SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE. DIST. UNLIMITED
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) SC 90297	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION The Regents of the University of California 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research, Seattle Regional Office
6c. ADDRESS (City, State, and ZIP Code) Contracts and Grants Office University of California 1156 High Street, Santa Cruz, CA 95064	7b. ADDRESS (City, State, and ZIP Code) 1107 North East 45th St., Suite 350 Seattle, WA 98105-4631
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-91-J-1162
8c. ADDRESS (City, State, and ZIP Code) Department of the Navy, 800 North Quincy St. Office of the Chief of Naval Research Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
"Analyzing the Performance of Learning Algorithms"

12. PERSONAL AUTHOR(S)
Dr. David Haussler and Dr. Manfred K. Warmuth

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
final technical	FROM 10/1/90 TO 5/31/93	08/14/93	

16 SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

19970716 143

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION U
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Scotomas

Our "lock-outs," or blind spots, we will refer to in this video curriculum as "scotomas." It is not a jargon word. It is a very useful word, because it describes everyone's blind spots. Whenever you hear anyone wailing, "I don't get it," or "It doesn't make sense to me," you can be assured that person has a scotoma. I want you to understand scotomas. They are very important as you break out of the limiting and into more valid ways of thinking. Make no mistake: friendships dissolve because of scotomas, marriages fail because of scotomas; nations, companies, schools, have scotomas. A scotoma is the block — the blind spot — that keeps us from seeing the truth, the many optional truths around us.

A scotoma is the sensory locking-out of our environment. We are imprisoned by our own "blind spots" because of our preconceived way of seeing things, our habitual way of doing things, our "lock-out" notion of what can be done. A scotoma causes us to **see what we expect to see, hear what we expect to hear and think what we expect to think.** "Oh, we've always done it that way." "He'll never be able to do it." "That company will never buy our product — it never has." Ad infinitum.

So "locking on" and "locking out" create our "blind spots," our scotomas. I have both bad news and good news about our blind spots, or scotomas.

The bad news: A great problem with scotomas is that we often don't know that we have them. We go about our daily routines, running a business, raising a family, doing our jobs, in a state of semi-myopia. We don't see the many optional truths around us.

Do I have certain "habits" at work? Conversely, do I "lock out" certain opinions, say, because a woman expresses them?

"The teacher is the pupil. The pupil is the teacher."

© Copyright 1983 - The Pacific Institute,® Inc. - Seattle, WA 98101 - All Rights Reserved

UNIVERSITY OF CALIFORNIA, SANTA CRUZ

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO

SANTA BARBARA • SANTA CRUZ

TELEPHONE (408) 459-2778
FAX (408) 459-4989

CONTRACTS AND GRANTS OFFICE
SANTA CRUZ, CALIFORNIA 95064

September 5, 1995

Chief of Naval Research
ATTN: Michael Shneier/Code 311
Mathematical, Computer, and Information Sciences Division
800 North Quincy Street, Room 607
Arlington, VA 22217-5660

re: ONR Grant No. N00014-91-J-1162
"Analyzing the Performance of Learning Algorithms"
P.I. Dr. David Haussler

Enclosed you will find three (3) copies of the final technical report from Dr. David Haussler for the above-referenced grant.

If you have any questions or require any additional information regarding this report, please do not hesitate to contact me.

Sincerely,

Elizabeth Capehart

Elizabeth Capehart
Senior Research Administrator
(408) 459-3136
capehart@cats.ucsc.edu

enclosures

cc: Defense Technical information Center (w/4 copies of report)
ONR Seattle Regional Office

DTIC QUALITY INSPECTED 5

Analyzing the Performance of Learning Algorithms

David Haussler and Manfred K. Warmuth
Department of Computer and Information Sciences
University of California at Santa Cruz, CA 95064

Abstract.

We discuss the approach to the analysis of learning algorithms that we have taken in our laboratory and summarize the results we have obtained in the last few years. We have worked on refining and generalizing the PAC learning model introduced by Valiant. Measures of performance for learning algorithms that we have examined include computational complexity, sample complexity, probability of misclassification (learning curves), and worst case total number of misclassifications or hypothesis updates. We have looked for theoretically optimal bounds on these performance measures, and for learning algorithms that achieve these bounds. Learning problems we have examined include those for decision trees, neural networks, finite automata, conjunctive concepts on structural domains, and various classes of Boolean functions. We also worked on clustering data represented as sequences over a finite alphabet. Many of the new learning algorithms that we have developed have been tested empirically as well.

Introduction

Recent years have brought a significant increase in research activity in the area of machine learning, both through increased interest among mainstream artificial intelligence researchers (see e.g. [MCM 83,86]) and through the resurgence of interest in connectionist/neural net models (see e.g. [RM86] [Hi88]). The researchers in this new field come from a variety of disciplines, including artificial intelligence, statistical pattern recognition and decision theory, neurobiology, cognitive science, and the theory of algorithms and computational complexity. While this confluence of disciplines has stimulated recent progress, it has also led to a "tower of Babel" problem, in which differences in language and methodology make it difficult to compare the results obtained. Without attempting to impose a uniform language and methodology on the field as a whole, it is our feeling that a significant part of the empirical work, namely the part that has been called *learning from examples* [CF82], can be treated in a clear, quantitative manner that is useful to the practitioner and is based on solid theoretical foundations. In this paper we summarize some of the work that we have done in our laboratory in this direction.

Of the fields mentioned above, our approach has the strongest kinship with the recent learning research by those trained in the theory of algorithms and computational complexity. Thus we view a learning strategy as an algorithm, and apply the techniques and perspectives

from this field to analyze its capabilities and performance. Early efforts along these lines were based primarily on the inductive inference model introduced by Gold [G67]. However, most work in this model has not placed sufficient emphasis on minimizing the resources required by learning algorithms to be of much use in actual learning applications. More recently, Valiant [V84] [V85] [PV88] [KLPV87] has introduced a probabilistic model for the study of learning algorithms which is often called the the PAC model (for Probably Approximately Correct learning) [Ang88]. This model has been more successful in addressing some of the requirements that are typically placed on a learning algorithm in practice.

In most formal models of learning from examples, the task is to identify an unknown target function f based on examples of that function (i.e. pairs of the form $(x, f(x))$). To allow greater emphasis on the computational efficiency of the learning algorithm, the PAC model requires only that a good approximation to the target function be found with high probability, rather than requiring exact identification of the target function. We elaborate on this model in the following section. Recent results have demonstrated that there are efficient learning algorithms that achieve this type of probably approximately correct learning for many types of target functions [KLPV87], [BEHW89], [H88].

In our work we have concentrated on refining and generalizing the PAC model. We have identified a number of ways that the performance of learning algorithms can be quantified in terms of the amount of resources that they consume in order to achieve a given level of performance. The most important resources are computation time and space, and the number of training examples used. We have tried to derive theoretical bounds that indicate the optimum performance that can be expected of any learning algorithm given particular resource constraints, and to develop algorithms that approach this optimum. We have looked for trade-offs between resources, and for general algorithm transformations that can improve the way a learning algorithm utilizes one resource without seriously degrading its utilization of others. We also look for algorithm transformations that make learning algorithms more robust to noise and other types of anomalies in the training data.

In addition to our theoretical work, we have also experimented with a number of the learning algorithms that we have developed. We have found experimental evaluation to be an important counterpart to theoretical analysis. Experiments can only estimate a learning algorithm's performance on particular distributions of training examples, whereas the theoretical bounds in the PAC model hold for any distribution. Nevertheless, experiments can sometimes provide a good indication of typical performance when theoretical analysis is intractable, and can also indicate when theoretical worst case upper bounds are overly pessimistic in practice.

Definitions

There are many ways that learning performance can be measured. In any given application, the appropriate metrics will depend largely on how the goal of learning is defined, and on what resources are deemed critical. In the PAC model, the goal of learning is to produce a good approximation to an unknown target function. For target functions that take on only two possible values (usually called *concepts*), this can be made precise as follows.

We assume that the target concept f is a $\{0,1\}$-valued function on a given domain X (called the *instance space*). Typically f is a Boolean function, i.e. the instance space X is defined by some set of Boolean attributes. Random examples of the target concept f are

generated by drawing instances independently at random from the instance space X, and for each instance x, forming the pair $(x, f(x))$. It is assumed that instances are drawn according to a fixed probability distribution P on the instance space X. We make no assumptions about the probability distribution by which the instances are selected, other than that it remains fixed and that the instances are chosen independently. In particular, the distribution is not assumed to be known to the learning algorithm.

From these random examples of the target concept f, a learning algorithm produces a *hypothesis* h, which is also a function from the instance space X to $\{0,1\}$. The *accuracy* of this hypothesis h is the probability that it will agree with the target concept f on a randomly drawn instance, i.e.

$$accuracy(h) = P(\{x \in X : h(x) = f(x)\}).$$

A good approximation to the target concept f is a hypothesis h with high accuracy.

Notice that the accuracy of the hypothesis is measured with respect to the same probability distribution that is used to generate the training examples. This is an essential part of the model. Under this assumption, it has been shown that many elementary learning strategies are guaranteed to produce a hypothesis with high accuracy with high probability, regardless of the underlying distribution P on the instances [V85],[H88],[BEHW89].

For PAC learning, the most important measures of performance are *sample complexity* and *time complexity*. The time complexity of a learning algorithm refers to the computational time it takes to produce a hypothesis from a given sequence of examples. The sample complexity of a learning algorithm is defined in terms of the number of random examples needed so that the hypothesis produced has high accuracy with high probability. It is a function that explicitly depends on the accuracy demanded of the hypothesis, and the confidence with which that accuracy is achieved. In general, the sample complexity will depend on the probability distribution that governs the generation of examples. In the PAC model, unless otherwise specified, sample complexity refers to the worst case sample complexity over any distribution that may be used to generate the training examples. Note that it is typically difficult to know in experimental learning studies just what distributions will prove the hardest for an algorithm to handle, or to know just what distributions will be encountered in particular application environments. Thus the guarantees supplied by this model on the performance of an algorithm for arbitrary distributions are of significant practical importance.

For incremental learning algorithms, i.e. algorithms that process examples one at a time and update a current hypothesis after each new example, other performance measures are relevant. One is *space efficiency*, defined in terms of the amount of memory space used to keep the current hypothesis and other data between examples. Another is *update time*, i.e. the time it takes to update the current hypothesis given a new example.

Incremental learning algorithms are usually used in settings where the current hypothesis is used to make a prediction of the value of the target concept on a given instance x, and subsequently told whether or not that prediction was correct. This type of interaction occurs whenever the module that incorporates the learning procedure is required to do useful work while it is learning, e.g. in robotics applications. We call this an *on-line* learning setting.

The on-line learning setting is also associated with another useful learning measure, the number of *mistakes* during learning. A mistake occurs whenever the prediction of the learning algorithm is incorrect. Since most on-line learning algorithms only update their hypotheses

when a mistake is made, this measure is usually the same as the number of *hypothesis updates* or *mind–changes* during learning. The classical perceptron convergence theorem gives a worst case bound on this number for the perceptron learning algorithm applied to the target class of linear threshold concepts (see e.g. [DH73]). This is not a probabilistic bound. No assumptions are made on the order in which instances are given. Our colleague Nick Littlestone has developed a new variant of the perceptron algorithm that gives better bounds in many important cases, and examined the relationship between this type of "mistake" bound and the bounds on sample complexity in PAC learning [Li87,89],[HKLW88].

A probabilistic performance measure can also be defined for the on-line setting. It is simply the probability of making a mistake on the t^{th} instance. Here we assume that the instances are drawn independently at random from some fixed distribution on the instance space, as in the standard PAC model. Since in the on-line model learning occurs after each instance, we expect the probability of making a mistake to go down at the instance number t increases. Plotting this probability of a mistake as a function of t gives what is typically called a learning curve. In [HLW88] we explore the close relationship between these learning curves and the sample complexity bounds in PAC learning.

Results

In this section we briefly discuss the results we have obtained by applying the methodology outlined in the introduction to the performance measures defined above. These results were obtained in collaboration with many of our colleagues, including Eric Baum, Anselm Blumer, Andrzej Ehrenfeucht, David Helmbold, Michael Kearns, Lenny Pitt, Bob Sloan, Les Valiant, and Emo Welzl, and our PhD students Nick Littlestone (now at Aiken Computing Lab., Harvard), Aleksandar Milosavljevic and Giulia Pagallo. Although it represents only a small fraction of the recent work in this area, due to space limitations we restrict ourselves in this report to work that was done at least partly in our laboratory. For a more complete picture of recent work in this area, we refer the reader to the proceedings of the two workshops on computational learning theory [HP89] [R89]. Our main results are as follows.

(1) The original PAC learning model was defined only for target concepts on Boolean instance spaces. We have extended this model to AI instance spaces that include real-valued attributes [BEHW89], multi-valued attributes with hierarchical value structure [H88], and structured, multi-object instances (e.g. blocks-world scenes) [H89c].

We have also generalized the model so that it can be used to analyze algorithms that learn multi-valued functions including real and vector-valued functions, as well as functions that take values in a finite or countably infinite set [H89a,b]. Here the accuracy of a hypothesis is defined as the average distance between the value that it predicts on a given random instance and the value that is observed for that instance. Thus a hypothesis with high accuracy is one that predicts values that are close to those actually observed, like a good scientific theory. There is a great deal of flexibility in how the distance between predicted and observed values can be defined, so a wide variety of performance measures can be cast in this framework, including those commonly used in statistical pattern recognition and neural net research (e.g. mean squared error, etc.) The generalized model also handles a wide variety of "noise" processes, so that one does not need to assume that the training data is generated precisely according to some underlying target function. This is seldom a realistic assumption when the

data is real-valued.

(2) For concept learning, we have provided a general combinatorial characterization of PAC learnability using the Vapnik-Chervonenkis (VC) dimension [VC71], [Vap82], [HW87], [BEHW89]. This has led to the discovery of necessary and sufficient conditions on a class of target concepts for the existence of PAC learning algorithms with polynomial sample and time complexity, and demonstrations of such algorithms for several types of concept classes [BEHW89].

Using further work of Vapnik and Chervonenkis [Vap82], and also work of Dudley [Du84] and Pollard [Po84], we have obtained related results for learning real and vector-valued functions [H89a,b]. However, here we have only sufficient conditions. Using these results, we have obtained upper bounds on the number of training examples needed for learning with feed-forward neural networks and radial basis functions [BH89] [H89a,b]. As above, here we do not need to assume there is an underlying target function.

(3) The VC dimension has also proven useful in obtaining lower bounds for sample complexity. General sample complexity lower bounds for learning target concepts given by k-DNF and k-CNF Boolean expressions, symmetric functions, decision lists [Riv87] and linear threshold functions are given in [EHKV89]. These results show that a number of important learning algorithms have sample complexity within either a constant or logarithmic factor of optimal. These algorithms include the classical AI algorithm for conjunctive concepts on instance spaces with tree-structured and linear attributes, and greedy variants of this algorithm for k-DNF, k-CNF and internal disjunctive concepts [H88].

(4) We have demonstrated that the principle of preferring the simpler hypothesis, usually called Occam's Razor, leads to provably good learning performance [BEHW87]. We have explored techniques for efficiently implementing this heuristic using a greedy algorithm [BEHW89], [H88], [H89c].

(5) We have shown that the target concept classes that have PAC learning algorithms with polynomial sample and time complexity are the same for nearly all variants of the PAC learning model that have been considered by various authors [HKLW88].

(6) For many natural concept classes no efficient learning algorithm is known. We suspect that many of these are in fact difficult to learn, but we currently lack techniques to settle the question. To gain a clearer picture, we have begun to develop a theory along the lines of the theory of NP-completeness that allows us to compare the difficulty of learning various classes. We have defined a notion of reducibility that preserves efficient learning. With respect to this type of reduction we have shown certain learning problems to be complete for standard complexity classes such as LOGSPACE, LOGCFL, and P.

As in the theory of NP-completeness, when a problem is complete for its class, then finding an efficient learning algorithm for it implies that an efficient algorithm exists for every learning problem in its complexity class. For the richer complexity classes, this strong implication provides evidence that efficient learning algorithms may not exist for these complete problems. We use completeness proofs to establish hardness results for learning various natural target classes [PW88]. Very important recent work of Kearns and Valiant also uses our notion of

reduction to show that certain fundamental learning problems, including the problem of learning finite automata, are intractable, based on cryptographic assumptions [KV89]. We also give related negative results for the problem of learning finite automata, based only on the weaker assumption that P $\neq$ NP [PW89].

(7) Nick Littlestone has developed a theory of optimal mistake-bounded algorithms for the on-line setting. These algorithms make the minimal total number of mistakes of prediction during learning for any target concept in a given target class and any sequence of instances from the instance space. He gives general constructions for optimal mistake-bounded algorithms that work for any concept class, and in a few cases yield computationally efficient algorithms [Li87]. He has also extended these results to the probabilistic setting in which the probability of a mistake on the t^{th} instance is the primary performance measure [HLW88]. Here again, new combinatorial techniques using the Vapnik-Chervonenkis dimension play an important role.

(8) One of the computationally efficient learning algorithms that Littlestone has developed is a variant of the classical perceptron learning algorithm that has a significantly better mistake bound for many important concept classes [Li87]. For disjunctions and conjunctions, he shows that its mistake bound is within a constant factor of optimal. The algorithm converges more rapidly by making small multiplicative changes to the individual weights during learning, instead of additive changes. Although the basic learning algorithm can learn only linearly separable functions, by certain transformations of the training examples it can be made to learn other types of target concepts [Li87]. In particular, it can be applied to learn k-DNF concepts with a mistake bound that is within a constant factor of optimal. This is a significant improvement over previous incremental algorithms for this target class [V85].

(9) A variant of Littlestone's algorithm called the *weighted majority algorithm* can be used more generally as a method of combining several learning algorithms into a single learning algorithm that is more powerful and more robust than any of the component algorithms [LW89]. In this scheme all of the component learning algorithms are run in parallel on the same training instances. For each instance, each algorithm makes a prediction and then these predictions are combined by a weighted voting scheme to determine the overall prediction by the "master" algorithm. After receiving feedback on its prediction, the master algorithm adjusts the weights for each of the component algorithms, increasing the weights of those that made the correct prediction, and decreasing the weights of those that predicted incorrectly. As in Littlestone's basic algorithm, these weight changes are multiplicative, and hence are different from the type of additive changes that one gets by applying the usual gradient descent techniques used in connectionist learning algorithms. With respect to the mistake bound performance measure, we have shown that this method of combining learning algorithms by weighted voting can lead to algorithms that are nearly optimally robust with respect to anomalies in the training data, and that the performance of the master algorithm approaches the performance of the best component algorithm for any given learning task.

(10) We have developed learning algorithms that apply to concept classes that include nested exceptions. These are concepts like "when to use form 1040X for your income tax" that include conditions like "use 1040X if you are married and have combined income greater than y unless you are over 65 and renting, except if you are also blind." Here there is a basic rule, then exceptions to the rule, and then exceptions to the exceptions, etc. We have shown that under certain

conditions, individual learning algorithms for each of the different types of rules can be combined into a master algorithm that learns rules with nested exceptions [HSW89]. We show that this algorithm is optimal or nearly optimal with respect to a variety of learning measures.

(11) In terms of experimental work, Giulia Pagallo has looked at the problem of learning Boolean functions with a short Disjunctive Normal Form representation using hypotheses that are decision trees [PH89a]. The novel aspect of her work is that her learning algorithm invents new attributes while it is learning, and uses these attributes to redescribe the training data at a higher level in order to facilitate the learning task (see also [Sch87]).

The algorithm, called *FRINGE*, begins by building a decision tree from the training examples in the standard way (as in Quinlan's ID3 method [Q86], see also [BFOS84]). Then it uses this decision tree to find new higher level attributes for the training examples, reexpresses the training examples using these attributes, and then repeats the process, building a decision tree from the modified training examples. She has demonstrated empirically that this algorithm outperforms the standard decision tree algorithm for learning short DNF formulae when the examples are drawn at random from the uniform distribution. Based on the work in [Riv87], she has also developed related learning algorithms for this task that use greedy methods and are somewhat more amenable to analysis [PH89b]. Further theoretical work on learning decision trees and DNF formulae is given in [EH89].

(12) In other experimental work, Aleksandar Milosavljevic has developed a clustering algorithm for data that consists of aligned sequences of letters over a finite alphabet [MHJ89]. This algorithm is intended for use in Biology, where the sequences of letters represent the chains of nucleic acids that form similar pieces of DNA, or chains of amino acids that form similar proteins. However, the principle is quite general, and may also be applicable to other types of sequences, such as those found in speech recognition and OCR applications.

The algorithm, called MASC for Multiple Aligned Sequence Classifier, is based on the minimum description length principle [Ris78], in which the classification that is preferred is the one that gives the most compact description of the data. This method can be justified as an application of Bayes' rule. The algorithm has been very successful in producing classifications that are acceptable to biologists, and is now being used as a research tool to provide initial hypotheses for new types of sequences that are being collected.

Conclusion

As mentioned above, our work represents only a small fraction of the work that has been done on the analysis of learning algorithms in the last few years. The introduction of the PAC model has been a major stimulus of new research in this area. Yet it is not the only approach that has been explored. We and others have looked at non-probabilistic mistake bound models, as well as Bayesian models such as the minimum description length principle.

The field is still young. New perspectives are emerging at a healthy rate. We don't expect the pace to slow, nor do we anticipate a quick resolution of the fundamental problems. However, we are pleased with the progress that has been made.

Acknowledgements We would like to thank Nick Littlestone and Michael Kearns for their help in preparing portions of this paper.

References

[Ang88] Angluin, D., "Queries and Concept Learning," *Machine Learning* 2 (4): 319-342, 1988.

[BEHW87] Blumer, A., A. Ehrenfeucht, D. Haussler and M. Warmuth, "Occam's Razor," *Inf. Proc. Let.*, 24, 1987, 377-380.

[BEHW89] Blumer, A., A. Ehrenfeucht, D. Haussler and M. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. ACM*, to appear.

[BFOS84] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., *"Classification and regression trees,"* Belmont: Wadsworth.

[BH89] Baum, E., and Haussler, D., "What size net gives valid generalization?," *Neural Computation*, 1 (1) (1989) 151-160.

[CF82] Cohen, P.R., Feigenbaum, E.A., "The handbook of Artificial Intelligence," William Kaufmann, 1982.

[DH73] Duda, R. and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[Du84] Dudley, R.M., "A course in empirical processes," *Lecture Notes in Mathematics*, 1097:2-142, 1984.

[EH89] Ehrenfeucht, A., D. Haussler, "Learning decision trees from random examples," *Information and Computation*, to appear.

[EHKV89] Ehrenfeucht, A., D. Haussler, M. Kearns and L.G. Valiant, "A general lower bound on the number of examples needed for learning," *Information and Computation*, to appear.

[G67] Gold, E.M., "Language identification in the limit," *Inf. Control 10, 447-474.*

[H88] Haussler, D., "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework", *Artif. Intell.*, 36 (1988) 177-221.

[H89a] Haussler, D., "Generalizing the PAC Model for Neural Net and Other Learning Applications," UCSC Tech. Rep. CRL-89-30, September 1989.

[H89b] Haussler, D., "Generalizing the PAC model: sample size bounds from metric dimension-based uniform convergence results," *Proc. 30th Symp. on Foundations of Computer Science,"* Research Triangle Park, NC, 1989, to appear.

[H89c] Haussler, D., "Learning conjunctive concepts in structural domains," *Machine Learning*, to appear.

[Hi88] Hinton, G., "Connectionist learning procedures," *Tech. Rep. CMU-CS-87-115*, Carnegie-Mellon Univ., Pittsburgh, PA, 1987, also to appear in *Artif. Intell.*

[HKLW88] Haussler, D., Kearns, M., Littlestone, N., and Warmuth, M.K., "Equivalence of models for polynomial learnability," *UC Santa Cruz Tech. Rep.* UCSC-CRL-88-6, 1988.

[HLW88] Haussler, D., Littlestone, N., Warmuth, M.K., "Predicting {0,1}-functions on randomly drawn points", *29th IEEE Symposium on Foundations of Computer Science*, White Plains, NY, October 1988, 100-109.

[HP89] Haussler, D. and L. Pitt, (Eds.) *Proc. First Workshop on Computational Learning Theory*, Cambridge, MA, August 1988, Morgan Kaufmann.

[HSW89] Helmbold, D., Sloan, R., and Warmuth, M. K., "Learning nested differences of intersection-closed concept classes," *Proc. Second Workshop on Computational Learning Theory*, Santa Cruz, California, July 1989, 41-56.

[HW87] Haussler, D., and E. Welzl, "Epsilon nets and simplex range queries," *Discrete and Computational Geometry*, 2, (1987) 127-151.

[KLPV87] Kearns, M., M. Li, L. Pitt, and L. Valiant, "Recent results on Boolean concept learning," *Proc. 4th Int. Workshop on Machine Learning*, Irvine, CA., 1987, 337-352.

[KV89] Kearns, M. and L. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata," *Proc. 21st ACM Symp. on Theory of Computing*, Seattle, WA, May 1989, 433-444.

[Li87] Littlestone, N., "Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm," *Machine Learning* 2 (4): 285-318, 1987.

[Li89] Littlestone, N., "From on-line to batch learning," *Second Workshop on Computational Learning Theory*, Santa Cruz, CA, August 1989, 269-284.

[LW89] Littlestone, N., and Warmuth, M. K. "The weighted majority algorithm." *Proc. Thirtieth Annual IEEE Symposium on Foundations of Computer Science*. October 1989, to appear.

[MCM83] Michalski, R.S., J.G. Carbonell, and T.M. Mitchell, *Machine learning: an artificial intelligence approach*, Morgan Kaufmann, 1983.

[MCM86] Michalski, R.S., J.G. Carbonell, and T.M. Mitchell, *Machine learning: an artificial intelligence approach, vol. 2*, Morgan Kaufmann, 1986.

[MHJ89] Milosavljevic, A., Haussler, D., and Jurka, J., "Informed parsimonious inference of prototypical genetic sequences," *Second Workshop on Computational Learning Theory*, Santa Cruz, CA, August 1989, 102-117.

[PH89a] Pagallo, G., and Haussler, D., "Boolean feature discovery in empirical learning." To appear in *Machine Learning*.

[PH89b] Pagallo, G., and Haussler, D., "Learning μ-DNF functions under the uniform distribution," UCSC Tech. Rep. CRL-89-12, September 1989.

[Po84] Pollard, D., *Convergence of Stochastic Processes*, Springer-Verlag, 1984.

[PW88] Pitt, L. and Warmuth, M. K., "Prediction preserving reducibility," Technical Report UCSC-CRL-88-26, UC Santa Cruz. To appear the special issue of the *Journal of Computer and System Sciences* for the *Third Annual Conference of Structure in Complexity Theory* (Washington, DC., June 1988).

[PW89] Pitt, L., and Warmuth, M. K., "The minimum consistent DFA problem cannot be approximated within any polynomial" Technical Report UCDCS-R-89-1499, University of Illinois at Urbana-Champaign. To appear in *J. ACM*.

[Q86] Quinlan, J.R., "Induction of decision trees," *Machine Learning*, 1 (1) (1986), 81-106.

[Ris78] Rissanen, J., "Modeling by shortest data description," *Automatica*, 14 (1978), 465-471.

[Riv87] Rivest, R., "Learning decision-lists," *Machine Learning* 2 (3) (1987), 229-246.

[Riv89] Rivest, R., (Ed.) *Proc. Second Workshop on Computational Learning Theory*, Santa Cruz, CA, August 1989, Morgan Kaufmann.

[RM86] Rumelhart, D.E., J.L. McClelland, and the PDP research group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, MIT Press, Cambridge, Mass., 1986.

[Sch87] Schlimmer, J. C., "Incremental adjustment of representations for learning," *Proc. 4th Int. Workshop on Machine Learning*, Irvine, 1987, 79-90.

[V84] Valiant, L.G., "A theory of the learnable," *Comm. ACM*, 27(11), 1984, pp. 1134-42.

[V85] Valiant, L.G., "Learning disjunctions of conjunctions," *Proc. 9th IJCAI*, Los Angeles, CA, August 1985, vol. 1, pp. 560-6.

[PV88] Pitt, L. and L.G. Valiant, "Computational Limitations on Learning from Examples," *J. ACM.*, 35 (1988) 965-984.

[Vap82] Vapnik, V.N., *Estimation of dependences based on empirical data*, Springer Verlag, New York, 1982.

[VC71] Vapnik, V.N. and A.Ya.Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Th. Prob. and its Appl.*, 16(2), 1971, pp. 264-80.