Abstracts of Select Recent Articles by Russell Greiner

[from Aug'90]

Available from `ftp://scr.siemens.com/pub/learning/Papers/greiner/`
or `http://www.cs.toronto.edu/~greiner/`

1. R. Greiner, A. Grove and A. Kogan, "Knowing What Doesn't Matter: Exploiting the Omission of Irrelevant Data", *Artificial Intelligence*, to appear (1997). {`superfluous-journal.ps`}

   (Expanded from

   R. Greiner, A. Grove and A. Kogan, "Exploiting the Omission of Irrelevant Data", Thirteenth International Conference on Machine Learning (IMLC96)*, Bari Italy, July, 1996.* {`exploit-omit-imlc96.ps`}

   R. Greiner, T. Hancock and R.B. Rao, "Exploiting the Absence of Irrelevant Information: What You Don't Know Can *Help You",* AAAI Fall Symposium on 'Relevance'*, New Orleans, November 1994.* {`superfluous.ps`} *)*

   Most learning algorithms work most effectively when their training data contain *completely specified* labeled samples. In many diagnostic tasks, however, the data will include the values of only some of the attributes; we model this as a *blocking* process that hides the values of those attributes from the learner. While blockers that remove the values of critical attributes can handicap a learner, this paper instead focuses on blockers that remove only *irrelevant* attribute values, *i.e.*, values that are *not needed to classify an instance*, given the values of the other unblocked attributes. We first motivate and formalize this model of "superfluous-value blocking," and then demonstrate that these omissions can be useful, by proving that certain classes that seem hard to learn in the general PAC model — *viz.*, decision trees and DNF formulae — are trivial to learn in this setting. We also show that this model can be extended to deal with (1) theory revision (*i.e.*, modifying an existing formula); (2) blockers that occasionally include superfluous values or exclude required values; and (3) other corruptions of the training data.

2. R. Greiner: "The Complexity of Revising Logic Programs", *The Journal of Logic Programming*", to appear, 1997. {`impure-journal.ps`}

   (Expanded from

   R. Greiner: "The Challenge of Revising an Impure Theory", Twelfth International Conference on Machine Learning (IML95)*, Lake Tahoe, July 1995.* {`impure-ml95.ps`}*)*

   A rule-based program will return a set of answers to each query. An *impure* program, which includes the Prolog `cut` "!" and "`not(·)`" operators, can return different answers if its rules are re-ordered. There are also many reasoning systems that return only the *first* answer found for each query; these first answers, too, depend on the rule order, even in pure rule-based systems. A theory revision algorithm, seeking a revised rule-base whose *expected accuracy*, over the distribution of queries, is optimal, should therefore consider modifying the order of the rules. This paper first shows that a polynomial number of training "labeled queries" (each a query paired with its correct answer) provides the distribution information necessary to identify the optimal ordering. It then proves, however, that the task of determining which ordering is optimal, once given this distributional information, is intractable even in trivial situations; *e.g.*, even if each query is an atomic literal, we are seeking only a "perfect" theory, and the rule base is propositional. We also prove that this task is not even approximable: Unless $P = NP$, no polynomial time algorithm can produce an ordering of an n-rule theory whose accuracy is within n^γ of optimal, for some $\gamma > 0$. We next prove similar hardness, and non-approximatability, results for the related tasks of determining, in these impure contexts, (1) the optimal *ordering of the antecedents*; (2) the optimal set of *new rules to add*; and (3) the optimal set of *existing rules to delete*.

3. R. Greiner, "The Complexity of Theory Revision", *Artificial Intelligence*, accepted subject to revision, 1997. {`pure-journal.ps`}

    (Expanded from
    R. Greiner, "The Complexity of Theory Revision", Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95)*, Montreal, August 1995.* {`comp-tr-ijcai95.ps`} *)*

    A knowledge-based system uses its database (a.k.a. its "theory") to produce answers to the queries it receives. Unfortunately, these answers may be incorrect if the underlying theory is faulty. Standard "theory revision" systems use a given set of "labeled queries" (each a query paired with its correct answer) to incrementally transform the given theory, by adding and/or deleting either rules and/or antecedents, into a related theory that is as accurate as possible. After formally defining the theory revision task and bounding the sample complexity for this process, this paper addresses the computational complexity of this task. It first proves that, unless $P{=}NP$, no polynomial time algorithm can identify this near-optimal theory, even given the exact distribution of queries, except in the most trivial of situations. It also shows that, except in such trivial situations, no polynomial-time algorithm can produce a theory whose inaccuracy is even close (*i.e.*, within a particular polynomial factor) to optimal. These results justify the standard practice of hill-climbing to a locally-optimal theory, based on a given set of labeled samples.

4. R. Greiner, A. Grove and D. Schuurmans, "Learning Bayesian Nets that Perform Well", *Thirteen Conference on Uncertainty in Artificial Intelligence (UAI97)*, Provdence, Aug, 1997. {`ggs-bnpw-uai97.ps`}

    A Bayesian net (BN) is more than a succinct way to encode a probabilistic distribution; it also corresponds to a function used to answer queries. A BN can therefore be evaluated by the accuracy of the answers it returns. Many algorithms for learning BNs, however, attempt to optimize another criterion (usually likelihood, possibly augmented with a regularizing term), which is independent of the distribution of queries that are posed. This paper takes the "performance criteria" seriously, and considers the challenge of computing the BN whose performance — read "accuracy over the distribution of queries" — is optimal. We show that many aspects of this learning task are more difficult than the corresponding subtasks in the standard model.

5. T. Scheffer, R. Greiner and C. Darken, "Why Experimentation can be better than 'Perfect Guidance' ", *Fourteennth International Conference on Machine Learning (IMLC96)*, Nashville, Tennesee, July, 1997. {`sgd-experiment-ml97.ps`}

    Many problems correspond to the classical control task of determining the appropriate control action to take, given some (sequence of) observations. One standard approach to learning these control rules, called *behavior cloning*, involves watching a perfect operator operate a plant, and then trying to emulate its behavior. In the *experimental learning* approach, by contrast, the learner first guesses an initial operation-to-action policy and tries it out. If this policy performs sub-optimally, the learner can modify it to produce a new policy, and recur.

This paper discusses the relative effectiveness of these two approaches, especially in the presence of perceptual aliasing, showing in particular that the experimental learner can often learn more effectively than the cloning one.

6. R. Greiner, T. Petsche, and S. Hanson (ed), *Computational Learning Theory and Natural Learning Systems — Volume IV: Making Learning Systems Practical*, MIT Press, 1997 (ISBN 0-262-57118-8) {`http://www-mitpress.mit.edu/mitp/recent-books/comp/greop.html`}.

   From the introduction:

   > This is the fourth and final volume of papers from a series of workshops called "Computational Learning Theory and 'Natural' Learning Systems." The purpose of the workshops was to explore the emerging intersection of theoretical learning research and natural learning systems. The workshops drew researchers from three historically distinct styles of learning research: computational learning theory, neural networks, and machine learning (a subfield of AI).
   >
   > [...]
   >
   > The editors divide the twenty-one contributions into four sections. The first three cover critical problem areas: (1) scaling up from small problems to realistic ones with large input dimensions, (2) increasing efficiency and robustness of learning methods, and (3) developing strategies to obtain good generalization from limited or small data samples. The fourth section discusses examples of real-world learning systems.

7. R. Greiner, "PALO: A Probabilistic Hill-Climbing Algorithm", *Artificial Intelligence*, 84:1–2 (July 1996). {`palo-aij.ps`}

   (Expanded from
   R. Greiner, "Probabilistic Hill-Climbing: Theory and Applications", Ninth Canadian Conference on Artificial Intelligence (CSCSI-92)*, Vancouver, May 1992.* {`palo-applic.ps`}
   Awarded "Best Paper Prize"

   W. Cohen, R. Greiner and D. Schuurmans, "Probabilistic Hill-Climbing", in Computational Learning Theory and Natural Learning Systems: Intersections between Theory and Experiment*, Volume 2, MIT Press, 1992.* {`clnl-phc.ps`} *)*

   Many learning tasks search through a space of possible performance elements, seeking an element with high expected utility. As the task of finding the globally optimal element is usually intractable, many practical learning systems use hill-climbing to find a local optimum. Unfortunately, even this is difficult, as it depends on the distribution of problems, which is typically unknown. This paper address the task of approximating this hill-climbing search when the utility function can only be estimated by sampling. We present an algorithm that returns an element that is, with provably high probability, essentially a local optimum. We then demonstrate the generality of this algorithm by sketching three meaningful applications, that respectively find an element whose efficiency, accuracy or completeness is nearly optimal. These results suggest approaches to solving the utility problem from explanation-based learning, the multiple extension problem from nonmononotic reasoning and the tractability/completeness tradeoff problem from knowledge representation.

8. R. Greiner, A. Grove and D. Roth, "Learning Active Classifiers", *Thirteenth International Conference on Machine Learning (IMLC96)*, Bari Italy, July, 1996. {`active-class-imlc96.ps`}

   Many classification algorithms are "passive", in that they assign a class-label to each instance based only on the description given, even if that description is incomplete. In contrast, an *active* classifier can — at some cost — obtain the values of missing attributes, before deciding upon a class label. The expected utility of using an active classifier depends on both the cost required to obtain the additional attribute values and the penalty incurred if it outputs the wrong classification. This paper considers the problem of *learning* near-optimal active classifiers, using a variant of the probably-approximately-correct (PAC) model. After defining the framework — which is perhaps the main contribution of this paper — we describe a situation where this task can be achieved efficiently, but then show that the task is often intractable.

9. R. Greiner and R. Isukapalli, "Learning to Select Useful Landmarks", *IEEE Transactions on Systems, Man and Cybernetics*, Special issue on Learning Approaches to Autonomous Robots Control (ed., Marco Dorigo), 26:6, June 1996, p. 437–449. {`useful-lms-smc.ps`}

   (Expanded from
   R. Greiner and R. Isukapalli, "Learning to Select Useful Landmarks", Annual National Conference on Artificial Intelligence (AAAI-94)*, Seattle, August 1994.* {`useful-lms-aaai.ps`}*)*

   To navigate effectively, an autonomous agent must be able to quickly and accurately determine its current location. Given an initial estimate of its position (perhaps based on dead-reckoning) and an image taken of a known environment, our agent first attempts to locate a set of landmarks (real-world objects at known locations), then uses their angular separation to obtain an improved estimate of its current position. Unfortunately, some landmarks may not be visible, or worse, may be confused with other landmarks, resulting in both time wasted in searching for invisible landmarks, and in further errors in the agent's estimate of its position. To address these problems, we propose a method that uses previous experiences to learn a selection function that, given the set of landmarks that might be visible, returns the subset which can probably be found correctly, and so provide an accurate registration of the agent's position. We use statistical techniques to prove that the learned selection function is, with high probability, effectively at a local optimal in the space of such functions. This report also presents empirical evidence, using real-world data, that demonstrate the effectiveness of our approach.

10. R. Greiner and P. Orponen, "Probably Approximately Optimal Satisficing Strategies", *Artificial Intelligence*, 82:1–2 (Apr 1996), pp. 21–44. {`pao.ps`}

    (Expanded from
    P. Orponen and R. Greiner, "On the Sample Complexity of Finding Good Search Strategies" Proceedings of Third Workshop on Computational Learning Theory (COLT-90)*, Aug 1990.*
    {`colt_search.ps`}*)*

    A *satisficing search problem* consists of a set of probabilistic experiments to be performed in some order, seeking a satisfying configuration of successes and failures. The expected cost of performing the experiments depends both on the success probability of the individual experiments, and on the *search strategy*, which specifies the order in which the experiments are to be performed. A strategy is *optimal* if its expected cost is minimal. Earlier work has provided "optimizing functions" that find optimal strategies for certain classes of search problems, when given the probabilities. This paper extends those results by addressing the complexities of identifying an approximately optimal strategy when the probability values are not known, within a fully general model. We present an algorithm PAO that first obtains estimates of the probabilities by observing a computed number of trials of each undetermined experiment, and then uses these estimates (and the proper optimization function) to identify a strategy whose cost is, with high probability, close to optimal. We also show that PAO can gather the necessary statistics while solving relevant problems.

11. D. Schuurmans and R. Greiner: "Practical PAC Learning", *Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95)*, Montreal, August 1995. {`pracpac-ijcai95.ps`}

    We present new strategies for "probably approximately correct" (PAC) learning that use fewer training examples than previous PAC-learning techniques, while maintaining the exact same worst-case guarantees. We achieve this improvement by using *sequential* learning procedures that observe training examples one-at-a-time, and decide "on-line" whether to halt and return a hypothesis, or continue training. We first provide a theoretical worst-case upper-bound on the average number of training examples these procedures will use, and show that it is often better than the fixed number required by previous "batch" learning algorithms. However, the efficiency of our approach is primarily established by a series of experiments which show a sequential learning algorithm that uses *many times* fewer training examples in practice than previous techniques. These results are robust to changes in the target concept, domain distribution, etc., and moreover, the reductions are obtained with little (additional) computational overhead. Overall, these results demonstrate how PAC-learning can be achieved far more efficiently in practice than previously thought, even in the worst case, countering the claim that PAC-learning can never be feasibly achieved in real applications.

12. D. Schuurmans and R. Greiner: "Sequential PAC Learning", *Eighth Annual Conference on Computational Learning Theory (COLT95)*, Santa Cruz, July 1995. {`seqpac-colt95.ps`}

    We consider the use of "on-line" stopping rules to reduce the number of training examples needed to PAC-learn. Rather than collect a large training sample that can be proved sufficient to eliminate all bad hypotheses *a priori*, the idea is instead to observe training examples one-at-a-time and decide "on-line" whether to stop and return a hypothesis, or continue training. The primary benefit of this approach is that we can detect when a hypothesizer has actually "converged" and halt training before the conservative fixed-sample-size bounds. This paper presents a series of such *sequential* learning procedures for: distribution-*free* PAC-learning, "mistake-bounded to PAC" conversion, and distribution-*specific* PAC-learning, respectively. We analyze the worst case *expected* training sample size of these procedures, and show that this is often smaller than existing fixed sample size bounds — while still providing the *exact same* worst case PAC-guarantees. We also provide lower bounds that show these reductions can at best involve constant (and possibly log) factors. However, empirical studies show these sequential learning procedures actually use *many times* fewer training examples in practice.

13. P. Langley, G. Drastal, R.B. Rao and R. Greiner, "Theory Revision in Fault Hierarchies", *Fifth International Workshop on Principles of Diagnosis (Dx94)*, New York, October 1994. {`th-rev.ps`}

    The fault hierarchy representation is widely used in expert systems for the diagnosis of complex mechanical devices. On the assumption that an appropriate bias for a knowledge representation language is also an appropriate bias for learning in this domain, we have developed a theory revision method that operates directly on a fault hierarchy. This domain presents several challenges: A typical training instance is missing most feature values, and the pattern of missing features is significant, rather than merely an effect of noise. Moreover, features that are present may be corrupted. Finally, the accuracy of a candidate theory is measured by considering both the sequence of tests required to arrive at a diagnosis, and its agreement with the diagnostic endpoints provided by an expert. We describe the Δ algorithm for theory revision of fault hierarchies, its application in knowledge base maintenance, and report on experiments with a diagnostic system in current use.

14. D. Schuurmans and R. Greiner, "Learning Default Concepts", *Tenth Canadian Conference on Artificial Intelligence (CSCSI-94)*, Banff, May 1994. {`default.ps`}

    Classical concepts, based on necessary and sufficient defining conditions, cannot classify logically insufficient object descriptions. Many reasoning systems avoid this limitation by using "default concepts" to classify incompletely described objects. This paper addresses the task of learning such *default concepts* from observational data. We first model the underlying performance task — classifying incomplete examples — as a probabilistic process that passes random test examples through a "blocker" that can hide object attributes from the classifier. We then address the task of learning accurate default concepts from random training examples. After surveying the learning techniques that have been proposed for this task in the machine learning and knowledge representation literatures, and investigating their relative merits, we present a more data-efficient learning technique, developed from well-known statistical principles. Finally, we extend Valiant's PAC-learning framework to this context and obtain a number of useful learnability results.

15. D. Schuurmans and R. Greiner, "Fast Distribution-Specific Learning", *Computational Learning Theory and Natural Learning Systems: Volume IV: Making Learning Systems Practical*, MIT Press, 1996. {`fast-disp-spec.ps`}

    PAC-learning results are often criticized for demanding impractically large training samples. The common wisdom is that these large samples follow from the worst case nature of the analysis, and therefore PAC-learning, though desirable, must not be a practical goal. We however consider an alternative view: perhaps these large sample sizes are due to the presumed learning strategies which make inefficient use of the available training data.

    To demonstrate this, we consider *sequential* learning strategies that autonomously decide when to stop training based on observing training examples as they arrive. We show that for distribution specific learning these algorithms require far fewer training examples (on average) than existing fixed sample size approaches, and are able to learn with *certainty* not just high probability. In fact, a simple sequential strategy is *optimally* efficient in many cases.

16. D. Schuurmans and R. Greiner, "Learning to Classify Incomplete Examples", *Computational Learning Theory and Natural Learning Systems: Volume IV: Making Learning Systems Practical*, MIT Press, 1996. {`missing.ps`}

    An inductive inference system uses labeled training examples to learn a classification function for labeling subsequent unlabeled performance examples. Most theoretical analyses assume that both training and performance examples are *complete*, in that the value of every attribute is known to both learner and classifier. Real-world data, however, is usually incomplete. This paper addresses this discrepancy by formally analyzing the task of learning to classify incompletely specified performance examples, based on completely- and incompletely-specified training examples, respectively. This formalism requires an extension of the classical notion of concept definition, called "default concept definition" (dcd), whose classification behavior can be *nonmonotonic*. We first present a formal account of these dcds and show that they are similar, but not identical, to important existing ideas from both learnability and AI knowledge representation formalisms. We next define a generalization of Valiant's probabilistic model of instance presentation that allows attribute values to be hidden from the classifier. Finally, we use this model to develop an extension to the PAC learning framework that can learn dcds from examples, and prove a number of learnability results, both positive and negative, which collectively suggest the appropriate ways of treating missing attribute values.

17. R. Greiner and D. Schuurmans: "Learning an Optimally Accurate Representational System", *ECAI Workshop on Foundations of Knowledge Representation and Reasoning*, Springer Verlag (LNAI series), 1993. {`accuracy-ecai.ps`}
[Expanded from workshop preceedings, Vienna, August 1992; {`accurate.ps`}.]

    A default theory can sanction different, mutually incompatible, answers to certain queries. We can identify each such theory with a set of related credulous theories, each of which produces but a single response to each query, by imposing a total ordering on the defaults. Our goal is to identify the credulous theory with optimal "expected accuracy" averaged over the natural distribution of queries in the domain. There are two obvious complications: First, the expected accuracy of a theory depends on the query distribution, which is usually not known. Second, the task of identifying the optimal theory, even given that distribution information, is intractable. This paper presents a method, OptAcc, that side-steps these problems by using a set of samples to estimate the unknown distribution, and by hill-climbing to a local optimum. In particular, given any error and confidence parameters $\epsilon, \delta > 0$, OptAcc produces a theory whose expected accuracy is, with probability at least $1 - \delta$, within ϵ of a local optimum.

18. R. Greiner, "Adaptive Derivation Processes", *Workshop on Knowledge Compilation and Speedup Learning*, Amherst, Massachusetts, June 1993. {`adaptive.ps`}

    Many reasoning systems must reach conclusions based on stored information; we can often model this as deriving logical conclusions from a given knowledge base of facts. We of course prefer derivation systems that draw all and only the correct conclusions, and that reach these conclusions as quickly as possible. Unfortunately, a sound and complete derivation process can be intractable, if not undecidable, in the worse case. This position paper discusses the general challenge of producing a derivation process that is as effective as possible, and argues for using a (cautious) adaptive derivation process here.

19. C. Elkan and R. Greiner, "Book Review: Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project", *Artificial Intelligence*, 61:1 (1993) 41–52. {`cyc.ps`}

    The book under review here, "Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project", describes progress so far in an attempt to build a system that is intended to exhibit general common-sense reasoning ability. This review first discusses aspects of the Cyc system, with a focus on important decisions made in designing its knowledge representation language, and on how claims about the performance of the system might be validated. The review then turns to the book itself, discussing both its merits and its faults.

20. R. Greiner and D. Schuurmans, "Learning Useful Horn Approximations", *Third International Conference on Knowledge Representation and Reasoning (KR-92)*, Boston, October 1992. {`horn.ps`}

    (Expanded from
    R. Greiner and D. Schuurmans, "Learning Good Horn Approximations", AAAI Spring Symposium on Knowledge Assimilation*, Stanford, March 1992.)*

    While the task of answering queries from an arbitrary propositional theory is intractable in general, it can typically be performed efficiently if the theory is Horn. This suggests that it may be more efficient to answer queries using a "Horn approximation"; *i.e.*, a horn theory that is semantically similar to the original theory. The utility of any such approximation depends on how often it produces answers to the queries that the system actually encounters; we therefore seek an approximation whose expected "coverage" is maximal. Unfortunately, there are several obstacles to achieving this goal in practice: *(i)* The optimal approximation depends on the query distribution, which is typically not known *a priori*; *(ii)* identifying the optimal approximation is intractable, even given the query distribution; and *(iii)* the optimal approximation might be too large to guarantee tractable inference. This paper presents an approach that overcomes (or side-steps) each of these obstacles. We define a learning process, AdComp, that uses observed queries to estimate the query distribution "online", and then uses these estimates to hill-climb, efficiently, in the space of size-bounded Horn approximations, until reaching one that is, with provably high probability, effectively at a local optimum.

21. R. Greiner and I. Jurisca, "A Statistical Approach to Solving the EBL Utility Problem", *Tenth Annual National Conference on Artificial Intelligence (AAAI-92)*, San Jose, July 1992. {`utility.ps`}

    Many "learning from experience" systems use information extracted from problem solving experiences to modify a performance element PE, forming a new element PE′ that can solve these and similar problems more efficiently. However, as transformations that improve performance on one set of problems can degrade performance on other sets, the new PE′ is not always better than the original PE; this depends on the distribution of problems. We therefore seek the performance element whose *expected performance*, over this distribution, is optimal. Unfortunately, the actual distribution, which is needed to determine which element is optimal, is usually not known. Moreover, the task of finding the optimal element, even knowing the distribution, is intractable for most interesting spaces of elements. This paper presents a method, PALO, that side-steps these problems by using a set of samples to estimate the unknown distribution, and by using a set of transformations to hill-climb to a local optimum. This process is based on a mathematically rigorous form of *utility analysis*: in particular, it uses statistical techniques to determine whether the result of a proposed transformation will be better than the original system. We also present an efficient way of implementing this learning system in the context of a general class of performance elements; and include empirical evidence that this approach can work effectively.

22. R. Greiner, "Learning Efficient Query Processing Strategies", *Eleventh Symposium on Principles of Database Systems (PODS-92)*, San Diego, June 1992. {`qp.ps`}

    A query processor QP uses the rules in a rule base to reduce a given query to a series of attempted retrievals from a database of facts. The QP's *expected cost* is the average time it requires to find an answer, averaged over its anticipated set of queries. This cost depends on the QP's *strategy*, which specifies the order in which it considers the possible rules and retrievals. This paper provides two related learning algorithms, PIB and PAO, for improving the QP's strategy, *i.e.*, for producing new strategies with lower expected costs. Each algorithm first monitors the QP's operations over a set of queries, observing how often each path of rules leads to a sufficient set of successful retrievals, and then uses these statistics to suggest a new strategy. PIB hill-climbs to strategies that are, with high probability, successively better; and PAO produces a new strategy that probably is approximately optimal. We describe how to implement both learning systems unobtrusively, discuss their inherent time and space complexities, and use methods from mathematical statistics to prove their correctness. We also discuss additional applications of these approaches to several other database tasks.

23. V. Chaudhri and R. Greiner, "A Formal Analysis of Solution Caching", *Ninth Canadian Conference on Artificial Intelligence (CSCSI-92)*, Vancouver, May 1992. {`cache.ps`}

    Many inference management systems store and maintain the conclusions found during a derivation process in a form that allows these conclusions to be used during subsequent derivations. As this approach, called "solution caching", allows the system to avoid repeating these derivations, it can reduce the system's overall cost for answering queries. Unfortunately, as there is a cost for storing these conclusions, it is not always desirable to cache every solution found — this depends on whether the savings achieved by performing fewer inference steps for these future queries exceeds the storage overhead incurred. This paper formally characterizes this tradeoff and presents an efficient algorithm, FOCL, that produces an optimal caching strategy: *i.e.*, given an inference graph of a knowledge base, anticipated frequencies of queries and updates of each node in this graph, and various implementation-dependent cost parameters, FOCL determines which of these nodes should cache their solutions to produce a system whose overall cost is minimal. The paper also presents empirical results that indicate that a system based on FOCL can significantly outperform one based on any of the standard approaches to solution caching.

24. R. Greiner and C. Elkan, "Measuring and Improving the Effectiveness of Representations". *Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, August 1991. {`ijcai-repn.ps`}

    This report discusses what it means to claim that a representation is an effective encoding of knowledge. We first present dimensions of merit for evaluating representations, based on the view that usefulness is an external property, and is necessarily relative to a specified task. We then provide methods (based on results of mathematical statistics) for reliably measuring effectiveness empirically, and hence for comparing different representations. We also discuss weak but guaranteed methods of improving inadequate representations. Our results are an application of the ideas of formal learning theory to concrete knowledge representation formalisms.

25. R. Greiner and P. Orponen, "Probably Approximately Optimal Derivation Strategies", *Second International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, April 1991. {`pao_kr91.ps`}

    An inference graph can have many "derivation strategies", each a particular ordering of the steps involved in reducing a given query to a sequence of database retrievals. An "optimal strategy" for a given class of queries is a complete strategy whose "expected cost" is minimal, where the expected cost depends on the conditional probabilities that each requested retrieval succeeds, given that a member of this class of queries is posed. This paper describes how to use a set of training examples to approximate these probability values, towards producing a "probably approximately optimal" (pao) strategy — that is, given any $\epsilon, \delta > 0$, a strategy whose cost is within ϵ of the cost of the optimal strategy, with confidence greater than $1 - \delta$. It also shows how to obtain these pao strategies in time polynomial in $1/\epsilon$, $1/\delta$ and the size of the inference graph, for many important classes of graphs, including all and-or trees.

26. P.E. Caines, R. Greiner and S. Wang: "Classical and Logic-Based Dynamic Observers", *Institute of Mathematics and its Applications: Journal on Control and Information*, 8 (45-80), 1991.

    This report formulates the state estimation problem for a partially observed input-state-output automaton in terms of an "observer automaton" whose nodes each correspond to the *set* of states consistent with a particular sequence of observations. It provides several complexity results about these observers — bounding their convergence time and size. The final part shows how to encode these observers in predicate calculus, and illustrates some of the advantages of this approach.

27. R. Greiner, "Finding Optimal Derivation Strategies in Redundant Knowledge Bases", *Artificial Intelligence*, 50:1 (1991) 95–115. {`redund.ps`}

    A backward chaining process uses a collection of rules to reduce a given goal to a sequence of data-base retrievals. A "derivation strategy" is an ordering on these steps, specifying when to use each rule and when to perform each retrieval. Given the costs of reductions and retrievals, and the *a priori* likelihood that each particular retrieval will succeed, one can compute the expected cost of any strategy, for answering a specific query from a given knowledge base. Smith [1] presents an algorithm that finds the minimal cost strategy in time (essentially) linear in the number of rules, for any disjunctive, irredundant knowledge base. This paper proves that the addition of redundancies renders this task NP-hard. Many Explanation-Based Learning systems work by adding in redundancies; this shows the complexities inherent in their task.

28. Y. Xiao and R. Greiner, "A Distributed Plan Verifier", *Third UNB Artificial Intelligence Workshop*, Oct 1990. {`distrib.ps`}

    This paper describes the architecture of an efficient plan verifier that can first detect faults in a planner's plans and use these observed errors to identify possible problems in the planner's knowledge base and suggest appropriate corrections.

Older Journal Articles

29. R. Greiner, "Against the Unjustified Use of Probabilities: A Critique of Cheeseman's 'An Inquiry into Computer Understanding' ", *Computational Intelligence: An International Journal*, IV:1 (79-83), February 1988.

    This review has two independent parts. Part 1 addresses Sections 2 and 3 in Cheeseman's *An Inquiry into Computer Understanding* paper (AIiCU), demonstrating that logic is capable of many tasks which AIiCU claims require the use of probability. Part 2 addresses AIiCU's Section 4, asking specific questions about "Bayesian Inductive Inference".

30. R. Greiner, "Learning by Understanding Analogies", *Artificial Intelligence*, 35:1 (81–125), May 1988.

    Analogical inference is a process that proposes new conjectures about a target analogue based on facts known about a source analogue. This article formally defines this process and discusses how to efficiently guide it to the conjectures that can help to solve a given problem. The intuition that a useful analogy provides the information needed to solve the problem, and no more, leads to two sets of heuristics: one set based on abstractions — abstract relations which encode solutions to previous problems — and the second, based on a preference for the most general set of new conjectures. Experimental data, collected using a program which embodies this theory of analogy, confirms the effectiveness of these ideas.

31. R. Greiner, B. A. Smith and R. W. Wilkerson, "A Correction to the Algorithm in Reiter's Theory of Diagnosis", *Artificial Intelligence*, 41:1 (79–88), November 1989. {`correct_tr.ps`}

    (Reprinted in
    "Readings in Model-based Diagnosis", edited by W. Hamscher, J. de Kleer and L. Console,
    Morgan Kaufmann, 1992.)

    Reiter [1987] has developed a general theory of diagnosis based on first principles. His algorithm computes all diagnoses which explain the differences between the predicted and observed behavior of a given system. Unfortunately, Reiter's description of the algorithm is incorrect in that some diagnoses can be missed under certain conditions. This note presents a revised algorithm and a proof of its correctness.

32. R. Lee, E. Milios, R. Greiner, J. Rossiter, and A. Venetsanopoulos: "On the machine analysis of radar signals for ice profiling", *Journal of Signal Processing*, Elsevier Science Publishers, 18 (1989) 371–386.

    We describe the design and implementation of an automated system for interpreting impulse radar signals for ice thickness profiling. We have adopted an integrated approach which includes numeric computation in the form of deconvolution filtering with rule-based classification of signal features at multiple levels. Noise reduction and deconvolution techniques are used to enhance the radar signals for better resolution of overlapping events. Motivated by human perceptual (visual) knowledge, a hierarchy of data structures is constructed as representations of signal characteristics at various levels of abstraction. Classification rules, based on the protocols collected from an expert, physical constraints on the helicopter motion and the nature of the radar signals are used to produce the current signal interpretation. A prototype system has been implemented on the Symbolics Lisp Machine and tested on real data from airborne impulse radar sounding of freshwater ice.