E4 — Machine Learning

Pedro Domingos
Department of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350
pedrod@cs.washington.edu
Tel.: 206-543-4229 / Fax: 206-543-2969

Abstract

Machine learning's focus on ill-defined problems and highly flexible methods makes it ideally suited for KDD applications. Among the ideas machine learning contributes to KDD are the importance of empirical validation, the impossibility of learning without *a priori* assumptions, and the utility of limited-search or limited-representation methods. Machine learning provides methods for incorporating knowledge into the learning process, changing and combining representations, combatting the curse of dimensionality, and learning comprehensible models. KDD challenges for machine learning include scaling up its algorithms to large databases, using cost information in learning, automating data pre-processing, and enabling rapid development of applications. KDD opens up new directions for machine learning research, and brings new urgency to others. These directions include interfacing with the human user and the database system, learning from non-attribute-vector data, learning partial models, and learning continuously from an open-ended stream of data.

E4.1 Use of machine learning methods for KDD

Machine learning is characterized by a focus on complex representations, ill-defined problems, and search-based methods. Representations studied include most of those described in Section B2, but particularly decision trees, sets of propositional or first-order rules, sets of instances, clusters, concept hierarchies and probabilistic networks. Ill-defined problems studied include generalizing from a set of tuples in the absence of a known model structure [C5.1], clustering [C5.5], combining logic theories of a domain with learning (de Raedt, 1996), and learning from delayed feedback in very large decision spaces (Sutton & Barto, 1998). Search methods [B8] used for learning include greedy search, gradient descent, expectation maximization, genetic algorithms, and some forms of lookahead and pruned breadth-first search. Other types of search frequently found in artificial intelligence, like best-first search and simulated annealing, tend to see less use in machine learning, for reasons discussed below.

The flexibility of most machine learning methods makes them well suited to applications where little is known *a priori* about the domain, and/or relevant knowledge is hard to elicit. This flexibility also means they are often able to successfully learn from data that was not gathered by a purposely designed experimental procedure, but rather obtained by some process whose end goal was not necessarily knowledge discovery. The flip side of this is that theoretical analysis of machine learning methods is often difficult, and strong guarantees regarding the correctness of results are consequently seldom available. This is compensated for by the fact that machine learning makes full use of the power of the computer to experimentally validate its methods and results. The same approach would seem to be indispensable in KDD. Standard elements of machine learning's empirical toolbox include the use of holdout sets and cross-validation [C8.1.6] to verify generalization, comparison of systems on large collections of benchmark problems (e.g., Blake, Keogh, & Merz, 1999), lesion studies to elucidate the contribution of specific system components, and experiments with carefully designed synthetic datasets to test specific hypotheses on when and why a given approach will work (Kibler & Langley, 1998).

Difficulties notwithstanding, a significant body of theory has been developed within machine learning (see Kearns and Vazirani (1994) for an introduction), and has produced highly successful practical algorithms like boosting (Freund & Schapire, 1996), Winnow (Littlestone, 1997) and support vector machines (Scholkopf, Burges, & Smola, 1998). Much of this theory is in the form of bounds on the generalization error of a learner, given its empirical error and a measure of the effective size of the hypothesis space it explores (its cardinality, for finite spaces; or its Vapnik-Chervonenkis dimension, for infinite ones (Vapnik, 1995)). The wealth of theoretical results produced is made possible by not insisting on absolute guarantees (e.g., "error will always be less than 5%"), but instead aiming for probabilistic ones (e.g., "error will be less than 5% with greater than 99% probability").

Machine learning's readiness to perform generalization in the absence of strong guiding assumptions has led it to face squarely the problem of what — and how much — is needed to generalize successfully. The lessons learned form an important part of any KDD practitioner's baggage. One is that generalization is impossible in the absence of assumptions or "biases"; purely empirical learning is a chimera (Mitchell, 1980; Schaffer, 1994; Wolpert, 1996). Induction can be seen as a "knowledge lever," with much higher leverage than deduction, but still of no use without an applied force. The converse lesson is that there is no "general-purpose" learning method; each method's utility is contingent on the assumptions it makes, and each application requires individual attention. "Universal" laws of discovery like "simple hypotheses are more accurate" (sometimes known as "Occam's razor") should be viewed with suspicion (Schaffer, 1993; Webb, 1996; Domingos, 1998). Having made the notion of bias explicit, machine learning has gone on to study the changes in bias (Gordon & desJardins, 1995) and combinations of different biases (Michalski & Wnek, 1996) that are often required for practical success. Awareness of the importance of knowledge has led to development of methods for explicitly incorporating it into the learning process. This knowledge can appear in the form of a propositional or first-order logic theory (e.g., Pazzani & Kibler, 1992; Saitta, Botta, & Neri, 1993; Ourston & Mooney, 1994; Towell & Shavlik, 1994), or in a variety of weaker forms (e.g., Clearwater & Provost, 1990; Donoho & Rendell, 1996; Pazzani, Mani, & Shankle, 1997).

Machine learning is concerned simultaneously with statistical soundness and computational efficiency. This has led it to explore issues that tend not to arise when either is considered in isolation, but that will often be of concern in KDD applications. One such issue is deciding where a KDD algorithm should fall in the lazy-eager computational spectrum. In the eager extreme, where most traditional modeling approaches fall, all generalization (and therefore most computation) is performed at learning time. In the lazy extreme, exemplified by nearest-neighbor algorithms [C5.1.6], all generalization and computation occur at performance time. Machine learning has gone between the extremes, identifying the entire lazy-eager spectrum as a useful design dimension, and proceeding to explore it (Aha, 1997). For example, the RISE system autonomously determines the best combination of rules and neighbors to use (Domingos, 1996b).

A central issue that involves both statistical soundness and computational efficiency is the effect of (often massive) search on the significance of the patterns discovered. When thousands or even millions of hypotheses are generated in the the course of search, the probability that apparently meaningful discoveries are simply the result of chance cannot be neglected. However, quantifying it is notoriously difficult. Traditional sigificance testing assumes that a single hypothesis is being tested. Techniques exist for "multiple comparison" or "simultaneous inference" problems [C8.1.2] (Miller, 1981; Klockars & Sax, 1986), but they tend to overpenalize and consequently reject valid patterns. Machine learning provides a number of techniques for assessing hypotheses in a search-conscious fashion, and controlling search to make the best use of computational power without falling into the trap of noise mining (Quinlan & Cameron-Jones, 1995; Freund, 1998; Domingos, 1999a; Jensen & Cohen, 1999). Although this is still very much an open problem, one of the hard-and-fast heuristics to emerge so far has been that apparently impoverished search methods (e.g., greedy search) are often preferable to more powerful ones (e.g., exhaustive search)(Quinlan & Cameron-Jones, 1995; Murthy & Salzberg, 1995; Dietterich, 1995). Like all subfields of computer science, machine learning is constrained by finite computational resources, but unlike most others, it is also constrained by another finite resource: the quantity of data available for learning. Either type of resource can be the bottleneck in any given application. If computation is the bottleneck, underfitting can result; if data, overfitting. In KDD projects, where large computational resources and large quantities of data are both frequent, either can be the case.

A generalization of the previous observation is that "less can be more." Machine learning researchers have found that more powerful representations do not necessarily lead to better results (Holte, 1993; Domingos & Pazzani, 1997). Flexibility can have a price in instability. This trade-off can be captured by the notions of statistical bias and variance, which were first developed for regression, but have been extended to classification (Geman, Bienenstock, & Doursat, 1992; Kong & Dietterich, 1995; Breiman, 1996b; Kohavi & Wolpert, 1996; Friedman, 1997). A related observation is that computational power is often better used to induce multiple models and combine them, instead of searching more for a single "best one." This is the approach followed by some of the best-performing learning methods available, including boosting (Freund & Schapire, 1996), bagging (Breiman, 1996a), stacking (Wolpert, 1992) and error-correcting output codes (Kong & Dietterich, 1995).

Another instance of "less can be more" is the "curse of dimensionality" (Duda & Hart, 1973). Human intuitions from the three-dimensional world fail in high dimensions. Although we might expect adding attributes to the data to improve learning, given that they provide additional information, after a point the reverse is typically the case. This is because increasing the dimension of the tuple space exponentially increases the quantity of data needed to populate it densely enough for reliable learning. This problem is particularly acute in large KDD applications where the attributes can often number in the hundreds or thousands. Machine learning provides some of the best techniques available for very high-dimensional problems (e.g., decision tree induction [C5.1.3] and rule induction [C5.1.4]) and for attribute selection (e.g., Moore & Lee, 1994).

A hallmark of machine learning is the focus on comprehensible results. While comprehensibility is difficult to define precisely and ultimately subjective, it is essential to the insights that are often the main goal of KDD. Many machine learning methods produce models that are comprehensible even to someone without mathematical training. For example, they can be sets of "If ... Then ..." rules, or in graphical form. Besides inducing such models directly, machine learning provides methods for converting less-comprehensible ones into them (e.g., neural networks into decision trees (Craven, 1996)).

E4.2 Research problems in machine learning relevant to KDD

From the point of view of accurate generalization, machine learning algorithms are often the most appropriate ones for a great variety of KDD applications. However, the volume of data available in many of these applications far outstrips the capacity of classical machine learning algorithms. Often, the solution adopted is simply to learn from a small-enough subset of the data, but the selection of this subset is typically done in a very *ad hoc* fashion — often randomly — potentially missing much of the learnable structure. The effort is now underway to enable machine learning algorithms to learn from several orders of magnitude more data than they were originally designed for.

The most basic requirement for algorithms that mine large databases is that they have linear or only slightly superlinear running time as a function of the database size. Since this is not true for most learning algorithms, it is necessary to adapt them. This can sometimes be done partly in a "lossless" fashion by optimizing the algorithms without changing their output, but typically requires a "lossy" approach: developing related algorithms that may not produce exactly the same results, but achieve similar levels of performance. Cohen's (1995) RIPPER and Domingos's (1996a) CWS algorithm are examples of this approach for the case of rule induction. Further, when the learning data is too large to fit in main memory, learning algorithms must be able to efficiently retrieve it from disk. This implies making only sequential passes through the data, as opposed to randomly accessing it, and making as few passes as possible. The SLIQ and SPRINT algorithms for decision tree induction exemplify this approach (Mehta, Agrawal, & Rissanen, 1996; Shafer, Agrawal, & Mehta, 1996), as does the Apriori algorithm for finding association rules(Agrawal, Mannila, Srikant, Toivonen, & Verkamo, 1996). The ability to learn efficiently from disk will increasingly be seen as one of the fundamental characteristics of machine learning algorithms appropriate to KDD. Ideally, algorithms should use only constant RAM and be able to learn from less than one full disk scan, making useful results available at any time after they start running (Smyth & Wolpert, 1997), and taking advantage of additional time to scan further and gracefully improve the output.

For still larger quantities of data the use of data reduction becomes inevitable. Two classes of approaches can be distinguished here: sampling and summarization. One sampling approach is to divide the examples into multiple subsets, learn on each, and combine the results (Chan & Stolfo, 1995; Breiman, 1996c). Another is to start with a small subset of the examples and iterate, adaptively selecting which new examples to include so as to obtain the maximum possible improvement from each new addition (Catlett, 1991; Musick, Catlett, & Russell, 1993). Summarization approaches attempt to produce summaries of the data that will fit in main memory, while still containing all or most of the information necessary to learn in an efficiently accessible form. These summaries may be in the form of sufficient statistics (e.g., Moore & Lee, 1997; Graefe, Fayyad, & Chaudhuri, 1998) or they may be the result of applying compression techniques to the data (e.g., Davies & Moore, 1999). The sampling and summarization approaches are complementary and can be used together (e.g., Bradley, Fayyad, & Reina, 1998).

For many problems, large quantities of data may be available, but may not be necessary to learn the desired concepts to the required level of accuracy (Oates & Jensen, 1997). For others, even the large quantities available may not be sufficient to capture all the relevant structure. It would thus be useful to have methods, even if heuristic in nature, to estimate early on how much data will be needed. Examples of research in this direction are the fitting of power laws to learning curves (Frey & Fisher, 1999) and statistical tests on the slope of these curves (Provost, Jensen, & Oates, 1999). A complementary approach is to attempt to estimate the Bayes rate, i.e., the error rate at which even an infinite-capacity learner will necessarily asymptote (Dasarathy, 1991; Cortes, 1995; Tumer & Ghosh, 1996), and to stop learning once this level is reached.

The numbers of examples, attributes, and classes presented by KDD applications effectively constitute a previously unexplored region of the machine learning space. Because of its empirical nature, the validity of much painstakingly assembled machine learning knowledge in these new circumstances is an open question. Therefore it is important to determine which elements of this knowledge need to be revised, and how.

Another type of adaptation to machine learning algorithms needed to make them useful for KDD involves aspects of KDD problems that they currently do not capture well. An example of this is cost information. Most machine learning algorithms assume that all errors have the same cost, but this is seldom the case in practice. A related problem is that of imbalanced classes: when there is a large majority of one class, it is easy to obtain high accuracy without useful results. Implicitly, misclassifying minority tuples incurs a higher cost, although this may be hard to quantify. Research on adapting machine learning algorithms for these problems is growing (e.g., Pazzani et al., 1994; Turney, 1995; Provost & Fawcett, 1997; Domingos, 1999b).

More generally, an important research direction involves methods for formulating problems in terms amenable to machine learning. Integrating, cleaning up and preprocessing the learning data is the stage of the machine learning application process that typically consumes the most time, because it is the one that requires the most human intervention. Automating this stage would produce an order-of-magnitude speedup in the process, with the corresponding reduction in cost and increase in the number of viable applications.

Although the main focus of machine learning research has been on classification problems [C5.1], a significant motivation for this has been the belief that classifiers can be used as building blocks for solutions to many other types of problems. Since many such problems are present in KDD (see Section C5), research on the interface between classification and those problems has become particularly relevant.

The perspective of widespread, large-scale application of machine learning creates the need for rapid development and deployment of learning systems. It should be possible for computer scientists with only minor knowledge of machine learning to produce a robust and reliable learning component for whatever system they are building. This requires developing libraries of standard machine learning components and of ways of putting them together. Despite a number of early developments in this direction (Gilks, Thomas, & Spiegelhalter, 1994; Buntine, 1994; Kohavi, Sommerfield, & Dougherty, 1996), for the most part it is still not clear how best to do this. Deciding what representations and techniques to use is still a "black art." The designer's personal preferences and a long trial-and-error process are often what determines the outcome. Many imprecise intuitions and rules of thumb exist, but more theoretical and empirical research is needed on what conditions favor what approaches and why, and on systematizing the current jungle of techniques. The results of this research can then be codified in a form that is easily used by non-experts, or directly incorporated into more self-sufficient learning modules.

E4.3 Impact of KDD on machine learning

KDD presents a veritable treasure of new research opportunities for machine learning. In many respects, it allows a renewed focus on problems that were original concerns of the field, but that have received decreasing attention over time, arguably in large measure due to the previous limited availability of relevant real-world datasets. Perhaps serendipitously, machine learning's powerful methods often find their most compelling applications in today's large databases, which were not available when the methods were originally developed. KDD also allows machine learning to extend its ideas and motivations in new directions, and to develop productive interfaces with disciplines like databases, statistics, human-computer interfaces, visualization, information retrieval, and high-performance computing.

Applying machine learning to the very large databases found in KDD involves qualitative changes that go beyond simply scaling up the algorithms. For example, the traditional goal of creating a model of everything that is represented in the database must often give way to finding only local patterns or deviations from the norm. Compared to the model-building case, very little theory has been developed so far for this type of problem. Current practical KDD approaches are often more concerned with efficiency than with sound generalization. Since the latter is a central concern of machine learning, analyzing and improving these approaches is potentially fertile ground for new theoretical and methodological developments. Also, if a database is too large to model in its entirety, even through the use of sampling, sufficient statistics or compression, then a "focus of attention" mechanism becomes necessary. Many heuristics and sources of information could be brought to bear on the design of such mechanisms. Further, in large databases gathered over a period of time (sometimes many years) and without learning in mind, the usual assumption of i.i.d. (independently and identically distributed) data often does not hold. Thus, an important research direction is taking into account that examples are not independent and that past, present and future data is not necessarily from the same population.

The majority of work to date in machine learning has focused on learning from examples represented as attribute vectors, where each attribute is a single number or symbol, and a single table contains all the vectors. However, much (or most) of the data in KDD applications is not of this type. For example, relational databases typically contain many different relations/tables, and performing a global join to reduce them to one without losing information is seldom computationally feasible. (Inductive logic programming (de Raedt, 1996) can handle data in multiple relations, but simultaneously focuses on learning concepts that are themselves in first-order form, thus addressing a doubly difficult problem.) The World Wide Web is mostly composed of a combination of text and HTML, plus image and audio files. The data recorded by many sensors and processes, from telescopes and Earth sensing satellites to medical and business records, has spatial and temporal structure. In the customer behavior mining applications that are of central concern to many companies, people can be hierarchically aggregated by occupation and other characteristics, products by category, etc. Simply converting data of all these types to attribute vectors before learning, as is common today, risks missing some of the most significant patterns. Although in each case traditional techniques for handling these types of data exist, they are typically quite limited in power compared to the machine learning algorithms available for the attribute-vector case, and there is much scope for extending the ideas and techniques of machine learning in this direction.

A machine learning system appropriate to future KDD applications should be able to function continuously, learning from an open-ended stream of data and constantly adjusting its behavior, while remaining reliable and requiring a minimum of human supervision. The future is likely to see an increasing number of applications of this type, as opposed to the one-shot, standalone applications common today. Early indicators of this trend are e-commerce sites that potentially respond to each new user differently as they learn his/her preferences, and systems for automated trading in the stock market. The trend is also apparent in the increasing preoccupation among corporations to instantly and continuously adapt to changing market conditions, leveraging for this purpose their distributed data gathering capabilities. While there has been some relevant research in machine learning (Widmer & Kubat, 1998), learners of this type must address several interesting new issues. One is smoothly incorporating new relevant data sources as they come online, coping with changes in them, and decoupling from them if they become unavailable. Another is maintaining a clear distinction between two types of change in the learner's evolving model(s): those that are simply the result of accumulating data and consequently progressing in the learning curve, and those that are the result of changes in the environment being modeled.

In KDD applications, learning is seldom an isolated process. More typically, it must be embedded into a larger system. Addressing the multiple problems this raises will be an opportunity for machine learning to expand its focus and its reach. The need to efficiently integrate learning algorithms with the underlying database system creates a new interface between machine learning and database research: finding query classes that can be executed efficiently while providing information useful for learning, and simultaneously finding learning approaches that use only efficiently executable queries. Some relevant questions are: What types of sampling can be efficiently supported, and how can they be used? What is the best use that can be made of a single sequential scan of the entire database? The outcome of this iterative process may be query types and learning algorithms that are both different from those known today. The interface between machine learning and databases also involves the use for learning purposes of the meta-data that is sometimes available in database systems. For example, definitions of fields and constraints between their values may be a valuable source of background knowledge for use in the learning process.

To be used to its full potential, KDD requires a well-integrated data warehouse. Assembling the latter is a complex and time-consuming process, but machine learning can itself be used to partially automate it. For example, one of the main problems is identifying the correspondences between fields in different but related databases (Knoblock & Levy, 1998) (or other data sources, like the results of Web searches (Perkowitz & Etzioni, 1995)). This problem can be formulated in learning terms: given a target schema $\{X_1, X_2, \ldots, X_n\}$ and examples of data in this schema, induce general rules as to what constitutes an X_i column. Given a table in a source schema $\{Y_1, Y_2, \ldots, Y_n\}$, the goal is now to classify each of the Y columns as one of the X's (or none), with the results for one Y potentially constraining those for the others. Data cleaning is another key aspect of building a data warehouse that offers many research opportunities for machine learning. Very large databases almost invariably contain large quantities of noise and missing fields. More significantly, noise is often of multiple types, and its occurrence varies systematically from one part of the database to another (e.g., because the data comes from multiple sources). Similarly, the causes of missing information can be multiple and can vary systematically within the database. Research enabling machine learning algorithms to deal with noise and missing data was one of the main drivers of their jump from the laboratory to widespread real-world application. However, example-independent noise and missing data are typically assumed. Modeling systematic sources of error and missing information, and finding ways of minimizing their impact, is the next logical step.

The need to produce learning results that contribute to a larger scientific or business goal leads to the research problem of finding ways to integrate these goals more deeply into the learning process, and of increasing the communication bandwidth between the learning process and its "clients" beyond simply providing (say) class predictions for new examples. The importance in KDD of interaction with the human user (expert or not) gives a new urgency to traditional machine learning concerns like comprehensibility and incorporation of background knowledge. Today's multiple KDD application domains provide a wealth of driving problems and testing grounds for new developments in this direction. Many major application domains (e.g., molecular biology, Earth sensing, finance, marketing, fraud detection) have unique concerns and characteristics, and developing machine learning algorithms specifically for each of them is likely to occupy an increasing number of researchers.

Most machine learning research to date has dealt with the well-circumscribed problem of finding a classification model given a single, small, relatively clean dataset in attribute-vector form, where the attributes have previously been chosen to facilitate learning and the end-goal (accurate classification) is simple and well-defined. With KDD, machine learning is now breaking out of each one of these constraints. Machine learning's many valuable contributions to KDD are reciprocated by the latter's invigorating effect on it. No doubt this mutually beneficial interaction will continue to develop in the future.

Acknowledgements

The author is grateful to David Aha, Tom Dietterich, Doug Fisher, Rob Holte, David Jensen, Ryszard Michalski, Foster Provost, Ross Quinlan and Lorenza Saitta for valuable comments and suggestions regarding this article.

Word count: 4150

Further reading

1. Aha, D. W. (Ed.). (1997). *Lazy learning.* Boston, MA: Kluwer. A collection of recent research on lazy learning.

2. Cohen, P. R. (1995). *Empirical methods for artificial intelligence.* MIT Press. A useful primer on the use of experiments in machine learning and other subfields of AI.

3. Dasarathy, B. W. (Ed.). (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques.* Los Alamitos, CA: IEEE Computer Society Press. One-stop source for the main papers from last three decades of pattern recognition research on learning concepts represented by sets of instances.

4. de Raedt, L. (Ed.). (1996). *Advances in inductive logic programming.* Amsterdam, the Netherlands: IOS Press. A collection of articles on learning with examples, background knowledge, and concepts expressed in a subset of first-order logic.

5. Dietterich, T. G. (1997). Machine learning research: Four current directions. *AI Magazine*, *18*(4), 97-136. An overview of recent developments in some of the main subareas of machine learning, including scaling up algorithms to large databases. A very useful complement to this article.

6. Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York, NY: Wiley. The classic textbook on statistical pattern recognition.

7. Jordan, M. I. (Ed.). (1998). *Learning in graphical models.* Boston, MA: Kluwer. Tutorials and recent research on learning with probabilistic representations.

8. Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory.* Cambridge, MA: MIT Press. An accessible introduction to the theory of machine learning.

9. Kibler, D., & Langley, P. (1998). Machine learning as an experimental science. In *Proceedings of the Third European Working Session on Learning.* London, UK: Pitman. Reprinted in Shavlik, J. W., & Dietterich, T. G. (Eds.) (1990), *Readings in machine learning*, San Mateo, CA: Morgan Kaufmann. A very useful introduction to the methodology of machine learning.

10. Langley, P. (1996). *Elements of machine learning.* San Mateo, CA: Morgan Kaufmann. A systematic introductory presentation of the field.

11. Michalski, R. S., Bratko, I., & Kubat, M. (Eds.). (1998). *Machine learning and data mining: Methods and applications.* New York, NY: Wiley. Collects a variety of recent research at the interface of machine learning and KDD.

12. Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (1983). *Machine learning: An artificial intelligence approach,* vols. 1–3. Palo Alto, CA: Tioga. A series of books containing much of the early research.

13. Michalski, R. S., & Tecuci, G. (Eds.). (1994). *Machine learning: A multistrategy approach.* San Mateo, CA: Morgan Kaufmann. Continuation of the previous series, with a focus on combining multiple machine learning biases and using background knowledge.

14. Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (Eds.). (1994). *Machine learning, neural and statistical classification.* New York, NY: Ellis Horwood. Describes a large-scale experimental comparison of many algorithms. Also contains introductions to the algorithms and discussion of their strengths and weaknesses. It is now out of print, but is available online at http://-www.amsta.leeds.ac.uk/∼charles/statlog/.

15. Mitchell, T. M. (1997). *Machine learning.* New York, NY: McGraw-Hill. The standard introductory machine learning textbook.

16. Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, *2*. An excellent overview of scaling-up research. The place to start if you're looking for a way to scale up your algorithm.

17. Quinlan, J. R. (1993). *C4.5: Programs for machine learning.* San Mateo, CA: Morgan Kaufmann. Describes the most widely used machine learning system.

18. Scholkopf, B., Burges, C., & Smola, A. (1998). *Advances in kernel methods: Support vector machines.* Cambridge, MA: MIT Press. Expanded papers from a workshop on support vector machines.

19. Shavlik, J. W., & Dietterich, T. G. (Eds.). (1990). *Readings in machine learning.* San Mateo, CA: Morgan Kaufmann. A collection of classic machine learning papers from the 1980's.

20. Sutton, R. S., & Barto, A. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press. Introduction to one of the most active research areas in machine learning, where the focus is on learning from delayed feedback.

21. Vapnik, V. N. (1995). *The nature of statistical learning theory.* New York, NY: Springer. Introduction to the Vapnik-Chervonenkis dimension, the theory of structural risk minimization, and its application to the development of support vector machines.

22. Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems.* San Mateo, CA: Morgan Kaufmann. An older textbook that also compares machine learning algorithms with alternative techniques.

The *Machine Learning* journal, published by Kluwer, is the single most important repository of research in the field. Machine learning articles also appear in the *Artificial Intelligence* journal, in the online *Journal of Artificial Intelligence Research* (http://www.cs.washington.edu/research/-jair/home.html), in the *Neural Computation* journal, in the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and others. The main conference in the field is the *International Conference on Machine Learning*, whose proceedings are published by Morgan Kaufmann. Recent machine learning research is also reported in the *European Conference on Machine Learning*, the *International Joint Conference on Artificial Intelligence*, the *National Conference on Artificial Intelligence (AAAI)*, the *European Conference on Artificial Intelligence*, the *Annual Conference on Neural Information Processing Systems*, the *International Workshop on Multistrategy Learning*, the *International Workshop on Artificial Intelligence and Statistics*, and others. Research on the theory of machine learning appears in the *International Conference on Computational Learning Theory*, the *European Conference on Computational Learning Theory*, and elsewhere. Useful online machine learning resources include: the UCI repository of machine learning databases (http://www.ics.uci.edu/-~mlearn/MLRepository.html); the list of home pages of machine learning researchers maintained by David Aha (http://www.aic.nrl.navy.mil/~aha/people.html); the online bibliographies of several subareas of machine learning maintained by Peter Turney (http://www.iit.nrc.ca/bibliographies/); the *Machine Learning List*, maintained by Michael Pazzani (mailto:ml-request@ics.uci.edu); and the *AI and Statistics List*, maintained by Doug Fisher (mailto:Majordomo@watstat.uwaterloo.ca, with "subscribe ai-stats"). Publicly-available machine learning software includes the MLC++ and Weka libraries, found respectively at http://www.sgi.com/Technology/mlc/ and http://www.cs.-waikato.ac.nz/ml/weka/.

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 307-328). Menlo Park, CA: AAAI Press.

Aha, D. W. (Ed.). (1997). Special issue on lazy learning. *Artificial Intelligence Review, 11.*

Blake, C., Keogh, E., & Merz, C. J. (1999). *UCI repository of machine learning databases* (Machine-readable data repository). Irvine, CA: Department of Information and Computer Science, University of California at Irvine. (http://www.ics.uci.edu/~mlearn/MLRepository.html)

Bradley, P. S., Fayyad, U., & Reina, C. (1998). Scaling clustering algorithms to large databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 9-15). New York, NY: AAAI Press.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, *24*, 123-140.

Breiman, L. (1996b). *Bias, variance and arcing classifiers* (Tech. Rep. No. 460). Berkeley, CA: Statistics Department, University of California at Berkeley.

Breiman, L. (1996c). *Pasting bites together for prediction in large data sets and on-line* (Tech. Rep.). Berkeley, CA: Statistics Department, University of California at Berkeley.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, *2*, 159-225.

Catlett, J. (1991). *Megainduction: Machine learning on very large databases.* Unpublished doctoral dissertation, Basser Department of Computer Science, University of Sydney, Sydney, Australia.

Chan, P. K., & Stolfo, S. J. (1995). Learning arbiter and combiner trees from partitioned data for scaling machine learning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 39-44). Montréal, Canada: AAAI Press.

Clearwater, S., & Provost, F. (1990). RL4: A tool for knowledge-based induction. In *Proceedings of the Second IEEE International Conference on Tools for Artificial Intelligence* (pp. 24-30). San Jose, CA: IEEE Computer Society Press.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 115-123). Tahoe City, CA: Morgan Kaufmann.

Cortes, C. (1995). *Prediction of generalization ability in learning machines.* Unpublished doctoral dissertation, Department of Computer Science, University of Rochester, Rochester, NY.

Craven, M. W. (1996). *Extracting comprehensible models from trained neural networks.* Unpublished doctoral dissertation, Department of Computer Sciences, University of Wisconsin – Madison, Madison, WI.

Dasarathy, B. W. (Ed.). (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques.* Los Alamitos, CA: IEEE Computer Society Press.

Davies, S., & Moore, A. (1999). Using Bayesian networks for lossless compression in data mining. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining.* San Diego, CA: ACM Press.

de Raedt, L. (Ed.). (1996). *Advances in inductive logic programming.* Amsterdam, the Netherlands: IOS Press.

Dietterich, T. G. (1995). Overfitting and undercomputing in machine learning. *Computing Surveys*, *27*, 326-327.

Domingos, P. (1996a). Linear-time rule induction. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (p. 96-101). Portland, OR: AAAI Press.

Domingos, P. (1996b). Unifying instance-based and rule-based induction. *Machine Learning*, *24*, 141-168.

Domingos, P. (1998). Occam's two razors: The sharp and the blunt. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 37-43). New York, NY: AAAI Press.

Domingos, P. (1999a). Process-oriented estimation of generalization error. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence.* Stockholm, Sweden: Morgan Kaufmann.

Domingos, P. (1999b). MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining.* San Diego, CA: ACM Press.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103-130.

Donoho, S., & Rendell, L. (1996). Constructive induction using fragmentary knowledge. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 113-121). Bari, Italy: Morgan Kaufmann.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York, NY: Wiley.

Freund, Y. (1998). Self bounding learning algorithms. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.* Madison, WI: Morgan Kaufmann.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 148-156). Bari, Italy: Morgan Kaufmann.

Frey, L. J., & Fisher, D. H. (1999). Modeling decision tree performance with the power law. In *Proceedings of Uncertainty '99: The Seventh International Workshop on Artificial Intelligence and Statistics* (pp. 59-65). Fort Lauderdale, FL: Morgan Kaufmann.

Friedman, J. H. (1997). On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, *1*, 55-77.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1-58.

Gilks, W. R., Thomas, A., & Spiegelhalter, D. J. (1994). A language and program for complex Bayesian modelling. *The Statistician*, *43*, 169-78.

Gordon, D. F., & desJardins, M. (Eds.). (1995). Special issue on evaluation and selection of biases in machine learning. *Machine Learning*, *20*(1).

Graefe, G., Fayyad, U., & Chaudhuri, S. (1998). On the efficient gathering of sufficient statistics for classification from large SQL databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 204-208). New York, NY: AAAI Press.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63-91.

Jensen, D., & Cohen, P. R. (1999). Multiple comparisons in induction algorithms. *Machine Learning.* (To appear)

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory.* Cambridge, MA: MIT Press.

Kibler, D., & Langley, P. (1998). Machine learning as an experimental science. In *Proceedings of the Third European Working Session on Learning.* London, UK: Pitman.

Klockars, A. J., & Sax, G. (1986). *Multiple comparisons.* Beverly Hills, CA: Sage.

Knoblock, C., & Levy, A. (Eds.). (1998). *Proceedings of the AAAI-98 Workshop on AI and Information Integration.* Madison, WI: AAAI Press.

Kohavi, R., Sommerfield, D., & Dougherty, J. (1996). Data mining using MLC++, a machine learning library in C++. *International Journal on Artificial Intelligence Tools*, *6*, 537-566.

Kohavi, R., & Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 275-283). Bari, Italy: Morgan Kaufmann.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 313-321). Tahoe City, CA: Morgan Kaufmann.

Littlestone, N. (1997). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, *2*, 285-318.

Mehta, M., Agrawal, A., & Rissanen, J. (1996). SLIQ: A fast scalable classifier for data mining. In *Proceedings of the Fifth International Conference on Extending Database Technology* (pp. 18-32). Avignon, France: Springer.

Michalski, R. S., & Wnek, J. (Eds.). (1996). *Proceedings of the Third International Workshop on Multistrategy Learning.* Harpers Ferry, VA: AAAI Press.

Miller, R. G., Jr. (1981). *Simultaneous statistical inference* (2nd ed.). New York, NY: Springer.

Mitchell, T. M. (1980). *The need for biases in learning generalizations* (Tech. Rep.). New Brunswick, NJ: Computer Science Department, Rutgers University.

Moore, A. W., & Lee, M. S. (1994). Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 190-198). New Brunswick, NJ: Morgan Kaufmann.

Moore, A. W., & Lee, M. S. (1997). Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, *8*, 67-91.

Murthy, S., & Salzberg, S. (1995). Lookahead and pathology in decision tree induction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1025-1031). Montréal, Canada: Morgan Kaufmann.

Musick, R., Catlett, J., & Russell, S. (1993). Decision theoretic subsampling for induction on large databases. In *Proceedings of the Tenth International Conference on Machine Learning* (pp. 212-219). Amherst, MA: Morgan Kaufmann.

Oates, T., & Jensen, D. (1997). The effects of training set size on decision tree complexity. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 254-262). Madison, WI: Morgan Kaufmann.

Ourston, D., & Mooney, R. J. (1994). Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, *66*, 273-309.

Pazzani, M., & Kibler, D. (1992). The utility of knowledge in inductive learning. *Machine Learning*, *9*, 57-94.

Pazzani, M., Mani, S., & Shankle, W. R. (1997). Beyond concise and colorful: Learning intelligible rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 235-238). Newport Beach, CA: AAAI Press.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 217-225). New Brunswick, NJ: Morgan Kaufmann.

Perkowitz, M., & Etzioni, O. (1995). Category translation: Learning to understand information on the Internet. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 930-936). Montréal, Canada: Morgan Kaufmann.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In (pp. 43-48). Newport Beach, CA: AAAI Press.

Provost, F., Jensen, D., & Oates, T. (1999). Optimal progressive sampling. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining.* San Diego, CA: ACM Press.

Quinlan, J. R., & Cameron-Jones, R. M. (1995). Oversearching and layered search in empirical learning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1019-1024). Montréal, Canada: Morgan Kaufmann.

Saitta, L., Botta, M., & Neri, F. (1993). Multistrategy learning and theory revision. *Machine Learning*, *11*, 153-172.

Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning*, *10*, 153-178.

Schaffer, C. (1994). A conservation law for generalization performance. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 259-265). New Brunswick, NJ: Morgan Kaufmann.

Scholkopf, B., Burges, C., & Smola, A. (1998). *Advances in kernel methods: Support vector machines.* Cambridge, MA: MIT Press.

Shafer, J. C., Agrawal, R., & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. In *Proceedings of the Twenty-Second International Conference on Very Large Databases* (pp. 544-555). Bombay, India: Morgan Kaufmann.

Smyth, P., & Wolpert, D. (1997). Anytime exploratory data analysis for massive data sets. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 54-60). Newport Beach, CA: AAAI Press.

Sutton, R. S., & Barto, A. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, *70*, 119-165.

Tumer, K., & Ghosh, J. (1996). Classifier combining: Analytical results and implications. In *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms* (pp. 126-132). Portland, OR: AAAI Press.

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree algorithm. *Journal of Artificial Intelligence Research*, *2*, 369-409.

Vapnik, V. N. (1995). *The nature of statistical learning theory.* New York, NY: Springer.

Webb, G. I. (1996). Further experimental evidence against the utility of Occam's razor. *Journal of Artificial Intelligence Research*, *4*, 397-417.

Widmer, G., & Kubat, M. (Eds.). (1998). Special issue on context sensitivity and concept drift. *Machine Learning*, *32*(2).

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, *5*, 241-259.

Wolpert, D. (1996). The lack of *a priori* distinctions between learning algorithms. *Neural Computation*, *8*, 1341-1390.