3519

A SCIENTIFIC APPROACH TO PRACTICAL INDUCTION

by

Larry Rendell

May 1985

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801-2987

This paper will appear in the Proceedings of the Third International Machine Learning Workshop, Rutgers University, June 24-26, 1985

This work was supported in part by the National Science Foundation under Grant No. DCR 84-06801 and by an operating grant from the Natural Sciences and Engineering Research Council of Canada.

A SCIENTIFIC APPROACH TO PRACTICAL INDUCTION

LARRY RENDELL

Department of Computer Science, University of Illinois at Urbana-Champaign,
1304 West Springfield Avenue, Urbana, Illinois 61801

ABSTRACT

The purpose of *practical induction* is to create systems for powerful (efficient and effective) generalization learning. This paper argues that a *scientific* approach to practical induction promotes discovery of essential principles. Some have emerged from development of the author's learning systems, which have contributed promising methods and unique results.

INTRODUCTION (Induction in Science and in Machine Learning)

A scientist creates and tests intelligent hypotheses. Experiment may falsify an hypothesis H; on the other hand repeated testing may support H—i.e. raise its *credibility* [26]. For example, H might be "Localization of credit improves machine learning." Because many implementations seem to support this, we tend to believe it (although it might be interesting to determine details [23]). The resources of science are limited, so we strive to direct efforts well, and we develop disciplines (methodologies) for this end. Powerful methodologies are both *efficient* and *effective*: they avoid poor hypotheses and promote discovery of credible ones.

Analogue in machine learning. Hypothesis formation is *induction*, which AI tries to mechanize. [1] In theory, induction presents no problem: hypotheses can simply be generated and tested [2, 26]. In practice, however, the problem is so complex that effective and efficient methods for limiting search are imperative.

Practical induction: power = effectiveness + efficiency. The study of *practical induction* in machine learning has two broad goals: construction of *powerful* (effective and efficient) representations and algorithms, and discovery of principles underlying this power. Aspects include scope of application, noise management, computational complexity, convergence to optimal control structures, etc. [2, 6, 9, 21].

Search for principles. What are the essential ingredients of a powerful inductive system? In confronting this question, some researchers have synthesized systems and created models, although this work is just beginning [2, 4, 6, 10, 20]. Despite the elusiveness of powerful induction, unified models have been aided by well-conceived systems. As is typical of science and engineering, theory guides design and experiment, which in turn hones theory.

1. Mechanized induction inputs *events* or *objects* and produces *classes* or *concepts* for prediction of future events. The importance of automated induction has been emphasized in, e.g. [10].

Thesis of this paper. In addition to experimentation, scientific methodology includes appropriate abstraction, inclination toward elegant theory, and determination of important relationships (which often become quantitative). The next section of this paper presents an abstraction useful for automated generalization learning. The third section analyzes inductive power. Throughout, we shall argue for *scientific* investigation of mechanized induction.

POWERFUL CLUSTERING (Ideas, Methods, and Clarifications)

Task utility for inductive guidance. The *utility* is directly related to domain of application [13, 16, 21]. Various measures are possible. Utility may be the value of an object in task performance, and it may be probabilistic [13-21]. The probability of task usefulness collapses to set membership in deterministic cases (a probabilistic utility subsumes positive and negative examples of a concept) [21]. The system *PLS* uses probabilistic methods to induce probabilistic utility, as utility provides a bridge between domain and induction.[1] Utility embodies ideas of active, goal-directed perception [3, 8] which can contribute to inductive power.

Constraints. Inductive power is related to restrictions imposed on data specification, on forms of classes or concepts, and on algorithmic processing [2, 21, 26]. For example, features (attributes of objects) selected by the user, are designed to compress data even before any mechanized induction [19, 20]. Further, utility almost always bears a smooth relationship to user-selected features. This allows meaningful *clustering* of objects in local neighborhoods of feature space. See [21] for further discussion and more references.

Cluster analysis. In our view, Samuel designed signature tables to compress similar utilities into feature space cells [25]. Much of this was not automated, whereas PLS1 and PLS2 mechanize the clustering. Cluster analysis is an established statistical technique for inductive inference which partitions similar objects into distinctive classes. Similarities and distinctions are formalized by the use of some *(dis)similarity criterion*. Normally the criterion depends

---

1. The following is a sketch of *probabilistic learning systems PLS* (see [6, 13-23] for details):
Basic system capability. The original *PLS1* is capable of efficient and effective generalization learning in domains for which features (attributes) can be defined and *utility* (performance) can be measured. PLS1 can handle noise, selecting features which are most discriminating despite error. While it can be applied to single concept learning [6], the system has been developed and tested in the difficult domain of heuristic search, which requires not only noise management, but also incremental learning and removal of bias from data acquired during task performance. The power of PLS1 has been demonstrated in comparisons with alternative methods [14, 23]. The system can discover optimal evaluation functions, a unique result [16, 20, 23].
System extension. *PLS2* is a doubly layered learning system which uses both PLS1 and a genetic algorithm [7]. PLS2 Operations performed on utility clusters include generalization, specialization, and reorganization. PLS2 is more stable, accurate, and efficient than its predecessor [18, 23].
A system for creation of new terms. A more ambitious project involves the sophisticated system *PLS0*, designed for substantial *constructive induction* [20,22]. PLS0 uses knowledge layering and invariance of utility surfaces to create concepts from progressively validated components. This system appears suitable for problems which were previously intractable [22].

only on features, but this simplification can cause problems [2].

New kinds of clustering (Utility, Conceptual, and Higher-dimensional). Criteria based on something other than features are *external* criteria [1, p. 194]. Several years ago the author introduced *utility similarity* as a suitable external criterion when the induction relates to performance of some task [13, 14, 16, 21]. Utility similarity involves the whole *data environment*, not just features. Utility provides a firm basis for *conceptual cohesiveness* [10].

Clusters may be constrained, e.g. PLS uses feature space rectangles—conjunctions of attribute ranges. Compressing data into preconceived forms is *conceptual clustering* [10].

PLS0, the author's system for substantial constructive induction, originates a kind of clustering which groups not just attributes, or even simple utilities, but rather utility *surfaces* in subspaces of very primitive features. These surfaces represent interrelationships among components of objects. The process of clustering utility surfaces *creates structure* [22].

Disguised conceptual clustering. Superficially, Quinlan's ID3 [14] is different from Michalski's systems [10], or from the author's PLS1. But ID3 is a veiled form of utility clustering. ID3 selects attributes having the greatest ability to discriminate. So does PLS1. The *utility dissimilarity* of PLS1 is essentially the *information* of ID3. Once ID3 chooses an attribute, it constructs one branch of the discrimination tree for each attribute value. In contrast, the clustering algorithm of PLS1 splits sets of attribute values only when discrimination is thereby improved. This suggests an obvious modification of ID3, and argues for continued syntheses like [2, 6, 19, 20]. Utility is the sole basis for clustering in PLS1 and "clustering" in ID3.

WHAT PRODUCES POWER? (Principles)

This section suggests a few incipient principles which may underlie inductive power. All paragraphs but the last refer to *mechanized* induction.

Mediating structures. Discussed further in [20, 22], this is a proposed addition to Buchanan's model [4]. Successful systems tend to incorporate knowledge structures which *mediate* objects and concepts during inductive processing. These structures are varied. One codes growing assurance of provisional hypotheses (through probabilistic information in PLS1). Another mediating structure houses components of tentative concepts (in PLS0). PLS0 employs divide and conquer techniques to build knowledge in chunks of increasing complexity [20, 22]. Hypotheses, gradually and tentatively constructed on lower levels, become confirmed elements of higher level concepts. Consequently the time complexity is improved [22].

Representation of whole sets of hypotheses using boundaries. Mitchell's deterministic candidate elimination for version spaces [11] is efficient because limited boundaries represent whole sets of hypotheses (the boundaries gradually converge). The author's PLS1 is efficient (yet cautious) because tentative boundaries represent the restricted set of partially confirmed hypotheses (boundaries provisionally converge, with increasing assurance).

Multiple use of single events in credit localization. In traditional methods of optimization (e.g. hill climbing, response surface fitting), solving a problem contributes only a single datum. In contrast, probabilistic learning systems like Samuel's checker player and PLS1 make use of every single event (e.g. each state in heuristic search). No one event can errantly overwhelm the system, but still, each one updates knowledge about *every* feature or feature space cell. A similar situation arises in PLS0, only it is much more pronounced. Here a single object provides information about a myriad of object components. (PLS0 focuses on the important ones.) This is reminiscent of *schemata* in genetic algorithms: a single structure codes and supports many combinations and generalizations of its components [7].

Mutual data support. As in the previous paragraph, this involves multiple use of scarce information for the inductive process. *Mutual data support* is a term coined by the author to express a subtle combination of phenomena. In many generalization algorithms (e.g. curve fitting, clustering), the agglomeration of similar events *simultaneously* promotes data compression, noise management, accuracy improvement, and concept formation. Mutual data support appears in various forms in all PLS systems. See [15-23], particularly [20, 22].

Proper system assessment. (How much knowledge is acquired?) This point refers not to mechanized induction, but to *our* inference about the power of systems. Precise assessment is important, not simply to know which methods are better, but also to help discover *why* they work well, in order to improve models, theories and designs. We need standards for answering questions such as: How difficult is the inductive task being studied? How much knowledge is acquired autonomously, versus the amount given by the user [21, 24]? To scientifically assess substantial learning in systems like PLS0, we need to quantify *inductive difficulty* of environments and *inductive power* of systems [19, 20, 21, 22]. This suggests analysis of computational complexity, and measurement of cost effectiveness.

CONCLUSIONS (Suitable scholarship)

In addition to specific methods, results, and contentions in or about mechanized practical induction (generalization learning), we have given a number of suggestions for scientific research in the field: Discovering equivalences in knowledge representations and algorithms is important for clear progress. So is quantification of the power of systems. Our machine learning investigations can also benefit from theoretical issues and results [2]. One example is the highly developed work on credibility criteria by Watanabe [26, pp. 154 ff.].

REFERENCES

[1] Anderberg, M.R., *Cluster Analysis for Applications*, Academic Press, 1973.

[2] Angluin, D., and Smith, C.H., Inductive inference: Theory and methods, *ACM Computing Surveys 15*, 3 (Sept. 1983), 237-269.

(3) Arbib, M.A., *The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory,* Wiley, 1972.

(4) Buchanan, B.G., Johnson, C.R., Mitchell, T.M., and Smith, R.G., Models of learning systems, in Belzer, J. (Ed.), *Encyclopedia of Computer Science and Technology 11* (1978), 24-51.

(5) Coles, D. and Rendell, L.A., Some issues in training learning systems and an autonomous design, *Proc. Fifth Biennial Conference of the Canadian Society for Computational Studies of Intelligence,* 1984, 99-102.

(6) Dietterich, T.G., London, B., Clarkson, K., and Dromey, G., Learning and inductive inference, STAN-CS-82-913, Stanford University, also Chapter XIV of *The Handbook of Artificial Intelligence,* Cohen, P.R., and Feigenbaum, E.A. (Ed.), Kaufmann, 1982.

(7) Holland, J.H., *Adaptation in Natural and Artificial Systems,* University of Michigan Press, 1975.

(8) Koestler, A., *The Act of Creation: A Study of the Conscious and Unconscious in Science and Art,* Macmillan (hard cover) or Dell (paperback), 1964. Other editions do not contain the important Book Two.

(9) Lenat, D.B., The nature of heuristics, *Artificial Intelligence 19* (1982), 189-249.

(10) Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (Ed.), *Machine Learning: An Artificial Intelligence Approach,* Tioga, 1983.

(11) Mitchell, T.M., Version spaces: A candidate elimination approach to rule learning, *Proc. Fifth International Joint Conference on Artificial Intelligence,* 1977, 305-310.

(12) Quinlan, J.R., Learning efficient classification procedures and their application to chess end games, in [12] 463-482.

(13) Rendell, L.A., A method for automatic generation of heuristics for state-space problems, Dept of Computer Science CS-76-10, University of Waterloo, 1976.

(14) Rendell, L.A., An adaptive plan for state-space problems, Dept of Computer Science CS-81-13, (PhD thesis), University of Waterloo, 1981.

(15) Rendell, L.A., State-space learning systems using regionalized penetrance, *Proc. Fourth Biennial Conference of the Canadian Society for Computational Studies of Intelligence,* 1982, 150-157.

(16) Rendell, L.A., A new basis for state-space learning systems and a successful implementation, *Artificial Intelligence 20,* 4 (1983), 369-392.

(17) Rendell, L.A., A learning system which accommodates feature interactions, *Proc. Eighth International Joint Conference on Artificial Intelligence,* 1983, 469-472.

(18) Rendell, L.A., A doubly layered, genetic penetrance learning system, *Proc. Third National Conference on Artificial Intelligence,* 1983, 343-347.

(19) Rendell, L.A., Toward a unified approach for conceptual knowledge acquisition, *AI Magazine 4,* 4 (Winter 1983), 19-27.

(20) Rendell, L.A., Conceptual knowledge acquisition in search, University of Guelph Report CIS-83-15, Dept. of Computing and Information Science, Guelph, Ontario, Canada, November, 1983 (to appear in Bolc, L. (ed.), *Knowledge Based Learning Systems,* Springer-Verlag).

(21) Rendell, L.A., Utility patterns as criteria for efficient generalization learning, *Proc. 1985 Conference on Intelligent Systems and Machines,* 1985 (to appear).

(22) Rendell, L.A., Substantial constructive induction using layered information compression: Tractable feature formation in search, *Proc. Ninth International Joint Conference on Artificial Intelligence,* 1985 (to appear).

(23) Rendell, L.A., Genetic plans and the probabilistic learning system: Synthesis and results (submitted).

(24) Ritchie, G.D. and Hanna, F.K., AM: a case study in AI methodology, *Artificial Intelligence 23* (1984), 249-268.

(25) Samuel, A.L., Some studies in machine learning using the game of checkers II—recent progress, *IBM J. Res. and Develop. 11* (1967) 601-617.

(26) Watanabe, S., *Knowing and Guessing: A Formal and Quantitative Study,* Wiley, 1969.

IOGRAPHIC DATA ET	1. Report No. UIUCDCS-R-85-1211	2.	3. Recipient's Accession No.
e and Subtitle A Scientific Approach to Practical Induction			5. Report Date May 1985
			6.
tor(s) Larry Rendell			8. Performing Organization Rept. No.
rforming Organization Name and Address Department of Computer Science University of Illinois Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. NSF DCR 84-06801
ponsoring Organization Name and Address National Science Foundation, Washington, D.C. and Natural Sciences and Engineering Research Council of Canada			13. Type of Report & Period Covered
			14.

Supplementary Notes

Abstracts

he purpose of *practical induction* is to create systems for powerful (efficient and ffective) generalization learning. This paper argues that a *scientific* approach o practical induction promotes discovery of essential principles. Some have merged from development of the author's learning systems, which have contributed romising methods and unique results.

Key Words and Document Analysis. 17a. Descriptors

machine learning
inductive inference
conceptual clustering

Identifiers/Open-Ended Terms

COSATI Field/Group

vailability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 5
	20. Security Class (This Page UNCLASSIFIED	22. Price

NTIS-35 (10-70) USCOMM-DC 40329-P71