F}$. Moreover, the complexity of the data-dependent posterior distribution is measured relative to a fixed distribution chosen in advance (that is, in a data independent way). The fixed distribution is called a "prior distribution". Note that the terms "prior" and "posterior" are borrowed from the analysis of Bayesian algorithms that start with a prior distribution on some parameter space and, on seeing data, make updates to the distribution using Bayes' rule to obtain a posterior distribution. Even though the terms are borrowed, their use is not required to be similar to their use in Bayesian analysis. In particular, the prior used in the PAC-Bayes theorem need not be "true" in any sense and the posterior need not be obtained using Bayesian updates: the bound holds for any choice of the prior and posterior.

Denote the space of probability distributions over $\mathcal{F}$ by $\Delta(\mathcal{F})$. There might be measurability issues in defining this for general function classes. So, we can assume that $\mathcal{F}$ is a countable class for the purpose of the theorem below. Note, however, that the cardinality of the class $\mathcal{F}$ never enters the bound explicitly anywhere. For any $\rho \in \Delta(\mathcal{F})$, define

$$R(\rho) = \mathbb{E}_{f\sim\rho}\left[R(f)\right] \ ,$$
$$\widehat{R}(\rho) = \mathbb{E}_{f\sim\rho}\left[\widehat{R}(f)\right] \ .$$

In the context of classification, a randomized classifier that, when asked for a prediction, samples a function from a distribution ρ and uses if for classification, is called a *Gibbs classifier*. Note that $R(\rho)$ is the risk of such a Gibbs classifier.

Also recall the definition of *Kullback-Leibler divergence* or *relative entropy*:

$$D\left(\rho||\pi\right) = \int_{f\in\mathcal{F}} \log\frac{\rho(f)}{\pi(f)} d\rho(f) \ .$$

For real numbers $p, q \in [0,1]$ we define (with some abuse of notation):

$$D\left(p||q\right) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q} \ .$$

With these definitions, we can now state the PAC Bayesian theorem (McAllester, 2003).

Theorem 21 (PAC-Bayesian Theorem)**.** *Let π be a fixed distribution over $\mathcal{F}$. Then, we have, with probability at least $1-\delta$,*

$$\forall \rho \in \Delta(\mathcal{F}),\ D\left(\widehat{R}(\rho)\middle|\middle|R(\rho)\right) \leq \frac{D\left(\rho||\pi\right) + \log\frac{2n}{\delta}}{n-1} \ .$$

In particular, for any learning algorithm that, instead of returning a single function $\hat{f}_n \in \mathcal{F}$, returns a distribution $\hat{\rho}_n$ over $\mathcal{F}$, we have, with probability at least $1-\delta$,

$$D\left(\widehat{R}(\hat{\rho}_n)\middle|\middle|R(\hat{\rho}_n)\right) \leq \frac{D\left(\hat{\rho}_n||\pi\right) + \log\frac{2n}{\delta}}{n-1} \ .$$

To get more interpretable bounds from this statement, the following inequality is useful for $0 \leq p < q \leq 1$:

$$\frac{(p-q)^2}{2} \leq \frac{(p-q)^2}{2q} \leq D\left(p||q\right) \ .$$

This implies that if $D\left(p||q\right) \leq x$ then two inequalities hold:

$$q \leq p + \sqrt{2x} \ ,$$
$$q \leq p + \sqrt{2px} + 2x \ .$$

The first gives us a version of the PAC-Bayesian bound that is often presented:

$$R(\hat{\rho}_n) - \widehat{R}(\hat{\rho}_n) \leq \sqrt{\frac{2(D\left(\hat{\rho}_n||\pi\right) + \log\frac{2n}{\delta})}{n-1}} \ .$$

The second gives us the interesting bound

$$R(\hat{\rho}_n) - \widehat{R}(\hat{\rho}_n) \leq \sqrt{\frac{2\widehat{R}(\hat{\rho}_n)(D\left(\hat{\rho}_n||\pi\right) + \log\frac{2n}{\delta})}{n-1}} + 2\frac{D\left(\hat{\rho}_n||\pi\right) + \log\frac{2n}{\delta}}{n-1} \ .$$

If the loss $\widehat{R}(\hat{\rho}_n)$ is close to zero, then the dominating term is the second term that scales as $\frac{1}{n}$. If not, then the first term, which scales as $\frac{1}{\sqrt{n}}$, dominates.

6 Computational Aspects

So far we have only talked about sample complexity issues ignoring considerations of *computational complexity*. To properly address the latter, we need a formal *model of computation* and need to talk about how functions are represented by the learning algorithm. The branch of machine learning that studies these questions is called *computational learning theory*. We will now provide a brief introduction to this area.

6.1 The PAC Model

The basic model in computational learning theory is the Probably Approximately Correct (PAC) model. It applies to learning binary valued functions and uses the 0-1 loss. Since a ± 1 valued function is specified unambiguously be specifying the subset of $\mathcal{X}$ where it is one, we have a bijection between binary valued functions and *concepts*, which are simply subsets of $\mathcal{X}$. Thus we will use (binary valued) function and concept interchangeably in this section. Moreover, the basic PAC model considers what is sometimes called the *realizable* case. That is, there is a "true" concept in $\mathcal{F}$ generating the data. This means that $y_i = f(x_i)$ for some $f \in \mathcal{F}$. Hence the minimal risk in the class is zero: $R(f^\star_{\mathcal{F}}) = 0$. Note, however, that the distribution over $\mathcal{X}$ is still assumed to be arbitrary.

To define the computational complexity of a learning algorithm, we assume that $\mathcal{X}$ is either $\{0,1\}^d$ or $\mathbb{R}^d$. We assume a model of computation that can store a single real number in a memory location and can perform any basic arithmetic operation (additional, subtraction, multiplication, division) on two real numbers in one unit of computation time.

The distinction between a concept and its representation is also an important one when we study computational complexity. For instance, if our concepts are convex polytopes in $\mathbb{R}^d$, we may choose a representation scheme that specifies the vertices of the polytope. Alternatively, we may choose to specify the linear equalities describing the faces of the polytope. Ther vertex-based and face-based representations can differ exponentially in size. Formally, a *representation scheme* is a mapping $\text{rep} : (\Sigma \cup \mathbb{R})^\star \to \mathcal{F}$ that maps a representation (consisting of symbols from a finite alphabet Σ and real numbers) to a concept. As noted above, there could be many σ's such that $\text{rep}(\sigma) = f$ for a given concept f. The size of representation is assumed to be given by a function $\text{size} : (\Sigma \cup \mathbb{R})^\star \to \mathbb{N}$. A simple choice of size could be simply the number of symbols and real numbers that appear in the representation. Finally, the size of a concept f is simply the size of its smallest representation:

$$\text{size}(f) = \min_{\sigma:\text{rep}(\sigma)=f} \text{size}(\sigma) \ .$$

The last ingredient we need before can give the definition of efficient learnability in the PAC model is the representation class $\mathcal{H}$ used by the algorithm for producing its output. We call $\mathcal{H}$ the *hypothesis* class and it is assumed that there is a representation in $\mathcal{H}$ for every function in $\mathcal{F}$. Moreover, it is required that for every $x \in \mathcal{X}$ and any $h \in \mathcal{H}$, we can evaluate $h(x)$ $(= \text{rep}(h)(x))$ in time polynomial in d and $\text{size}(h)$.

Recall that we have assumed $\mathcal{X} = \{0,1\}^d$ or $\mathbb{R}^d$. We say that an algorithm *efficiently PAC learns* $\mathcal{F}$ using hypothesis class $\mathcal{H}$ if, for any $f \in \mathcal{F}$, given access to iid examples $(x_i, f(x_i)) \in \mathcal{X} \times \{\pm 1\}$, inputs $\varepsilon \in (0,1)$ (accuracy), $\delta \in (0,1)$ (confidence), and $\text{size}(f)$, the algorithm satisfies the following conditions.

1. With probability at least $1-\delta$, it outputs a hypothesis $h \in \mathcal{H}$ with $R(h) \leq \varepsilon$.

2. It runs in time polynomial in $\frac{1}{\varepsilon}$, $\frac{1}{\delta}$, $\text{size}(f)$ and d.

For examples of classes $\mathcal{F}$ that are efficiently PAC learnable (using some $\mathcal{H}$), see the texts Natarajan (1991); Kearns and Vazirani (1994); Anthony and Biggs (1997).

6.2 Weak and Strong Learning

In the PAC learning definition, the algorithm must be able to achieve arbitrarily small values of ε and δ. A weaker definition would require the learning algorithm to work only for fixed choices of ε and δ. We can modify the definition of PAC learning by removing ε and δ from the input to the algorithm. Instead,

we assume that there are fixed polynomials $p_\delta(\cdot,\cdot)$ and $p_\varepsilon(\cdot,\cdot)$ such that the algorithm outputs $h \in \mathcal{H}$ that satisfies

$$R(h) \leq \frac{1}{2} - \frac{1}{p_\varepsilon(d, \text{size}(f))}$$

with probability at least

$$\frac{1}{p_\delta(d, \text{size}(f))} .$$

Such an algorithm is called a *weak PAC learning* algorithm for $\mathcal{F}$. The original definition, in contrast, can be called a definition of *strong PAC learning*. The definition of weak PAC learning does appear quite weak. Recall that an accuracy of $\frac{1}{2}$ can be trivially achieved by an algorithm that does not even look at the samples but simply outputs a hypothesis that outputs a random ± 1 sign (each with probability a half) for any input. The desired accuracy above is just an inverse polynomial away from the trivial accuracy guarantee of $\frac{1}{2}$. Moreover, the probability with which the weak accuracy is achieved is not required to be arbitrarily close to 1.

A natural question to ask is whether a class that is weakly PAC learnable using $\mathcal{H}$ is also strongly learnable using $\mathcal{H}$? The answer, somewhat surprisingly, is "Yes". So, the weak PAC learning definition only appears to be a relaxed version of strong PAC learning. Moreover, a weak PAC learning algorithm, given as a subroutine or blackbox, can be converted into a strong PAC learning algorithm. Such a conversion is called *boosting*.

A boosting procedure has to start with a weak learning algorithm and boost two things: the confidence and the accuracy. It turns out that boosting the confidence is easier. The basic idea is to run the weak learning algorithm several times on independent samples. Therefore, the crux of the boosting procedure lies in boosting the accuracy. Boosting the accuracy seems hard until one realizes that the weak learning algorithm does have one strong property: it is guaranteed to work no matter what the underlying distribution of the examples is. Thus, if a first run of the weak learning only produces a hypothesis h_1 with slightly better accuracy than $\frac{1}{2}$, we can make the second run focus only on those examples where h_1 makes a mistake. Thus, we will focus the weak learning algorithm on "harder portions" of the input distribution. While this simple idea does not directly work, a variation on the same theme was shown to work by Schapire (1990). Freund (1995) then presented a simpler boosting algorithm. Finally, a much more practical boosting procedure, called AdaBoost, was discovered by them jointly (Freund and Schapire, 1995). AdaBoost counts as one of the great practical success stories of computational learning theory. For more details on boosting, AdaBoost, and the intriguing connections of these ideas to game theory and statistics, we refer the reader to Freund and Schapire (1996).

6.3 Random Classification Noise and the SQ Model

Recall that the basic PAC model assumes that the labels are generated using a function f belonging to the class $\mathcal{F}$ under consideration. Our earlier devel-

opment of statistical tools for obtaining sampling complexity estimates did not require such an assumptions (though the rates did depend on whether $R(f^\star_{\mathcal{F}}) = 0$ or not).

There are various *noise models* that extend the PAC model by relaxing the noise-free assumption and thus making the problem of learning potentially harder. Perhaps the simplest is the *random classification noise* model (Angluin and Laird, 1987). In this model, when the learning algorithm queries for the next labeled example of the form $(x_i, f(x_i))$, it receives it with probability $1-\eta$ but with probability η, it receives $(x_i, -f(x_i))$. That is, with some probability $\eta < \frac{1}{2}$, the label is flipped. The definition of efficient learnability remains the same except that now the learning algorithm has to run in time polynomial in $\frac{1}{\varepsilon}, \frac{1}{\delta}, d, \text{size}(f)$ and $\frac{1}{1-2\eta_0}$ where $\eta_0 < \frac{1}{2}$ is an input to the algorithm and is an upper bound on η.

Kearns (1998) introduced the statistical query (SQ) model where learning algorithms do not directly access the labeled example but only make statistical queries about the underlying distribution. The queries take the form (χ, τ) where $\chi : \mathcal{X} \times \{\pm 1\} \to \{\pm 1\}$ is a binary valued function of the labeled examples and $\tau \in (0, 1]$ is a tolerance parameter. Given such a query, the oracle in the SQ model responds with an estimate of $P(\chi(x, f(x)) = +1)$ that is accurate to within an additive error of τ. It is easy to see, using Hoeffding's inequality, that given access to the usual PAC oracle that provides iid examples of the form $(x_i, f(x_i))$, we can simulate the SQ oracle by just drawing a sample of size polynomial in $\frac{1}{\tau^2}$ and $\log \frac{1}{\delta}$ and estimating the probability $P(\chi(x, f(x)) = 1)$ based on it. The simulation with succeed with probability $1 - \delta$. Kearns gave a more complicated simulation that mimics the SQ oracle given access to only a PAC oracle with random missclassification noise.

Lemma 22. *Given a query (χ, τ), the probability $P(\chi(x, f(x)) = 1)$ can be estimated to within τ error, with probability at least $1 - \delta$, using*

$$O\left(\frac{1}{\tau^2(1-2\eta)^2} \log \frac{1}{\delta}\right)$$

examples that have random misclassification noise at rate $\eta < \frac{1}{2}$ in them.

This lemma means that learning algorithm that learns a concept class $\mathcal{F}$ in the SQ model can also learn $\mathcal{F}$ in the random classification noise model.

Theorem 23. *If there is an algorithm that efficiently learns $\mathcal{F}$ from statistical queries using $\mathcal{H}$, then there is an algorithm algorithm that efficiently learns $\mathcal{F}$ using $\mathcal{H}$ in the random classification noise model.*

6.4 The Agnostic PAC Model

The agnostic PAC model uses the same definition of learning as the one we used in Section 2.3). That is, the distribution of examples (x, y) is arbitrary. In particular, it is not assumed that $y = f(x)$ for any f. The goal of the

learning algorithm is to output a hypothesis $h \in \mathcal{H}$ with $R(h) - R(f^\star_{\mathcal{F}}) \leq \varepsilon$ with probability at least $1 - \delta$. We say that a learning algorithm efficiently learns $\mathcal{F}$ using $\mathcal{H}$ in the agnostic PAC model if the algorithm not only outputs a probably approximately correct hypothesis h but also runs in time that is polynomial in $\frac{1}{\varepsilon}$, $\frac{1}{\delta}$ and d.

The agnostic model has proved to be a very difficult one for exhibiting efficient learning algorithms. Note that the statistical issue is completely resolved: a class $\mathcal{F}$ is learnable iff it has finite VC dimension. However, the "algorithm" inplicit in this statement is ERM which, even for the class of halfspaces

$$\{x \mapsto \text{sign}(\langle w, x\rangle - \theta) \ : \ w \in \mathbb{R}^d, \ \theta \in \mathbb{R}\},$$

leads to NP-hard problems. For instance, see Feldman et al. (2009) for very strong negative results about agnostic learning of halfspaces. However, these results apply to *proper* agnostic learning only. That is, the algorithm outputs a hypothesis that is also a halfspace. An efficient algorithm for learning halfspaces in the agnostic PAC model using a general hypothesis class would be a major breakthrough.

6.5 The Mistake Bound Model

The mistake bound model is quite different from the PAC model. Unlike the PAC model, no assumption is made on the distribution of the inputs. There is also no distinction between a training phase where the learner uses a batch of examples to learn a hypothesis and a test phase where the risk of the learned hypothesis determines the learner's expected loss. Instead, learning happens in a series of trials (or rounds) in an *online* fashion. At any given round $t \in \mathbb{N}$, the order of events is as follows:

- Learner receives $x_t \in \mathcal{X}$ where $\mathcal{X} = \{0,1\}^d$ or $\mathbb{R}^d$
- Learner outputs a label $\hat{y}_t \in \mathcal{Y}$
- Learner receives the correct label $y_t = f(x_t)$

Note that the basic model assumes that is a true function f generating the labels (the case when such an f is assumed to exist is often called the *realizable case*). The identity of this function is, of course, hidden from the learner. The goal of the learner is to minimize the total number of *mistakes* it makes:

$$\sum_{t=1}^{\infty} \ell(\hat{y}_t, f(x_t))$$

where ℓ is the 0-1 loss. If a learning algorithm can guarantee, for any choice $f \in \mathcal{F}$ of the true concept, that the total mistake bound will be a polynomial function of d and size(f) as well as the computation per round is polynomial in the same parameters, then we say that the algorithm *efficiently learns $\mathcal{F}$ in the mistake bound model*.

Recall that, in the PAC model, VC dimension of $\mathcal{F}$ characterizes learnability of $\mathcal{F}$ if we ignore computational considerations. Moroever, VC dimension characterizes learnability in the agnostic PAC model as well. In the mistake bound model, it is the Littlestone dimension (Littlestone, 1987) whose finiteness provides a necessary and sufficient condition for learnability. There is also an *agnostic version* of the mistake bound model where we drop the assumption that $y_t = f(x_t)$ for some $f \in \mathcal{F}$. It is also known that the Littlestone dimension characterizes learnability (ignoring efficiency) even in the agnostic online mistake bound model (Ben-David et al., 2009).

The Littlestone dimension of a class $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{X}}$ is defined as follows. A *$\mathcal{X}$-valued tree* (or simply an $\mathcal{X}$-tree) of height t is a complete binary tree of height t whose *internal* nodes are labeled with instances from $\mathcal{X}$. A (complete binary) tree of height t has exactly 2^t leaves which we identify, from left to right, with the 2^t sequences length t ± 1 sequences ordered lexicographically (with -1 taking precedence over $+1$ in determining the lexicographic order). For example, the leftmost leaf correponds to all -1's sequence whereas the rightmost leaf corresponds to the all $+1$'s sequences. If λ is a leaf, we denote its associated ± 1 sequence by $\epsilon(\lambda)$.

An $\mathcal{X}$-tree of height t is said to be *shattered* by $\mathcal{F}$ if for each of the 2^t leaves the following is true: there is a function $f \in \mathcal{F}$ such that the sequence of values taken by f on the internal nodes on the path from the root to the leaf λ is exactly $\epsilon(\lambda)$. The *Littlestone dimension* of $\mathcal{F}$ is the height of the tallest tree that is shattered by $\mathcal{F}$.

It is easy to see that if an $\mathcal{X}$-valued sequence $(x_1, \ldots, x_t)$ is shattered by $\mathcal{F}$ then the $\mathcal{X}$-tree of height t obtained by repeating x_i across level i for $i \in [t]$ is also shattered by $\mathcal{F}$. This proves the following relationship between $\mathrm{VCdim}(\mathcal{F})$ and $\mathrm{Ldim}(\mathcal{F})$.

Theorem 24. *For any $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{X}}$ we have*

$$\mathrm{VCdim}(\mathcal{F}) \leq \mathrm{Ldim}(\mathcal{F}) \ .$$

A converse to the above theorem is not possible. Indeed, the class of theshold functions on $\mathbb{R}$:

$$\mathcal{F} = \{x \mapsto \mathrm{sign}(x - \theta) \ : \ \theta \in \mathbb{R}\}$$

has $\mathrm{VCdim}(\mathcal{F}) = 1$ but $\mathrm{Ldim}(\mathcal{F}) = \infty$. A finite class $\mathcal{F}$ has a finite Littlestone dimension satisfying the upper bound $\mathrm{Ldim}(\mathcal{F}) \leq \log_2(|\mathcal{F}|)$.

6.5.1 Halving and Weighted Majority

Note that we proved risk bounds for a finite class in the probabilistic framework using Hoeffding's inequality along with a simple union bound. Here we show how to deal with a finite class in the online mistake bound model. In the realizable case, a simple algorithm called *halving* is guaranteed to make no more than $\log_2(|\mathcal{F}|)$ mistakes.

Algorithm 1 Halving

Initialize $\mathcal{F}_0 = \mathcal{F}$
for $t = 1, 2, \ldots$ **do**
  Receive x_t
  Predict $\hat{y}_t$ using a majority vote over $\mathcal{F}_{t-1}$
  Receive true label y_t
  if $\hat{y} \neq y_t$ **then**
    $\mathcal{F}_t = \{f \in \mathcal{F}_{t-1} \ : \ f(x_t) = y_t\}$
  end if
end for

At the beginning of round t, the halving algorithm considers a subset $\mathcal{F}_t \subseteq \mathcal{F}$ of the original function class. Predictions are made by taking a majority vote over $\mathcal{F}_{t-1}$:

$$\hat{y}_t = \text{sign}\left(\sum_{f \in \mathcal{F}_{t-1}} f(x_t)\right) \ .$$

Therefore, if a mistake is made, then $|\mathcal{F}_{t-1}|$ reduces by at least a factor of 2. Moreover, the true function is never eliminated since it will always satisfy $f(x_t) = y_t$. Therefore, if M_t is the number of mistakes made by the halving algorithm in t rounds, then we have

$$1 \leq |\mathcal{F}_t| \leq \frac{|\mathcal{F}_0|}{2^{M_t}} = \frac{|\mathcal{F}|}{2^{M_t}} \ .$$

This implies that $M_t \leq \log_2(|\mathcal{F}|)$. Thus we have proved a mistake bound for halving.

Theorem 25. *The halving algorithm when run on a finite class $\mathcal{F}$ enjoys a mistake bound of* $\log_2(|\mathcal{F}|)$.

In the non-realizable case, halving does not work directly but a suitable modification does. Since no function in the class can be guaranteed to be correct all the time, it does not make sense to eliminate functions that make mistakes. The crucial idea here is to keep *weights* for different functions in the class. Then, when a functions makes a mistake, it is not eliminated but its weight is multiplied by a factor of $e^{-\eta} < 1$ where $\eta > 0$ is a parameter. The prediction are now taken using a *weighted majority* vote over all $\mathcal{F}$:

$$\hat{y}_t = \text{sign}\left(\sum_{f \in \mathcal{F}} \rho_{t-1}(f) f(x_t)\right) \ . \tag{6}$$

Theorem 26. *At any the end of any given round, let M_f be the number of mistakes made by a function $f \in \mathcal{F}$ so far. Then the number of mistakes M*

Algorithm 2 Weighted Majority

Initialize $\rho_0(f) = \pi(f)$ for some distribution π over $\mathcal{F}$
for $t = 1, 2, \ldots$ **do**
  Receive x_t
  Predict $\hat{y}_t$ using a ρ_{t-1}-weighted majority vote over $\mathcal{F}_{t-1}$
  Receive true label y_t
  for $f \in \mathcal{F}$ **do**
    $\rho_t(f) = \rho_{t-1}(f) \exp(-\eta \ell(f(x_t), y_t))$
  end for
end for

made by weighted majority so far is bounded as

$$\forall f \in \mathcal{F},\ M \leq 2 \cdot \frac{\eta M_f + \log \frac{1}{\pi(f)}}{1 - e^{-\eta}} \ .$$

In particular, using the uniform distribution $\pi(f) = 1/|\mathcal{F}|$, we get

$$\forall f \in \mathcal{F},\ M \leq 2 \cdot \frac{\eta M_f + \log |\mathcal{F}|}{1 - e^{-\eta}} \ .$$

Proof. Fix a round index $T \geq 1$ and let M and M_f be defined as the numbers of mistakes in the first T rounds. Let F_t be the fraction of weights on the functions that made a mistake on round $t \leq T$. Since we make a mistake if and only if $F_t \geq \frac{1}{2}$, we have

$$M = \sum_{t=1}^{T} \mathbf{1}\left[F_t \geq \tfrac{1}{2}\right] \leq \sum_{t=1}^{T} 2F_t \ . \tag{7}$$

Define the total weight

$$W_t = \sum_{f \in \mathcal{F}} \rho_t(f) \ .$$

At round t, the total weight changes as

$$W_{t+1} \leq (1 - (1 - e^{-\eta})F_t) W_t \ .$$

Hence the final total weight is

$$W_{T+1} = W_0 \prod_{t=1}^{T} (1 - (1 - e^{-\eta})F_t) \ .$$

Note that $W_0 = 1$. If the function f made M_f mistakes, its weight is

$$\rho_{t+1}(f) = \pi(f) e^{-\eta M_f} \ .$$

Since $\rho_{t+1}(f) \leq W_{t+1}$, we have

$$\pi(f) e^{-\eta M_f} \leq \prod_{t=1}^{T} (1 - (1 - e^{-\eta})F_t) \ .$$

Taking negative logs of both sides reverses the inequality giving

$$\log \frac{1}{\pi(f)} + \eta M_f \geq -\sum_{t=1}^{T} \log(1 - (1 - e^{-\eta})F_t) \ .$$

Using the elementary but useful inequality $-\log(1-x) > x$ gives

$$(1 - e^{-\eta}) \sum_{t=1}^{T} F_t \leq \eta M_f + \log \frac{1}{\pi(f)} \ .$$

Combining this with the bound (7) proves the theorem. □

It turns out we can shave off a factor of 2 in the above mistake bound by using a randomized version of the weighted majority algorithm. Instead of taking the majority vote with the weights $\rho_{t-1}(f)$, we can consider a Gibbs classifier that chooses a function f_t randomly from $\mathcal{F}$ with probability proportional to $\rho_{t-1}(f)$ and then outputs $\hat{y}_t = f_t(x_t)$. This makes the number of mistakes a random variable but the bound (7) turns into an equality without the factor of 2. That is, for the randomized weighted majority algorithm

$$\mathbb{E}\left[M\right] = \sum_{t=1}^{T} F_t \ .$$

Thus we have the following result.

Theorem 27. *Suppose that we change the weighted majority step* (6) *to the randomized weighted majority step*

$$f_t \sim \rho_{t-1}, \ \hat{y}_t = \text{sign}(f_t(x_t)) \ .$$

Then, this randomized version of weighted majority satisfies the expected mistake bound

$$\forall f \in \mathcal{F}, \ \mathbb{E}\left[M\right] \leq \frac{\eta M_f + \log \frac{1}{\pi(f)}}{1 - e^{-\eta}} \ .$$

6.5.2 Perceptron and Winnow

We now consider two online algorithms that learn a linear threshold function. *Perceptron* is a classic algorithm that dates back to the 1960s. The *Winnow* algorithm was proposed by Littlestone (1987) as a better alternative when there are many irrelevant features in the inputs.

Both algorithms follow the general schema given in Algorithm 3. Perceptron uses the *additive* update rule:

$$\text{LTF-}\textsc{Update}(w_{t-1}, \eta, x_t, y_t) = w_{t-1} - \eta y_t x_t$$

whereas Winnow uses a *multiplicative* update rule

$$\text{LTF-}\textsc{Update}(w_{t-1}, \eta, x_t, y_t) = \frac{w_{t-1} \odot \exp(-\eta y_t x_t)}{Z_t}$$

where $\odot$ and exp denote entry wise multiplication and exponentiation respectively. The normalization constant $Z_t = \sum_{j=1}^{d} w_{t-1,j} \exp(-\eta y_t x_{t,j})$ ensures that the weight vector remains a probability distribution. Also note that in Perceptron, we use the initialization $w_0 = \mathbf{0}$ whereas Winnow initializes w_0 with the uniform distribution over the d features.

Algorithm 3 Online Learning of Linear Threshold Functions

```
Initialize w_0
for t = 1, 2, ... do
  Receive x_t
  Predict ŷ_t = sign(⟨w, x_t⟩)
  Receive true label y_t
  if ŷ_t ≠ y_t then
    w_t ← LTF-UPDATE(w_t, η, x_t, y_t)
  else
    w_t ← w_{t-1}
  end if
end for
```

To give a mistake bound for Perceptron and Winnow, we will assume that we are in realizable case. That is, there is some weight vector $w^\star$ that can perfectly classify all the examples that the learner sees. Moreover, we can assume that there is a *margin* available in the correct classifications. For a correct classification, it should simply be the case that $y_t \langle w^\star, x_t \rangle > 0$. The margin condition will ensure that there is a lower bound $\gamma > 0$ such that

$$\forall t,\ y_t \langle w^\star, x_t \rangle \geq \gamma \ . \tag{8}$$

Theorem 28. *Suppose there exists a $w^\star$ satisfying the margin condition* (8)*. Then Perceptron with learning rate $\eta = 1$ enjoys the mistake bound*

$$M \leq \frac{\|w^\star\|_2^2 \cdot B_2^2}{\gamma^2} \ ,$$

where $B_2 = \max_t \|x_t\|_2$.

Proof. Let $T \geq 1$ and note that the number $M = M_T$ of mistakes till time T is simply

$$M_T = \sum_{t=1}^{T} m_t$$

where $m_t = \mathbf{1}[\hat{y} \neq y_t] = \mathbf{1}[y_t \langle w_t, x_t \rangle \leq 0]$. Let us calculate how the inner product of w_t with the vector $w^\star$ evolves. Note that we can write

$$w_t = w_{t-1} + m_t y_t x_t \ .$$

Thus, we have,

$$\langle w^\star, w_t \rangle = \langle w^\star, w_{t-1} + m_t y_t x_t \rangle$$

$$
\begin{aligned}
&= \langle w^\star, w_{t-1} \rangle + m_t y_t \langle w^\star, x_t \rangle \\
&\geq \langle w^\star, w_{t-1} \rangle + m_t \gamma \ ,
\end{aligned}
$$

where the last step is due to the margin assumption (8). Summing the above inequality for $t = 1, \ldots, T$ yields

$$
\gamma M_T + \langle w^\star, w_0 \rangle \leq \langle w^\star, w_T \rangle \ .
$$

Since $w_0 = \mathbf{0}$ and $\langle w^\star, w_T \rangle \leq \|w^\star\|_2 \|w_T\|_2$, we get

$$
\gamma M_T \leq \|w^\star\|_2 \|w_T\|_2 \ . \tag{9}
$$

Let us now try to upper bound $\|w_T\|_2$. We have,

$$
\begin{aligned}
\|w_t\|_2^2 &= \|w_t + m_t y_t x_t\|_2^2 \\
&= \|w_t\|_2^2 + 2 m_t y_t \langle w_t, x_t \rangle + m_t^2 y_t^2 \|x_t\|_2^2 \\
&\leq \|w_t\|_2^2 + m_t^2 y_t^2 \|x_t\|_2^2 \ ,
\end{aligned}
$$

where the last line is true because $y_t \langle w_t, x_t \rangle \leq 0$ when $m_t > 0$. Also note that $y_t^2 = 1$, $m_t^2 = m_t$ and $\|x_t\|_2 \leq B_2$. Therefore,

$$
\|w_t\|_2^2 \leq \|w_t\|_2^2 + m_t B_2^2 \ .
$$

Summing this for $t = 1, \ldots, T$ gives

$$
\|w_T\|_2^2 \leq \|w_0\|^2 + M_T B_2^2 = M_T B_2^2 \ .
$$

Combining this with (9) gives

$$
\gamma M_T \leq \|w^\star\|_2 \sqrt{M_T} B_2 \ ,
$$

which gives us the final bound

$$
M_T \leq \frac{\|w^\star\|_2^2 \cdot B_2^2}{\gamma^2} \ .
$$

□

Theorem 29. *Suppose there exists a $w^\star$ with positive entries that satisfies the margin condition* (8)*. Let $B_\infty = \max_t \|x_t\|_\infty$. Then Winnow enjoys the mistake bound*

$$
M \leq \frac{\log d}{C(\eta)}
$$

whenever

$$
C(\eta) = \left(\frac{\eta \gamma}{\|w^\star\|_1} - \log \left(\frac{e^{\eta B_\infty} + e^{-\eta B_\infty}}{2} \right) \right) > 0 \ .
$$

In particular, by setting η optimally, we get,

$$
M \leq 2 \frac{\|w^\star\|_1^2 \cdot B_\infty^2}{\gamma^2} \ .
$$

Proof. Let $u^\star = w^\star / \|w^\star\|_1$. Since $w^\star > 0$ (inequality is entrywise), $u^\star$ is a probability distribution. Moreover, Winnow maintains a probability distribution w_t at all times. We will track the progress of the algorithm using the relative entropy

$$D\left(u^\star || w_t\right) = \sum_{j=1}^{d} u_j^\star \log \frac{u_j^\star}{w_{t,j}}$$

between $u^\star$ and w_t.

Suppose a mistake occurs at time t. Then we have,

$$\begin{aligned}
D\left(u^\star || w_t\right) - D\left(u^\star || w_{t-1}\right) &= \sum_{j=1}^{d} u_j^\star \log \frac{w_{t-1,j}}{w_{t,j}} \\
&= \sum_{j=1}^{d} u_j^\star \log \frac{Z_t}{\exp(\eta y_t x_{t,j})} \\
&= \log Z_t \sum_{j=1}^{d} u_j^\star - \eta y_t \sum_{j=1}^{d} u_j^\star x_{t,j} \\
&= \log Z_t - \eta y_t \left\langle u^\star, x_t \right\rangle \\
&\leq \log Z_t - \frac{\eta\gamma}{\|w^\star\|_1} \ , \qquad (10)
\end{aligned}$$

where the last step is due to the margin assumption (8). Note that $y_t x_{t_j} \in [-B_\infty, B_\infty]$ □

For any $\alpha \in [-B_\infty, B_\infty]$ and $\eta > 0$, we have the inequality

$$e^{\eta\alpha} \leq \frac{1 + \alpha/B_\infty}{2} e^{\eta B_\infty} + \frac{1 - \alpha/b_\infty}{2} e^{-\eta B_\infty} \ .$$

To see this consider a random variable Z that takes value ηL with probability $(1 + \alpha/B_\infty)/2$ and takes value $-\eta L$ with probability $(1 - \alpha/B_\infty)/2$. Then the inequality above is claims that $\exp(\mathbb{E}[Z]) \leq \mathbb{E}[\exp(Z)]$ which is indeed true by Jensen's inequality. Using the inequality above, we have

$$\begin{aligned}
Z_t &= \sum_{j=1}^{d} w_{t-1,j} e^{\eta y_t x_{t-1,j}} \\
&\leq \sum_{j=1}^{d} w_{t-1,j} \left(\frac{1 + y_t x_{t-1,j}/B_\infty}{2} e^{\eta B_\infty} + \frac{1 - y_t x_{t-1,j}/B_\infty}{2} e^{-\eta B_\infty} \right) \\
&= \left(\frac{e^{\eta B_\infty} + e^{-\eta B_\infty}}{2} \right) \sum_{j=1}^{d} w_{t-1,j} + \left(\frac{e^{\eta B_\infty} - e^{-\eta B_\infty}}{2B_\infty} \right) \sum_{j=1}^{d} y_t w_{t-1,j} x_{t,j} \\
&= \left(\frac{e^{\eta B_\infty} + e^{-\eta B_\infty}}{2} \right) + \left(\frac{e^{\eta B_\infty} - e^{-\eta B_\infty}}{2B_\infty} \right) \sum_{j=1}^{d} y_t \left\langle w_{t-1}, x_t \right\rangle
\end{aligned}$$

$$\leq \left(\frac{e^{\eta B_\infty} + e^{-\eta B_\infty}}{2} \right) ,$$

where the last step is true because, in a mistake round, $y_t \langle w_{t-1}, x_t \rangle \leq 0$. Combining this bound on Z_t with (10) gives us, on any mistake round,

$$D(u^\star || w_t) - D(u^\star || w_{t-1}) \leq -C(\eta)$$

where, we have defined

$$C(\eta) = \frac{\eta\gamma}{\|w^\star\|_1} - \log \left(\frac{e^{\eta B_\infty} + e^{-\eta B_\infty}}{2} \right) .$$

Thus, for any round,

$$D(u^\star || w_t) - D(u^\star || w_{t-1}) \leq -C(\eta) m_t$$

where $m_t = \mathbf{1}[\hat{y}_t \neq y_t]$. Summing this over $t = 1, \ldots, T$ gives

$$D(u^\star || w_T) - D(u^\star || w_0) \leq -C(\eta) \sum_{t=1}^{T} m_t = -C(\eta) M_T .$$

Since relative entropy is always non-negative $D(u^\star || w_T) \geq 0$. On the other hand,

$$D(u^\star || w_0) = \sum_{j=1}^{d} u_j^\star \log(u_j^\star d) \leq \sum_{j=1}^{d} u_j^\star \log d = \log d .$$

Thus, we have

$$-\log d \leq -C(\eta) M_T$$

which gives, whenever $C(\eta) > 0$,

$$M_T \leq \frac{\log d}{C(\eta)} .$$

By choosing η to maximize $C(\eta)$, we get

$$\eta = \frac{1}{B_\infty} \log \left(\frac{L + \gamma/\|w^\star\|_1}{L - \gamma/\|w^\star\|_1} \right) .$$

which yields the bound

$$M_T \leq \frac{\log d}{g\left(\frac{\gamma}{B_\infty \|w^\star\|_1} \right)} .$$

where

$$g(x) = \frac{1+x}{2} \log(1+x) + \frac{1-x}{2} \log(1-x) .$$

The second statement of the theorem now follows because $g(x) \geq x^2/2$ for $x \in [-1, 1]$.

7 Beyond the Basic Probabilistic Framework

Our overview of learning theory emphasized the classical setting in which a learning algorithm sees fully labeled data that is an iid draw from an unknown distribution. There are many extensions possible to this basic setting. We mention a few directions that are have been explored by researchers. Many of these continue to evolve actively as new results appear.

Richer Output Spaces We mostly talked about classification and regression. One can certainly consider richer output spaces. The field of *structured prediction* (Bakir et al., 2007) deals with exponentially large output spaces consisting of graphs, trees, or other combinatorials objects. Another output space that arises in ranking applications is the space of permutations of a finite set of objects. There has also been work on learning vector valued functions (Micchelli and Pontil, 2005).

Learning with Dependent Data The iid assumption has also been relaxed. Aldous and Vazirani (1990), Gamarnik (1999) consider Markov chains. Irle (1997), Meir (2000), Vidyasagar (2003), Lozano et al. (2006), Mohri and Rostamizadeh (2009) and Steinwart et al. (2009) consider mixing processes. There is also a negative result (Nobel, 1999) showing that there is no universally consistent learning algorithm for stationary ergodic processes.

Learning with Adversarially Generated Data As in the mistake bound model, it is possible to develop a theory of learning from online adversarially generated data provided one defines learnability in terms of low regret

$$\sum_{t=1}^{T} \ell(f_{t-1}(x_t), y_t) - \inf_{f \in \mathcal{F}} \ell(f(x_t), y_t)$$

which compares the performance of the learner's functions $f_1, \ldots, f_T$ to the best function chosen in hindsight (note that the learner has to commit to f_{t-1} seeing only examples till (x_{t-1}, y_{t-1}). Analogues of Rademacher complexity, covering numbers, and fat shattering dimension have been developed (Rakhlin et al., 2010). This topic has fascinating connections to information theory and game theory. See Cesa-Bianchi and Lugosi (2006) for a comprehensive overview.

Active Learning In active learning, we consider situations where the learner has some control over the data it uses to learn. This is in contrast to the basic iid model where the learner passively sees iid data drawn from a distribution. In particularly useful model of active learning, we assume that the learning has access to *unlabeled* points $X_1, \ldots, X_n$ drawn iid from a distribution either all at one (a *pool*) or one at a time (a *stream*). The learner can only query for labels of examples in the pool or the stream. The *label complexity* of the learner is the number of label queries it needs to make in order to achieve a desired

accuracy with a given level of confidence. In a lot of situations, unlabeled data is cheap and easy to get. So, it makes sense to charge the learner only for the number of labels it requires. The label complexity of different problems in an active learning has been studied (see, for example, Hanneke (2011) and references therein). Given the practical importance of reducing the number of labels required and the relatively unexplored nature of active learning, a lot remains to be done.

Semisupervised Learning Semisupervised learning refers to learing with labeled data and large amounts of unlabeled data. A lot of work has been done in this area including methods that try to learn classifiers whose decision boundary passed through low density regions of the unlabeled data to graph based methods that use regularization to penalize functions that are nonsmooth along edges of a graph built from unlabeled data. For a good collection of articles on semisupervised learning, see the collection Chapelle et al. (2006). In the same collection, Balcan and Blum propose a PAC style framework for semisupervised learning.

Learning Multiple Tasks In multitask learning, the learner is faced with learning multiple related tasks. Because of task relatedness, we expect the learner to require fewer samples when the tasks are learned together rather than separately. Many approaches try to encode task relatedness into a regularization function that couples all the tasks together. Some important advances in our theoretical understanding of multitask learning have been made (Caruana, 1997; Baxter, 2000; Evgeniou et al., 2005; Maurer, 2006) but a comprehensive theory, if one exists, remains to be found.

8 Conclusions and Future Trends

Understanding intelligence and constructing intelligent machines is a great scientific challenge. An understanding of learning is central to an understanding of intelligence. Whether we talk about learning on an evolutionary time scale or learning within the lifespan of an organism, learning plays a crucial role in the development of intelligent behavior. What learning theory has attempted to do so far is provide a solid mathematical framework within which to pose and solve questions regarding the process of learning. Admittedly, the most mature part of learning theory is the theory of supervised learning including classification and regression problems. But this is just a beginning rather than the end. We hope that future research will make learning theory much richer and applicable to a broader variety of problems in both natural and artificial systems.

Learning theory is an interdisciplinary field. It draws ideas and inspiration from a variety of disciplines including biology, computer science, economics, philosophy, psychology, statistical physics, and statistics. It attempts to achieve clarity and rigor by using the language of mathematics to precisely state assumptions and prove results. We expect that interdisciplinary cross-fertilization of

ideas will continue to enrich and give new directions to learning theory. For example, learning theory has much to contribute to problems arising in diverse areas such as control (Vidyasagar, 2003) and dynamical systems (Campi and Kumar, 1998), learning in games (Young, 2004), privacy (Balcan et al., 2012), evolution (Valiant, 2009), and mechanism design (Balcan et al., 2008).

References

D. Aldous and V. Vazirani. A Markovian extension of Valiant's learning model. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 392–396 vol.1, 1990.

N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4): 615–631, 1997.

A. Ambroladze, E. Parrado-Hernández, and J. Shawe-Taylor. Complexity of pattern classes and the lipschitz property. *Theoretical Computer Science*, 382 (3):232–246, 2007.

D. Angluin and P. D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

M. Anthony and N. Biggs. *Computational Learning Theory: An Introduction*. Number 30 in Cambridge Tracts in Computer Science. Cambridge University Press, 1997.

G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, Ben Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data*. MIT Press, 2007.

M.-F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour. Reducing mechanism design to algorithm design via machine learning. *Journal of Computer and System Sciences*, 74(8):1245–1270, 2008.

M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *JMLR Workshop and Conference Proceedings*, pages 26.1–26.22, 2012.

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3: 463–482, 2002.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

S Ben-David, D. Pál, and S. Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

M. Campi and P. R. Kumar. Learning dynamical systems in a stationary environment. *Systems and Control Letters*, 34(3):125–132, 1998.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–295, 1995.

F. Cucker and D.-X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2007.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, 1996.

R. M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2): 606–645, 2009.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of online-learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.

Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.

D. Gamarnik. Extension of the PAC framework to finite and countable Markov chains. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 308–317, 1999.

L. Györfi, M. Kohler, A. Krzyjak, and H. Walk. *A Distribution Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer, 2002.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, second edition, 2009.

D. Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

A. Irle. On the consistency in nonparametric estimation under mixing assumptions. *Journal of Multivariate Analysis*, 60:123–147, 1997.

M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, November 1998.

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 275–282, 2002.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

A. Lozano, S. Kulkarni, and R. Schapire. Convergence and consistency of regularized boosting algorithms with stationary β-mixing observations. In *Advances in Neural Information Processing Systems 18*, pages 819–826, 2006.

P. Massart. *Concentration inequalities and model selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, 2007.

A. Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7:117–139, 2006.

David A. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–137, 1943.

R. Meir. Nonparametric time series prediction through adaptive model selection. *Machine Learning*, 39(1):5–34, 2000.

C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

M. Mohri and A. Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In *Advances in Neural Information Processing 21*, pages 1097–1104, 2009.

S. Mukherjee, P. Niyogi, T. Poggio, and R. M. Rifkin. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25 (1-3):161–193, 2006.

B. K. Natarajan. *Machine Learning: A Theoretical Approach*. Morgan Kauffmann, 1991.

A. Nobel. Limits to classification and regression estimation from ergodic processes. *The Annals of Statistics*, 27(1):262–273, 1999.

A. B. J. Novikoff. On convergence proofs for perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata (New York, 1962)*, pages 615–622, 1963.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In *Advances in Neural Information Processing Systems 23*, pages 1984–1992, 2010.

F. Rosenblatt. Two theorems of statistical separability in the perceptron. In *Proceedings of a Symposium on the Mechanization of Thought Processes*, pages 421–456, London, 1959. Her Majesty's Stationary Office.

D. E. Rumelhart, G. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

A. L. Samuels. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–233, 1959.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory Series A*, 13:145–147, July 1972.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5: 197–227, 1990.

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11: 2635–2670, 2010.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

I. Steinwart, D. Hush, and C. Scovel. Learning from dependent observations. *Journal of Multivariate Analysis*, 100(1):175–194, January 2009.

C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5 (4):595–620, 1977.

A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

L. G. Valiant. A theory of the learnable. *Journal of the ACM*, 27(11):1134–1142, 1984.

L. G. Valiant. Evolvability. *Journal of the ACM*, 56(1), 2009.

V. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, 1998.

V. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for engineering and information science. Springer, second edition, 2000.

V. Vapnik. *Estimation of Dependences Based on Empirical Data; Empirical Inference Science: Afterword of 2006*. Springer, second edition, 2006. Reprint of 1982 edition with afterword of 2006.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of event to their probabilities. *Soviet Mathematics Doklady*, 9: 915–918, 1968.

M. Vidyasagar. *Learning and Generalization: With Application to Neural Networks*. Communications and control engineering. Springer, second edition, 2003.

M. K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. In *Proceedings of the Third European Conference on Computational Learning Theory*, pages 1–2, 1997.

H. Peyton Young. *Strategic Learning and Its Limits*. Oxford University Press, 2004.

Cross References

Related articles within the Work: Model Selection, Neural Networks, Kernel Methods and SVMs, Probabilistic Graphical Models, Clustering, Semi-supervised learning, Sparse Representations and Compressed Sensing, Multi-task learning, On-line learning

Author Biography and Photograph

Ambuj Tewari is with the Department of Statistics, University of Michigan, Ann Arbor. He has served on senior program committees of the conferences Algorithmic Learning Theory (ALT), Conference on Learning Theory (COLT), and Neural Information Processing Systems (NIPS). His work has received both the

student paper award (2005) and the best paper award (2011) at COLT. He received his M.A. in Statistics (2005) and Ph.D. in Computer Science (2007) from the University of California at Berkeley where his advisor was Peter Bartlett. He was a research assistant professor in Toyota Technological Institute at Chicago (2008-2010), an assistant professor (part-time) in the Department of Computer Science, University of Chicago (2008-2010), and a post-doctoral fellow in the Institute for Computational Engineering and Sciences, University of Texas at Austin (2010-2012). He has also been a Visiting Researcher at Microsoft Research, Redmond.

Peter Bartlett is with the Division of Computer Science and Department of Statistics, University of California at Berkeley, and with the School of Mathematical Sciences, Queensland University of Technology. He is the co-author of the book 'Learning in Neural Networks: Theoretical Foundations.' He has served as associate editor of the journals Machine Learning, Mathematics of Control Signals and Systems, the Journal of Machine Learning Research, the Journal of Artificial Intelligence Research, and the IEEE Transactions on Information Theory. He was awarded the Malcolm McIntosh Prize for Physical Scientist of the Year in Australia for his work in statistical learning theory. He was a Miller Institute Visiting Research Professor in Statistics and Computer Science at U.C. Berkeley in Fall 2001, and a fellow, senior fellow and professor in the Research School of Information Sciences and Engineering at the Australian National University's Institute for Advanced Studies (1993-2003).

Relevant Websites

- Scholarpedia has a section devoted to machine learning containing links to articles written by leading researchers.
  `http://www.scholarpedia.org/article/Category:Machine_learning/`
- The Encyclopedia of Machine Learning is a valuable online resource avail-

able through subscription.
http://www.springerlink.com/content/978-0-387-30768-8/

- The Journal of Machine Learning Research a leading open access journal publishing high quality articles in machine learning and related topics.
http://jmlr.csail.mit.edu/

- Another high quality journal publishing articles on machine learning and related topics is Machine Learning.
http://www.springer.com/computer/ai/journal/10994

- There is a Google group for postings of possible interest to learning theory researchers.
http://groups.google.com/group/learning-theory/

- The website of the International Machine Learning Society. The society organizes the annual International Conference on Machine Learning (ICML).
http://www.machinelearning.org/

- The website of the Association for Computational Learning. The association organizes the annual Conference on Learning Theory (COLT).
http://seed.ucsd.edu/joomla/

- The website of the Neural Information Processing Systems (NIPS) Foundation. The foundation organizes the annual NIPS conference.
http://nips.cc/

- The arXiv is good way to find recent preprints in machine learning and learning theory.
http://arxiv.org/list/cs.LG/recent/