Introduction to Boosting : Origin, Practice and Recent Developments

Abe, Naoki
IBM T. J. Watson Research Center

Lozano, Aurélie C.
IBM T. J. Watson Research Center

出版情報：COE Lecture Note. 12, pp.56-64, 2008-09-16. Faculty of Mathematics, Kyushu University
バージョン：
権利関係：

Introduction to Boosting: Origin, Practice and Recent Developments

Naoki Abe
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
nabe@us.ibm.com

Aurélie C. Lozano
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
aclozano@us.ibm.com

Abstract

In this review, we will introduce the audience to the notion of boosting, which has become one of the most successful techniques in machine learning and statistical modeling today. We will review its historical origin in computational learning theory, as well as more recent developments that relate it to other notions in statistics (e.g. gradient boosting), and discuss some recent theoretical developments. We will also touch upon its wide usage and successes in practical applications.

1 Introduction: Origin of Boosting

We will begin with some historical perspective on boosting. Boosting has its origin in the branch of computer science known as "computational learning theory." Computational learning theory is an attempt to formalize learning problems, develop and analyze performance of learning algorithms, and characterize the computational complexity of learning problems. The phenomenon of learning is complex and the subject matter is related to many other disciplines, making the field of computational learning theory a rich and diverse inter-disciplinary field, rather than a mere branch of computer science. There are many related fields of study, most of which predate computational learning theory, such as statistics, inductive inference and information theory.

Even with so many related fields, there is still a defining characteristic of this field, which distinguishes it from the others, and that is the emphasis it places on the computational aspects of learning. The field takes the issue of computational complexity more seriously than perhaps any of its parent fields. This is a point worth emphasizing, since in a sense, the invention of "boosting" has much to do with this aspect.

Although some very interesting bodies of work had existed – most notably on learning formal languages using queries by Angluin [1], and the inductive inference in the limit paradigm of Gold [18] – it was not until Valiant's proposal of the so-called "PAC Learning Model" [31], that this area of study picked up significant interest and popularity. It is probably fair to say that Valiant's PAC learning model is the first model that made a significant progress towards integrating statistical and computational views on learning into a single mathematical framework. Since PAC learning model is so pivotal to the development of the boosting theory, we will review its definition below.

In the subsequent description, we will focus on the problem of binary classification as is done in the original PAC learning model. That is, the target function to be learned is a function f from some domain X into $\{-1, +1\}$. The learner's goal is to find a hypothesis h that approximates this target function f, given access to some feasibly small set of training data $S = \{(x_i, f(x_i))|i = 1, ..., m\}$, where each datum x_i is assumed to be drawn i.i.d. from a fixed but unknown distribution $\mathcal{D}$ over the domain X.

Definition 1 *A learning algorithm $\mathcal{A}$ is a (*strong*) PAC learning algorithm for a class of functions $\mathcal{F}$, if for any target function $f \in \mathcal{F}$, for any distribution $\mathcal{D}$, for any $\epsilon > 0$, for any $\delta > 0$, given an*

input training data set S for f of size exceeding $p(\frac{1}{\epsilon}, \frac{1}{\delta}, s)$ for some polynomial p, with probability at least $1-\delta$, $\mathcal{A}$ outputs a hypothesis h with classification error at most ϵ, namely

$$Pr\{E_{\mathcal{D}}[h \neq f] \leq \epsilon\} \geq 1 - \delta$$

with computation time polyomial in $\frac{1}{\epsilon}, \frac{1}{\delta}$ and s, where s is a measure of complexity of the target function f.

Given the above definition, one can then identify a learning problem with a class of target functions, and define the computational complexity of a learning problem using bounds on the computation time and sample size for all of the target functions in the class, using some fixed learning algorithm. In particular, a class of target functions is said to be *PAC-learnable* if and only if its complexity is bounded above by a polyonomial.

The most notable aspect of this model is the way it unifies the statistical complexity (sample size required for desired predictive accuracy) with the computational complexity aspect (computation time required to achieve the same), as a function of the common parameters, ϵ, δ, and s. Extensive theoretical study has been carried out using this model as the formal framework, and many interesting results were obtained [2]. The early results obtained, however, were mostly negative – it was established using some cryptographic assumptions that many subclasses of boolean functions are not PAC learnable, and the learnability of the most representative class, DNF, is still an open problem. As an attempt to obtain positive results, Kearns and Valiant [21] introduced the notion of "weak learnability", one that relaxes the "probably approximately correct" aspect of the PAC learning model to a definition that insists only a "slightly better than random" classification. More precisely, a weak learning algorithm is defined as follows.

Definition 2 *A learning algorithm $\mathcal{A}$ is a* weak *PAC-learning algorithm for a class of functions $\mathcal{F}$, if for any target function $f \in \mathcal{F}$, for any distribution $\mathcal{D}$, and for any $\delta > 0$, given a training set S for f of size exceeding $p_1(s, \frac{1}{\delta})$ for some polynomial p_1, with probability at least $1-\delta$, $\mathcal{A}$ outputs a hypothesis h with classification error at most $\frac{1}{2} - 1/p_2(s)$ for another polyonomial p_2, namely,*

$$Pr\{E_{\mathcal{D}}[h \neq f] \leq \frac{1}{2} - \frac{1}{p_2(s)}\} \geq 1 - \delta$$

with computation time polyomial in $\frac{1}{\delta}$ and s, where s is a complexity measure of the target function f.

The idea was to investigate into a general algorithmic mechanism that can be used to transform a weak learing algorithm into a (strong) PAC learning algorithm. (The existence of weak learning algorithms could be established easier for some target function classes.) Schapire provided the first "boosting algorithm", which indeed was able to convert any weak learning algorithm into a strong one. The technique he used in his original boosting procedure was quite unlike any other learning algorithm: it was to change the distribution over the examples in each iteration, feed the resulting sample into the weak learner, and then combine the resulting hypotheses into a voting ensemble, which, in the end, would have a boosted classification accuracy, as compared to the component learning procedure. His technique established the following theorem.

Theorem 1 ([29]) *A class of target functions is (strongly) PAC-learnable if and only if it is weakly learnable.*

The original boosting procedure of Schapire was a means to proving this theorem, and did not provide a method for practical use for a number of reasons. One of them was that the algorithm consisted of a process that boosts the accuracy by combining 3 calls to the weak learning algorithm, and hence the process had to be iterated to result in a complex hierarchical hypothesis. Another weakness was that it had to be given the value of γ of the component weak learning algorithm apriori, for it to work properly.

2 Practice and Success

When Freund and Schapire came up with AdaBoost, the first practical boosting algorithm, they had resolved both of these weaknesses. AdaBoost is a simple and elegant procedure that simply keeps

AdaBoost

Given $(x_1, y_1), \ldots, (x_m, y_m)$, where $x_i \in X$, $y_i \in \{-1, +1\}$

1. Initialize the weights $d^{(1)}(i) = 1/m$
2. For $t = 1, \ldots, T$:
   (a) Fit a base classifier $h_t(x)$ to the training data using weights $d^{(t)}(i)$
   (b) Compute
   $$\epsilon_t = \sum_{i=1}^{m} d^{(t)}(i) I(y_i \neq h_t(x_i))$$
   (c) Compute $\alpha_t = \frac{1}{2} \ln ((1 - \epsilon_t)/\epsilon_t)$
   (d) Update
   $$d^{(t+1)}(i) = \frac{d^{(t)}(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$
   where Z_t is a normalization factor.
3. Output the final classifier: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Figure 1: The AdaBoost algorithm

a weighting function on the input training data, which are updated in each iteration using a simple update formula and then the final hypothesis was obtained using a weighted average of the output hypotheses. The complete AdaBoost algorithm is given in Figure 1.

In their original paper in which AdaBoost was introduced, they provided a performance guarantee, which among other things, established that AdaBoost indeed was a "boosting algorithm", which converts an arbitrary weak learner into a strong PAC learner. Here we quote their theorem (Theorem 5, [14]).

Theorem 2 ([14]) *Suppose that the weak learning algorithm, when called by AdaBoost, generates hypotheses with errors $\epsilon_1, ..\epsilon_T$. Then the error ϵ of the final ensemble hypothesis h_t output by AdaBoost is bounded above by*

$$\epsilon \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t (1 - \epsilon_t)}$$

This theorem not only established the boosting property of AdaBoost, but it also shows that the algorithm is adaptive, in the sense that any advantage that the weak learner happens to achieve over random guessing can be leveraged, without the prior knowledge of the edge as was needed by earlier boosting methods. This was one of the key factors, in addition to the simplicity of the algorithm, that made AdaBoost practically useful. Subsequent studies have shown that AdaBoost, when combined with appropriate weak learners such as decision tree algorithms (e.g. C4.5) achieved the state-of-the-art accuracy in numerous bench mark classification tasks. (c.f. [13].)

While the performance guarantee given for AdaBoost in terms of boosting ability was satisfactory, it did not completely explain some of the desirable properties of AdaBoost, particularly with regard to its ability to generalize, or its predictive accuracy on unseen test data sets. The PAC learning framework provided a general theorem stating that, for any restricted class of target functions, predictive accuracy on the training data tarnslates well to that on the test data, the general theorem did not fully explain the excellent performance of AdaBoost in practice. In particular, practitioners noticed that, even after AdaBoost attained a perfect predictive accuracy on the training data set, its accuracy on the test data kept improving.

Schapire, Fruend, Bartlett and Lee [30] provided the first account for this phenomenon, in terms of the notion of "margin." The margin is a measure of "confidence" of a clsasifer's prediction, formerly defined as $-y \cdot f(x)$, where y is the true label (+1 or -1) and $f(x)$ is the predicted score, in the continuous range of $-1, +1]$. For boosting, the final ensemble hypothesis takes a weighted

voting by the weak learner's hypotheses, each predicting either +1 or -1 on each data point x, and hence the weighted average will be in the range $-1, +1]$, with values close to ± 1 indicating a higher degree of agreement among the weak learners, and hence higher confidence.

Schapire et al's theorem from [30], in one version, is repeated below.

Theorem 3 ([30]) *Suppose the class H of weak hypotheses has VC-dimension d, and that $m \geq d \geq 1$. Then, with probability at least $1 - \delta$, every weighted ensemble f over H satisfies the following bound for all $\theta > 0$*

$$P[yf(x) \leq 0] < P[yf(x) \leq \theta] + O(\frac{1}{\sqrt{m}}(\frac{d \log^2(m/d)}{\theta^2} + \log(1/\delta))^{1/2})$$

This theorem can be interpreted as saying that the predictive accuracy will be improved, as its confidence (margin) improves, upto statistical variations. They then show that AdaBoost is indeed guaranteed to increase the margin, given the same "weak learning" assumptions as before, completing an explanation for the fact that AdaBoost's generalization tends to improve, even after its training data has reached zero and stabilized.

Theorem 4 ([30]) *Suppose that the weak learning algorithm, when called by AdaBoost, generates hypotheses with errors $\epsilon_1, ..\epsilon_T$. Then for any θ we have that*

$$P[yf(x) \leq 0] < P[yf(x) \leq \theta] \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\theta}(1 - \epsilon_t)^{1+\theta}}$$

These theorems provided early account for the impressive performance that many empiricists found on AdaBoost's predictive performance in practice. Indeed, starting with Freund and Schpire's early experimental work [13], in which consistent advantage of AdaBoost over a popular ensemble method known as Bagging proposed by the prominent statistician, Leo Breiman [5], among other competing methods, were repeated in many application areas. Figure 2 shows some representative experimental results, due to [13], which show this clearly.

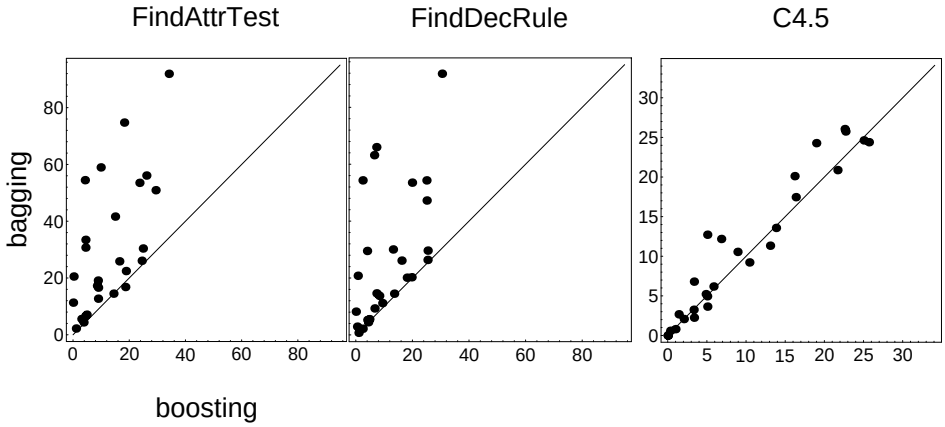

Figure 2: Comparison of Boosting versus Bagging for several weak learners [13]

Forward Stagewise Additive Modeling

1. Initialize $f_0(x) = 0$
2. For $t = 1$ to T :
   (a) Compute
   $$(\beta_t, \gamma_t) = \arg\min_{\beta,\gamma} \sum_{i=1}^{m} \phi\left(y_i, f_{m-1}(X_i) + \beta b(x_i;\gamma)\right)$$
   (b) Set $f_t(x) = f_{t-1}(x) + \beta_t b(x;\gamma_t)$.

Figure 3: Forward Stagewise Additive Modeling

3 Recent Developments

3.1 The Statistical view of Boosting

A major theoretical advance towards the understanding of boosting's success is due to Friedman et al. [17], who showed that AdaBoost can be seen as minimizing the exponential loss by stage-wise fitting of additive models.

Additive modeling forms a prediction function as a basis function expansion:

$$f(x) = \sum_{t=1}^{T} \beta_t b(x;\gamma_t),$$

where $\beta_t \in \mathbb{R}$ are expansion coefficients, and $b(x;\gamma)$ are some simple functions of the argument x depending on some parameter γ. Such models are fit to the training data by minimizing an empirical cost, i.e., the average over the training set of a loss function:

$$\min_{\{\beta_t,\gamma_t\}_1^T} \sum_{t=1}^{m} \phi\left(y_i, \sum_{t=1}^{T} \beta_t b(x_i,\gamma_t)\right). \tag{1}$$

An example of a common loss function is the squared error $\phi(y, f(x)) = (y - f(x))^2$.

As solving (1) is computationally intensive, finding an approximate solution is pursued. This is accomplished by forward stagewise additive modeling, which consists in adding one coefficient and one parameter at a time in a greedy fashion, i.e., without adjusting the coefficients and parameters previously added. The algorithm is depicted in Figure 3.

Specifically, the AdaBoost algorithm is equivalent to the forward stagewise additive modeling algorithm of Figure 3 if one sets $\phi(y, f(x)) = \exp(-yf(x))$ and lets the simple functions $b(x;\gamma_t)$ correspond to the weak hypothesis $h_t(x)$. This view leads to the derivation of additional boosting algorithms based on various loss functions (e.g. logistic loss, squared loss, etc.) and to the consideration of boosting as a greedy stagewise procedure that minimizes a convex loss function empirically. Let $z = yf$ denote the margin of a prediction f. Examples of loss functions are the Logistic Loss: $\ln(1+z)$; the Exponential Loss: $\exp(-z)$; the Least Squares: $\frac{1}{2}(z-1)^2$; the modified Least Squares: $\frac{1}{2}\max(1-z,0)^2$; the p-norm Loss: $|z-1|^p$, $p \geq 2$. These are represented in Fig. 4 along with the indicator of misclassification ($I(z \leq 0)$) of which these loss functions can be viewed as convex approximation. Notice that the loss functions listed here can all be transformed into upper bounds of the indicator of misclassification through scaling by a positive factor. Considering loss functions that are convex surrogate of the indicator of classification leads to computationally tractable procedures.

The statistical view proposed by Friedman et al has been instrumental in further developments of boosting theory and techniques, since it has led to the generalization of boosting to various loss functions, as well as its extension to the regression setting (e.g. [8]), and it has also enabled significant progress in understanding of the success of boosting methodology. In the next section, we choose to focus on a key theoretical question, of whether or not the boosting algorithm enjoys the so-called "Bayes consistency."

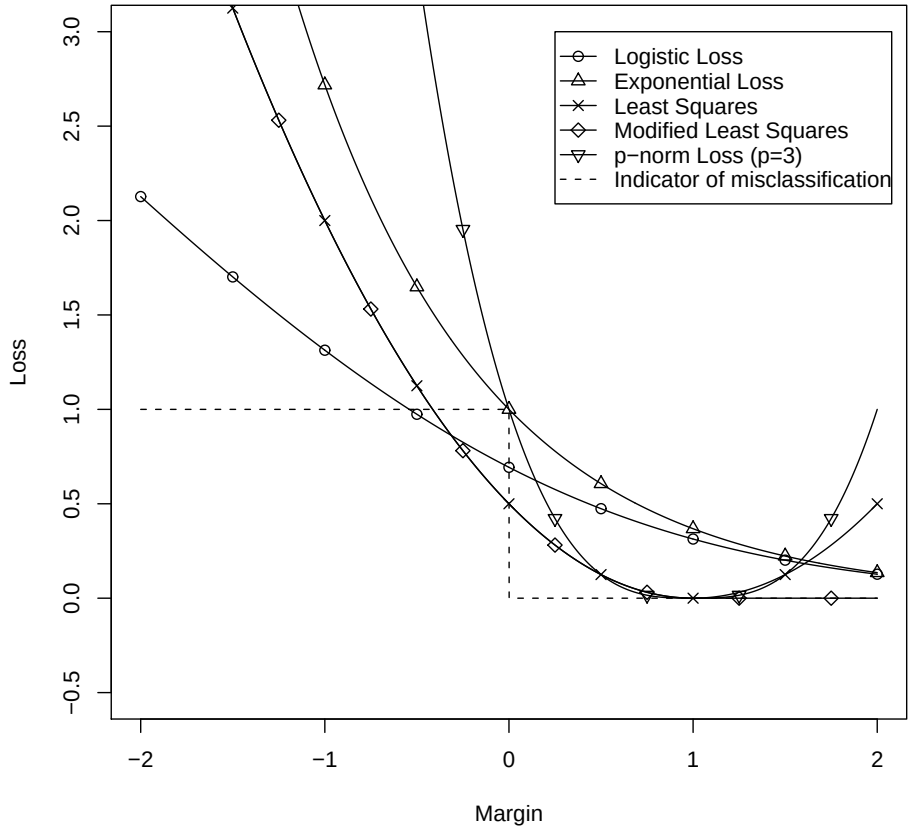

Figure 4: Example of loss functions for binary classification. For $y \in \{\pm 1\}$ and a prediction f, the margin is $z = yf$. The loss functions are logistic: $\ln(1 + f(x))$; exponential: $\exp(-z)$; least squares: $\frac{1}{2}(z-1)^2$; modified least squares: $\frac{1}{2}\max(1-z, 0)^2$; p-norm: $|z-1|^p$, $p \geq 2$,; and misclassification: $I(z \leq 0)$.

3.2 Consistency Results

We first recall the definition of Bayes consistency. Given a training set of size m, denote by h_m the classifier constructed by a certain learning method. The performance of h_m is evaluated by the classification error, which is the probability of misclassifying a new example

$$L(h_m) = \mathbb{P}\left\{h_m(x) \neq y\right\}.$$

For a given distribution $\mathcal{D}$, the classifier with smallest error is called the *Bayes classifier* and is defined by

$$h^*(x) = \left\{ \begin{array}{ll} 1 & \text{if } \eta(x) > 1/2 \\ -1 & \text{otherwise.} \end{array} \right),$$

where $\eta(X)$ is the true class conditional probability, i.e.,

$$\eta(x) = \mathbb{P}(y = 1|x).$$

The corresponding misclassification error, which is the minimum probability of error, is called the *Bayes error* and is given by

$$L^* = \mathbb{E}\min(\eta(X), 1 - \eta(X)).$$

A learning method is called Bayes consistent if as the training sample size increases the error of the resulting classifiers approaches the Bayes error.

Definition 3 ([12], Definition 6.1) *A classification rule is* consistent *for a certain distribution $\mathcal{D}$ if $\mathbb{E}L(h_n) = \mathbb{P}\{h_n(X) \neq Y\} \rightarrow L^*$ as $n \rightarrow \infty$. It is* strongly Bayes-risk consistent *if $\lim_{n\to\infty} L(h_n) = L^*$ almost surely.*

The question of whether boosting methods achieve Bayes-consistency then arises, since minimizing an empirical loss function does not necessarily imply minimizing the generalization error. When run a very long time, boosting algorithms, though resistant, are not immune to overfitting [19, 17]. There exist cases when running a boosting algorithm "forever" leads to a prediction error larger than the Bayes error in the limit of infinite sample size. Consequently, one approach for the study of consistency is to analyze how the prediction error $L(\text{Boosting}(m,t))$ for sample size m evolves with the number of rounds t in order to see whether the algorithm is process consistent, i.e. whether

$$\lim_{m\to\infty} \inf_{t\in\{1,2,3,\ldots\}} L(\text{Boosting } (m,t)) = L^*,$$

where L^* is the Bayes error. This strategy suggests stopping the boosting procedure early to achieve consistency. This approach was adopted by Bartlett & Traskin [4] who showed that if AdaBoost is stopped after $m^{1-\epsilon}$ iterations, where m is the sample size, Bayes consistency is achieved. Earlier consistency studies under the early stopping strategy include [20, 7, 34].

An alternative approach is to modify the boosting algorithms by imposing some constraints on the weights of the composite classifier to avoid overfitting. In these regularized versions of boosting, the 1-norm of the weights of the base classifiers is restricted to a fixed value. The minimization of the loss function is performed over the restricted class. This approach is adopted by [23, 24, 25]. A similar approach is to add a penalty term to the cost for complexity regularization such as [3] in the context of regression, and [33] where a kernel formulation is considered whose penalty function is the square norm of the composite classifier.

3.3 Resistance to Overfitting: The margin theory revisited

The statistical view of boosting offers an insightful perspective but has its limitations, as illustrated very recently in [26]. Specifically, such view fails to shed light on boosting's remarkable resistance to overfitting.

The "margin theory" offers a complementary view, and explains boosting's resistance to overfitting by the fact that boosting tends to maximize the margins of the training examples (and hence the confidence of the composite classifier on its decisions), and that such increase results in improving the performance on test data even if the number of iterations is large. Breiman [6] subsequently challenged this justification by proposing a boosting algorithm, arc-gv, that leads to larger margins than AdaBoost but exhibit worse performance. However Reyzin & Schapire [28] showed that the poorer performance of Breiman's algorithm is due to the increased complexity of the base classifiers, and that this is in agreement with the margin theory.

3.4 Boosting for high-dimensional data

Many recent application areas of machine learning involve data with increasing number of features. Boosting performs very well in high dimensions. An intriguing question is why? Empirical evidence was reported in [8]. Bühlman [10] subsequently stated his belief that it is mostly for high-dimensional data that boosting offers a major advantage over classical methods. Further experimental results were provided to back such statement, as well as a consistency result for L_2Boosting under very high-dimensional linear models, where the number of features can grow as $O(\exp(m))$, where m is the sample size. It was suggested in [9] that a possible explanation for the success of boosting with high-dimensional data is in the way boosting seems to perform variable selection over its iterations intelligently, when given a base learner that does variable selection itself. Whether this variable selection process of boosting is performed in an optimal way, in some well-defined sense, remains an open question.

3.5 Boosting and Bregman Divergence

Another insightful view of boosting comes from relating it to bregman distances. Kivinen and Warmuth [22] demonstrated that boosting can be seen as preforming entropy projection, and Collins

et al [11] showed that boosting and logistic regression can be unified by casting them as special cases of optimization of bregman distances. These interpretations served as a basis for the derivation of new boosting variants, most notably the U-boost algorithms proposed by Murata et al [27].

The various theories on boosting illuminate several aspects of boosting's success, but many aspects of the remarkable efficiency of boosting procedures are still mysterious. The connections between statistical, information-theoretic, optimization-theoretic and computational aspects is a distinguishing feature of boosting and we believe that neither aspect should be overlooked.

References

[1] D. Angluin, "Learning regular sets from queries and counterexamples", *Information and Computation*, Vol. 75(2), pp. 87–106, 1987.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *Journal of the ACM*, Vol. 36(4), pp. 929-965, 1989.

[3] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," in L. G. Valiant and M. K. Warmuth, editors, *Proc. of the 4th Annual Workshop on Computational Learning Theory*, pp. 243–249, 1991.

[4] P. L. Bartlett, M. Traskin, "AdaBoost is Consistent," *J. Machine Learning Research*, vol. 8, pp. 2347-2368, 2007.

[5] L. Breiman, "Bagging predictors", *Machine Learning*, pp. 123-140, 1996.

[6] L. Breiman, "Prediction games and arcing classifiers," Neural Computation, 11, 14931517, 1999.

[7] P. Bühlmann, *Consistency for L_2 boosting and matching pursuit with trees and tree-type basis functions.* Technical report, ETH Zürich, 2002.

[8] P. Bühlmann, B Yu, "Boosting with the L2 loss: regression and classification," *Journal of the American Statistical Association*, vol. 98, pp. 324-339, 2003.

[9] P. Bühlmann, "Boosting methods: why they can be useful for high-dimensional data," Proceedings of the 3rd International Workshop on Distributed Computing, 2003.

[10] P. Bühlmann, "Boosting for high-dimensional linear models," Annals of Statistics 34, 559-583, 2006.

[11] M. Collins, R.E. Schapire, and Y. Singer, "Logistic Regression, Adaboost and Bregman distances," in Proc. COLT, San Francisco, 2000.

[12] L. Devroye, L. Györfi , G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* Springer, New York, 1996.

[13] Y. Freund, R. Schapire, "Experiments with a new boosting algorithm," *Proc. 13th ICML*, pp. 148-156, 1996.

[14] Y. Freund, R. Schapire, "Decision-theoretic generalization of on-line learning and an application to boosting.," *Journal of Computer and System Sciences*, vol. 55(1), pp. 119-139, 1997.

[15] J. H. Friedman, W. Stutzle, "Projection pursuit regression," *Journal of the American Statistical Association*, vol. 76, pp. 817-823, 1981.

[16] J. H. Friedman, W. Stutzle, "Projection pursuit classification," unpublished manuscript, 1980.

[17] J. Friedman, T. Hastie , R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 38, pp. 337-374, 2000.

[18] E. M. Gold, "Language identification in the limit," classification," —em Information and Control, Vol. 10(5), pp.447-474, 1967.

[19] W. Jiang, *Does Boosting Overfit:Views From an Exact Solution.* Technical Report 00-03, Department of Statistics, Northwestern University, 2000.

[20] W. Jiang, "Process consistency for Adaboost," *Ann. Statist.*, vol. 32, pp. 13-29, 2004.

[21] M. Kearns, L. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata", *Journal of the ACM*, Vol. 41(1), pp. 67-95, 1994.

[22] J. Kivinen, M.K. Warmuth, "Boosting as entropy projection," in Proc. 12th Annu. Conference on Comput. Learning Theory, 1999.

[23] G. Lugosi, N. Vayatis, "On the Bayes-risk consistency of boosting methods," *Ann. Statist.*, vol. 32, pp. 30–55, 2004.

[24] S. Mannor, R. Meir, T. Zhang, "The Consistency of Greedy Algorithms for Classification," *LNCS* 2375 *Proc. 15th COLT*, pp. 319-333, 2002.

[25] S. Mannor, R. Meir, T. Zhang, "Greedy algorithms for classification - consistency, convergence rates, and adaptivity," *Journal of Machine Learning Research*, vol. 4, pp. 713-741, 2003.

[26] D. Mease and A. Wyner, "Evidence Contrary to the Statistical View of Boosting", JMLR 9:131156, 2008

[27] N. Murata, T. Takenouchi, T. Kanamori, and S. Eguchi, "Information geometry of U-Boost and Bregman divergence," Neural Comput. 16, 7 1437-1481, 2004

[28] L. Reyzin and R. E. Schapire. How boosting the margin can also boost classifier complexity. In Proceedings of the 23rd International Conference on Machine Learning, 2006.

[29] R. Schapire, "The strength of weak learnability" *Machine Learning*, vol. 5, pp. 197-227, 1990.

[30] R. Schapire, Y. Freund, P. L. Bartlett, W. S. Lee "Boosting the margin: "A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, J. Machine Learning Research, vol. 26(5), 1651-1686, 1998.

[31] L. Valiant, "A theory of the learnable," *Communications of the ACM*, Vol. 27(11), pp. 1134-1142, 1984.

[32] T. Zhang, "Sequential Greedy Approximation for Certain Convex Optimization Problems," *IEEE Trans. Inform. Theory*, vol. 49, pp. 682–691, 2003.

[33] T. Zhang, "Statistical Behavior and Consitency of Classification Methods based on Convex Risk Minimization," *Ann. Statist.*, vol. 32, pp. 56–85, 2004.

[34] T. Zhang, B. Yu, "Boosting with Early Stopping: Convergence and Consistency," *Ann. Statist.*, vol. 33(4), pp. 15381579, 2005.