Agnostic Boosting

Shai Ben-David[1], Philip M. Long[2], and Yishay Mansour[3]

[1] Department of Computer Science
Technion
Haifa 32000
Israel
shai@cs.technion.ac.il
[2] Genome Institute of Singapore
1 Research Link
IMA Building
National University of Singapore
Singapore 117604, Republic of Singapore
plong@comp.nus.edu.sg
[3] Department of Computer Science
Tel-Aviv University
Tel-Aviv, Israel
mansour@math.tau.ac.il

Abstract. We extend the boosting paradigm to the realistic setting of agnostic learning, that is, to a setting where the training sample is generated by an arbitrary (unknown) probability distribution over examples and labels. We define a β-weak agnostic learner with respect to a hypothesis class F as follows: given a distribution P it outputs some hypothesis $h \in F$ whose error is at most $\mathbf{er}_P(F) + \beta$, where $\mathbf{er}_P(F)$ is the minimal error of an hypothesis from F under the distribution P (note that for some distributions the bound may exceed a half).
We show a boosting algorithm that using the weak agnostic learner computes a hypothesis whose error is at most $\max\{c_1(\beta)\mathbf{er}(F)^{c_2(\beta)}, \epsilon\}$, in time polynomial in $1/\epsilon$. While this generalization guarantee is significantly weaker than the one resulting from the known PAC boosting algorithms, one should note that the assumption required for β-weak agnostic learner is much weaker. In fact, an important virtue of the notion of weak agnostic learning is that in many cases such learning is achieved by efficient algorithms.

1 Introduction

Boosting has proven itself as a powerful tool both from a theoretical and a practical perspective of Machine Learning [11]. From a theoretical perspective, it gives a very clean and elegant model in which one can develop new algorithmic ideas and even hope to analyze some existing heuristics. From a practical perspective, although the weak learning assumption can rarely be proven, the boosting algorithms have had a dramatic impact on practitioners. In a sense, this work can be viewed as a step towards providing a theoretical explanation to this phenomenon. We prove that under certain conceivable conditions, boosting has some nontrivial performance guarantees even when no weak learners exist.

The Probably Approximately Correct (PAC) model [12] developed two separate models. The first one assumes that the target function belongs to the class studied, which was the original PAC assumption, and tries to drive the error as close to zero as possible. The second allows an arbitrary target function, but rather than shooting for absolute success, compares the error of the learner's hypothesis to that of the best predictor in some pre-specified comparison class of predictors. This model is also known as agnostic learning [8]. When one tries to consider which model is more realistic, it has to be the case that the agnostic model wins. We rarely know if there is a clear target function, let alone if it belongs to some simple class of hypotheses.

The aim of this paper is to study the boosting question in an agnostic setting. The first step has to be to define an analogue of the weak learning assumption. In the original formulation, a fixed but unknown target function generated labels, and a weak learner was assumed to achieve error less than $1/2 - \gamma$ for any distribution over the instances.

We define a β-weak agnostic learner with respect to a hypothesis class F as follows: given any distribution P over instance-label pairs it outputs some hypothesis $h \in F$ whose error is at most $\mathbf{er}_P(F) + \beta$. Since error of $1/2$ can be trivially achieved (let us assume that every concept class we consider contains both the constant 1 and the constant 0 functions), this implies that in order for the answer of the weak learner to convey interesting information it has to be the case that $\mathbf{er}_P(F)$ is less than $1/2 - \beta$.

Note that the β-weak agnostic learner assumption is only an assumption about the learner and not about the hypothesis class F, that is, such learners exist for every hypothesis class, even if we take $\beta = 0$. The interesting aspect of such weak learners is their complexity (both sample complexity and computational complexity).

The search for a 'strong' hypothesis in the agnostic setting is NP hard. More precisely, there are no known learning algorithm that, for a non-trivial hypothesis class F, finds in time polynomial in $1/\epsilon$ a hypothesis in F that has error below $\mathbf{er}_P(F) + \epsilon$. Furthermore, for many interesting classes F, for small enough β (in the order of 0.005), unless P=NP there exist no β-weak agnostic learner ([2], [3]). However, no currently known result rules out the existence of agnostic β-weak learners for these classes once β is sufficiently large (say, $\beta > 0.1$). Furthermore, the hardness results cited above rule out only the existence of efficient finders of good hypothesis *within* a class F. Since the output of a boosting algorithm is a member of a larger class of functions - the convex hull of F, all currently known hardness results are consistent with the existence of efficient algorithms that solve the agnostic learning task for non-trivial classes via boosting.

The question is what can one hope to achieve under the β-weak agnostic learner assumption. It seems unreasonable to expect that agnostic weak learners can be always transformed into strong learners, as weak learners exist also for trivial classes, say F that includes only a single hypothesis. On the other hand, clearly we can find a hypothesis whose error is $\mathbf{er}_P(F)+\beta$, but can we do better?

In this paper we answer this question in the affirmative. We show that given a parameter ϵ, there is an efficient algorithm that, using a β-weak agnostic learner as an oracle, can construct a predictor whose error is at most

$$c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon,$$

where $c_2(\beta) = 2(1/2-\beta)^2/\ln(1/\beta - 1)$, and $c_1(\beta)$ is a constant which depends only on β. Note that for small values of $\mathbf{er}_P(F)$ we outperform the naive bound.

Our algorithm simply runs AdaBoost [6] for a certain number of pre-specified steps. The interesting part is to show that one can exhibit a significant gain in the accuracy this way. By no means do we think that this is the ultimate answer. A major open problem of this research is whether one can achieve a bound of $O(\mathbf{er}_P(F)+\epsilon)$ using a β-weak agnostic learner.

To motivate our results and model we consider agnostic learning of a general hypothesis class of VC dimension d. Assume that the error-minimization task - finding a member of the class that minimizes the error over a sample of size m - is performed in $T(m)$ time. The naive way to produce a hypothesis h such that $\mathbf{er}_P(h) \leq \mathbf{er}_P(F)+\epsilon$, is to sample $m = \tilde{O}(d/\epsilon^2)$ examples, and try to find the best hypothesis from our class. The running time of such a learner is therefore of order $T(\frac{d}{\epsilon^2})$. Our approach would be to create β-weak agnostic learner, for some fixed β. For that purpose we need to sample only $m' = \tilde{O}(d)$, and the running time of the minimization is $T(\tilde{O}(d))$, independent of ϵ. Applying our boosting result we get a learner that runs in time $T(\tilde{O}(d)) \times poly(1/\epsilon)$ for some polynomial $poly()$, and finds a hypothesis such that $\mathbf{er}_P(h) \leq c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)}+\epsilon$. The main benefit is that the running time depends fairly weakly on ϵ. This benefit becomes apparent as $\mathbf{er}_P(F)$ approaches zero. (A similar result, for the case of $\mathbf{er}_P(F) = 0$, is implicit in [10, 5]. While they consider the sample complexity, their ideas can be used also for the computational complexity.)

2 Preliminaries

Fix a set X. An *example* is an element of $X \times \{-1,1\}$ and a *sample* is a finite sequence of examples. A *hypothesis* is a function from X to $\{-1,1\}$. For a hypothesis h, and a probability distribution P over $X \times \{-1,1\}$, define the *error* of h with respect to P, to be

$$\mathbf{er}_P(h) = \mathbf{E}_{(x,y)\sim P}(h(x) \neq y)).$$

Similarly, for a sample S, let $\mathbf{er}_S(h)$ be the fraction of examples (x,y) in S for which $h(x) \neq y$. For a set F of functions from X to $\{-1,1\}$, define $\mathbf{er}_P(F) = \inf_{h\in F} \mathbf{er}_P(h)$.

A *learning strategy* is a mapping from samples to hypotheses; If it is computable, then it is a *learning algorithm.*

For some domain X and a comparison class F of functions from X to $\{-1, 1\}$, a *β-weak agnostic learning oracle*, given as input an oracle for sampling according to some probability distribution P over $X \times \{-1, 1\}$, returns a hypothesis h such that

$$\mathbf{er}_P(h) \leq \mathbf{er}_P(F) + \beta.$$

The following definition places a mild requirement that is satisfied by most common concept classes.

Definition 1. *Let F be a class of functions from some domain X to $\{-1, 1\}$.*

- *The F-*consistency problem *is defined as follows:*
  Input: *A finite labeled sample S.*
  Output: *A hypothesis $h \in F$ such that $\mathbf{er}_S(h) = 0$, if such h exists, and* NO *otherwise.*
- *We say that a class F is* con-decidable *if the F - consistency problem is decidable.*

Notation: Let $H_2(p)$ be the binary entropy function, i.e. $H_2(p) = -p \log_2(p) - (1-p)\log(1-p)$. It is well known that for $k \leq n/2$, $\sum_{i=0}^{k} \binom{n}{i} \leq 2^{H_2(k/n)n}$.

3 Existence of Efficient Agnostic Weak Learners

It should be clear from the definition that, for every hypothesis class F of finite VC-dimension, for every fixed β, a β-weak agnostic learning strategy always exists – simply chose h to minimize $\mathbf{er}_S(h)$ for a sufficiently large sample S drawn independently at random according to P. The interesting question that arises in this context is the computational complexity of weak learning algorithms.

Theorem 1. *Let F be a con-decidable class with VC-dim$(F) = d < \infty$. Then, for every $\beta > 0$ there exist a β - weak learner for F that succeeds with probability $\geq 1 - \delta$ and runs in time*

$$O\left(t_F(s(\beta, d)) \times \exp\left(H_2(\mathbf{er}_P(F) + \beta/2)s(\beta, d)\right)\ln(\frac{1}{\delta})\right),$$

where $t_F : \mathbf{N} \mapsto \mathbf{N}$ is the running time of an algorithm for the consistency problem for the class F, and $s(\beta, d) = c\frac{d}{\beta^2}$, for some constant c.

Proof. Let d denote the VC-dimension of the class F. Having access to an oracle sampling according to a distribution P, the weak learner starts by asking for a sample S of size $s(\beta, d)$. By the standard VC-dimension uniform convergence bounds, such sample size guarantees that with probability exceeding 1/2, for every $h \in F$, $|\mathbf{er}_S(h) - \mathbf{er}_P(h)| \leq \beta/2$.

Next, the weak learner performs an exhaustive search for Argmin$\{\mathbf{er}_S(h) : h \in F\}$. One way of carrying out such a search is as follows:

Given a sample S of size m, the algorithm considers all subsets $T \subset S$ in order of their size, breaking ties arbitrarily. Once it finds a hypothesis $h \in F$ that classifies all the examples in $S-T$ correctly, it returns this h. It follows that the running time of the algorithm can be bounded by $t_F(|S|) \times \exp(H_2(\mathbf{er}_P(F) + \beta/2)|S|)$. Finally, using a standard 'test and re-sample' trick, for a multiplicative factor of order $\ln(1/\delta)$, the confidence parameter can be boosted from $1/2$ to $1-\delta$ (see e.g., [10]). □

Corollary 1. *If F is s con-decidable class having a constant VC-dimension, then, for every $\beta > 0$ there exist a β - weak agnostic learner for F that runs in time $O(\ln(1/\delta))$ and succeeds with probability $\geq 1-\delta$.*

4 Agnostic Boosting

In this section we prove our main theorem about boosting using β-weak agnostic learner. Theorem 1 above essentially states that for classes of finite VC dimension there are learning algorithms that run in time exponential in a parameter β (and some other parameters) and outputs a hypothesis whose expected error is within an additive factor β from the best hypothesis in the class. The boosting results of this section show that, as long as the additive approximation factor β (or, as it is commonly denoted, ϵ) is above some threshold, there are learning algorithms whose running time is only polynomial in ϵ. The threshold for which we can prove these results is a function only of $\mathbf{er}_P(F)$ and goes to zero as $\mathbf{er}_P(F)$ does.

The algorithm that we analyze is a slight variant of AdaBoost [6]. It uses the oracle for sampling according to P to generate oracles for sampling under a sequence $D_1, D_2, ...$ of filtered distributions, and passes these to the weak learner, which in response returns a sequence $h_1, h_2, ...$ of weak hypotheses.

The main intuition is as follows. The generalization guarantee that the algorithm has to achieve is sufficiently weak as to allow a trivial hypothesis for input probability distributions that drive the error rate of a weak learner close to $1/2$. Consequently, we only have to address input distributions relative to which a β-weak agnostic learner is guaranteed to have small error. In order to carry out the usual analysis of boosting, we have to make sure that this assumption remains valid for the new distributions that are generated by the boosting algorithm. We therefore work out an upper bound on the rate at which the boosting distributions may change. We can keep iterating the boosting steps as long as we do not generate distributions that are too far from the input sample distribution. The final step of our analysis is a calculation of the amount of progress that boosting can achieve under this constraint.

Theorem 2. *Fix a domain X, and a class F of functions from X to $\{-1,1\}$. There is an algorithm A such that, for any probability distribution P over $X \times \{-1,1\}$, if A is given access to a β-weak agnostic learning oracle for F, and a source of random examples of P, then for any $\epsilon > 0$, in polynomial in $1/\epsilon$ time, with probability at least $1/2$, algorithm A returns a hypothesis h such that*

$$\mathbf{er}_P(h) \leq c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon,$$

where $c_2(\beta) = 2(1/2 - \beta)^2/\ln(1/\beta - 1)$*, and* $c_1(\beta)$ *is a constant which depends only on* β*.*

Proof. We will begin by assuming that the algorithm is also given $\mathbf{er}_P(F)$ as input. We will discuss how to remove this assumption at the end of the proof (in short, standard guessing and hypothesis testing techniques suffice).

We now spell out algorithm A, which is simply AdaBoost [6], with some of the parameters fixed. Let

$$T = \min\left\{ \left\lceil \frac{\ln((1/2-\beta)/\mathbf{er}_P(F))}{\ln(1/\beta - 1)} \right\rceil, \left\lceil \frac{1}{(1/2-\beta)^2} \ln \frac{1}{\epsilon} \right\rceil \right\}.$$

and

$$\eta_i = (1/2 - \beta) - (1/\beta - 1)^i \mathbf{er}_P(F).$$

This implies that $\eta_i \geq 0$ for $i \leq T$. Also

$$\alpha_i = \frac{1}{2} \ln \frac{1/2 + \eta_i}{1/2 - \eta_i}$$

which implies that

$$e^{2\alpha_i} = \frac{1/2 + \eta_i}{1/2 - \eta_i} \leq \frac{1}{\beta} - 1.$$

Algorithm A starts by setting D_0 to be P, then for $t = 0, ..., T$, it

- passes D_t to the weak learning algorithm,
- gets h_t in return
- generates D_{t+1} in two steps by first, for each $(x, y) \in X \times \{-1, 1\}$, setting

$$D'_{t+1}(x,y) = \begin{cases} e^{\alpha_t} D_t(x,y) & \text{if } h_t(x) \neq y \\ e^{-\alpha_t} D_t(x,y) & \text{otherwise} \end{cases}$$

  then normalizing by setting $Z_{t+1} = \sum_{(x,y)} D'_{t+1}(x,y)$ and $D_{t+1}(x,y) = D'_{t+1}(x,y)/Z_t$.

Finally, it outputs a function h obtained through a majority vote over $h_1, ..., h_T$. Note that

$$Z_t \geq \sum_{(x,y)} e^{-\alpha_{t-1}} D_{t-1}(x,y) \geq e^{-\alpha_{t-1}}.$$

This implies that for any (x, y), $D_{t+1}(x,y) \leq e^{2\alpha_t} D_t(x,y)$. By induction, for each $t \leq T$,

$$D_t(x,y) \leq e^{2\sum_{i=0}^{t} \alpha_i} P(x,y)$$

Since, by assumption, $\mathbf{er}_{D_t}(h_t) \leq \mathbf{er}_{D_t}(F) + \beta$, this implies that

$$\mathbf{er}_{D_t}(h_t) \leq e^{2\sum_{i=0}^{t} \alpha_i} \mathbf{er}_P(F) + \beta \leq 1/2 - \eta_t.$$

and A achieves an edge of at least η_t in round t. The performance of AdaBoost [6] guarantees that

$$\mathbf{er}_P(h) \leq e^{-2\sum_{i=0}^{T} \eta_i^2}.$$

Recall that,

$$\begin{aligned}
\sum_{i=0}^{T} \eta_i^2 &= \sum_{i=0}^{T} \left[(\frac{1}{2} - \beta) - (\frac{1}{\beta} - 1)^i \mathbf{er}_P(F) \right]^2 \\
&= (\frac{1}{2} - \beta)^2 T - 2(\frac{1}{2} - \beta)\mathbf{er}_P(F) \frac{(\frac{1}{\beta} - 1)^{T+1} - 1}{\frac{1}{\beta} - 2} \\
&\quad + \mathbf{er}_P^2(F) \frac{(\frac{1}{\beta} - 1)^{2(T+1)} - 1}{(\frac{1}{\beta} - 1)^2 - 1} \\
&= (\frac{1}{2} - \beta)^2 T + c(\beta)
\end{aligned}$$

where the last identity uses the fact that $(1/\beta - 1)^T \leq (1/2 - \beta)/\mathbf{er}_P(F)$. In the case that $T = \left\lceil \frac{1}{(1/2-\beta)^2} \ln \frac{1}{\epsilon} \right\rceil$, then $\mathbf{er}_P(h) \leq \epsilon$, completing the proof. In the case that $T = \left\lceil \frac{\ln((1/2-\beta)/er_P(F)}{\ln(1/\beta - 1)} \right\rceil$, then

$$\mathbf{er}_P(h) \leq \exp\left(-2(\frac{1}{2} - \beta)^2 \frac{\ln((\frac{1}{2} - \beta)/er_P(F)}{\ln(1/\beta - 1)} \right) = c_1(\beta) \left(\mathbf{er}_P(F)\right)^{2\frac{(\frac{1}{2}-\beta)^2}{\ln(\frac{1}{\beta}-1)}}$$

for some constant $c_1(\beta)$ which depends only on β.

It is easy to see how to simulate the distributions $D_1, ..., D_T$, given access to a source of examples for P, in polynomial time, using the rejection method [9], since always $e^{-\sum_{i=1}^{T} \alpha_i} \geq (1/\beta - 1)^{-T/2}$. Therefore, since T is bounded by a logarithm in $1/\epsilon$, the time for Algorithm A is polynomially bounded.

Recall that we assumed that Algorithm A "knew" $\mathbf{er}_P(F)$. One can construct an algorithm that does not need to know $\mathbf{er}_P(F)$ from A as follows. Note that Algorithm A can use an upper bound b on $\mathbf{er}_P(F)$, and achieve $\mathbf{er}_P(h) \leq c_1(\beta) b^{c_2(\beta)} + \gamma$ in poly$(1/\gamma)$ time. Define $\phi : [0, 1] \rightarrow [0, c_1(\beta)]$ by $\phi(x) = c_1(\beta) x^{c_2(\beta)}$. Consider the Algorithm B that uses as guesses for b all values of $\phi^{-1}(z)$ for multiples z of $\epsilon/4$, sets $\gamma = \epsilon/4$, calls Algorithm A for each of these values, then uses hypothesis testing as in [7] to estimate which of those roughly $4/\epsilon$ hypotheses is the best. One of the poly$(1/\epsilon)$ runs would produce a hypothesis with error at most $c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon/2$, and hypothesis testing can be applied to find from among a set of hypothesis with one such good one a hypothesis with error at most $c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon$. □

We can now apply the bound of Theorem 1 on the time complexity of weak learners to the above boosting result to obtain:

Corollary 2. *Let F be a con-decidable class so that VC-dim$(F) = d < \infty$. For every $\beta > 0$, there is an algorithm A such that, for any probability distribution*

P over $X \times \{-1,1\}$, if A is given access to a source of random examples of P, A runs in time

$$O(t_F(s(\beta, d, \frac{1}{\ln(\ln(1/\epsilon))})) \times \exp(H_2(\mathbf{er}_P(F) + \beta/2)s(\beta, d, \frac{1}{\ln(\ln(1/\epsilon))}) \times \ln(1/\epsilon))$$

(where $t_F : \mathbf{N} \mapsto \mathbf{N}$ is the running time of an algorithm for the consistency problem for the class F, and $s(\beta, d, \delta) = \frac{c}{\beta^2}\left(d + \ln(\frac{1}{\delta})\right)$, for some constant c).

Also, with probability at least $1/2$, algorithm A returns a hypothesis h such that

$$\mathbf{er}_P(h) \leq c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon,$$

where $c_2(\beta) = 2(1/2 - \beta)^2/\ln(1/\beta - 1)$, and $c_1(\beta)$ is a constant which depends only on β.

Proof Sketch. We apply the boosting algorithm to the agnostic weak learning algorithm of Theorem 1. However, one has to make sure that the success probability of the weak learner is high enough to endure the T many iterations required by the boosting algorithm. For that purpose, we have to chose the δ of the weak learner to be of order $(\ln(\ln(1/\epsilon)))^{-1}$. □

Corollary 3. *Let F be a con-decidable class of functions from some domain X to $\{-1,1\}$. If the VC dimension of F is finite then, for every $\beta > 0$, there is an algorithm A such that, for any probability distribution P over $X \times \{-1,1\}$, if A is given access to a source of random examples of P, then for any $\epsilon > 0$, in polynomial in $1/\epsilon$ time, with probability at least $1/2$, algorithm A returns a hypothesis h such that*

$$\mathbf{er}_P(h) \leq c_1(\beta)\mathbf{er}_P(F)^{c_2(\beta)} + \epsilon,$$

where $c_2(\beta) = 2(1/2 - \beta)^2/\ln(1/\beta - 1)$, and $c_1(\beta)$ is a constant which depends only on β.

5 Learning with large-margin half-spaces

As a first application of the above results we briefly present a learning algorithm for learning with margin half-spaces. In this learning problem the instance space is the n-dimensional Euclidean unit ball and the learner is assessed by comparison with the best half-space, but where examples falling within a given distance γ of a separating hyper-plane in the comparison class are counted as wrong.

The motivation for such learning is that, as agnostically learning with half-spaces is computationally infeasible (see [3]), a hypothesis half-space that is computed an efficient learner is bound to make more mistakes that the best possible hyper-plane. However, it may be argued that making mistakes near the boundary of a separating hyper-plane is less costly than erring on points that are classified with large margins. The margin half-space learning model can be viewed as a model that adopts this view by ignoring mistakes on points that are within γ margins of a comparison half-space.

Previous work [4] provided an algorithm for this problem whose hypothesis h satisfies $\mathbf{er}_P(h) \leq \mathbf{er}_P(H_{\gamma,n}) + \epsilon$ in $(1/\epsilon)^{O(1/\gamma^2)}$ time.

Using the basic margin generalization bound (see [1]) it is not difficult to prove the following weak learner result.

Let B^n be the unit ball in $\mathbf{R}^n$ and, for a probability distribution P over $\mathbf{B}^n \times \{-1,1\}$, let $\mathbf{er}_P(H_{\gamma,n})$ denote the minimal P- expected error of any half-space in $\mathbf{R}^n$, when points that have margin less than γ to the half-space are counted as errors.

Theorem 3. *Choose $\gamma, \epsilon > 0$. There is a polynomial-time learning algorithm A and a polynomial p such that, for any natural number n, any $\delta > 0$, and any probability distribution P over $\mathbf{B}^n \times \{-1,1\}$ (where B^n is the unit ball in $\mathbf{R}^n$), if $p(n, 1/\delta)$ examples are drawn according to P and passed to algorithm A, then with probability at least $1-\delta$, the output h of algorithm A satisfies*

$$\mathbf{er}_P(h) \leq \mathbf{er}(H_{\gamma,n}) + \epsilon.$$

We can now apply our boosting technique, namely Theorem 2, to obtain:

Theorem 4. *Choose $\gamma > 0$. There is a learning algorithm that runs in time poly($c_1^{O(1/\gamma^2)}$,$1/\epsilon$) time, while achieving $\mathbf{er}_P(h) \leq c_2 \mathbf{er}_P(H_{\gamma,n})^{c_3} + \epsilon$, where c_1, c_2 and c_3 are constants.*

Acknowledgments

We thank Peter Bartlett and Nadav Eiron for valuable conversations.

Phil Long acknowledges the support of National University of Singapore Academic Research Fund Grant RP252–000–070–107.

Yishay Mansour acknowledges the support of the Israeli Science Foundation.

References

1. P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.
2. P. L. Bartlett and S. Ben-David. Hardness Results for Neural Network Approximation Problems. *Proceedings of the 4th European Conference on Computational Learning Theory (EuroCOLT'99)*, pp 50-62. A revised version will appear in JCSS.
3. S. Ben-David, N. Eiron, and P. Long. On the difficulty of approximately maximizing agreement. *Proceedings of the 13th Annual Conference on Computational Learning Theory, COLT00*, pages 266–274, 2000.
4. S. Ben-David and H. U. Simon. Efficient learning of linear perceptrons. In *NIPS*, 2000.
5. Yoav Freund. *Data Filtering and Distribution Modeling Algorithms for Machine Learning.* PhD thesis, University of California at Santa Cruz, 1993. Retrievable from: ftp.cse.ucsc.edu/pub/tr/ucsc-crl-93-37.ps.Z.

6. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT '95*, pages 23–37. Springer-Verlag, 1995.
7. D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95:129–161, 1991.
8. M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.
9. D. E. Knuth. *The Art of Computer Programming, Volume II: Semi numerical Algorithms.* Addison-Wesley, 1981.
10. Robert E. Schapire. *The Design and Analysis of Efficient Learning Algorithms.* PhD thesis, M. I. T., 1991.
11. Robert E. Schapire. Theoretical views of boosting and applications. In *Tenth International Conference on Algorithmic Learning Theory*, 1999.
12. L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.