In: Machine Learning, Meta-reasoning and Logics, pp207-232,
Eds: P. B. Brazdil and K. Konolige, (1990), Boston: Kluwer.
http://www.cs.utexas.edu/users/pclark/papers

Learning from Imperfect Data

Pavel Brazdil*
Peter Clark†

1990

Abstract

Systems interacting with real-world data must address the issues raised by the possible presence of errors in the observations it makes. In this paper we first present a framework for discussing imperfect data and the resulting problems it may cause. We distinguish between two categories of errors in data – random errors or 'noise', and systematic errors – and examine their relationship to the task of describing observations in a way which is also useful for helping in future problem-solving and learning tasks. Secondly we proceed to examine some of the techniques currently used in AI research for recognising such errors.

1 Introduction

Systems interacting with real-world data must address the issues raise by the possible presence of errors in the observations it makes. Systems rarely make observations of important features directly – instead they are observed indirectly through the medium of measuring and transmission equipment, which may give rise to errors in the observation. Such errors may cause problems in the system's attempt to describe its observations in a useful way. Without addressing the issue of accounting for such imperfections in data, a system may form an unnecessarily complex theory about its observations due to its inability to tolerate a degree of approximation in its theory, or its inability to take into account imperfections in the source from which data was obtained. In this paper we present a framework for discussing imperfect data and the problems it presents, and summarise some of the techniques currently used in AI research for addressing these problems.

One source of imperfection in data is that of *random errors* or 'noise'. A dictionary definition of noise gives noise as "meaningless components of data". Similarly, the related concept randomness is defined as "that without assignable cause". Implicit in both of these definitions is the notion of modelling and predictability; from these definitions components of noise in data can be viewed as those elements for which there is no perceivable causality, in other words for which there is no model and hence are apparently random.

This immediately raises issues for work in artificial intelligence. Any system performing non-repetitive tasks has to make choices between alternatives during problem-solving. To make good choices, the system needs to be able to predict and assess the likely outcomes of its alternative choices – and in order to predict such outcomes some form of

*Current address: Faculdade de Economia, Univ. of Porto, 4200 Porto, Portugal
†Current address: The Turing Institute, 36 N.Hanover St, Glasgow, UK

model of the world is required. Learning systems should correct and refine their internal models in the light of failure, when predictions and observations disagree. But what if there is noise in the observable world? A learning system may be wasting resources and introducing unnecessary complexity to its model in an attempt to model unmodellable elements of observation. Such undesirable effects on systems failing to deal adequately with noise are discussed later, but first a more detailed examination at sources of noise is presented.

In addition to random corruption of data, we also consider the introduction of *systematic errors* into observation. Such errors may be caused, for example, by incorrect calibration of an item of measuring equipment so that it is reading consistently low. Again, the ability of a system to identify such errors is partially dependent on its ability to describe its observations, and requires the ability to model the source of information (eg. the piece of measuring equipment) as well as the object which the information describes.

The principle problem we examine in this paper is that of distinguishing between errors (both random and systematic) in data and errors in the theory the system has formed to explain its observations. Firstly, we examine those phenomena which give rise to imperfect data and also examine those characteristics of the learning system itself which may be responsible for errors in its theory of the world. Then, we proceed to review some of the techniques currently used in AI research to distinguish between these two sources of error in a system's theory of the world.

2 Origins of Error in a System's Model

Learning systems learn by observing phenomena of the world within which they operate. Sometimes they can also perform actions in that world. From observations, such systems attempt to form some model explaining what has been observed and hence be able to predict the effects of future actions.

Forming an adequate description of the world is prone to error, and features of both the world and the learning system itself can be responsible for such errors. If the errors are introduced from the world (in the form of errors in the data the system observes) we refer to them as **external sources** of error. External errors can be caused by both **random** or **systematic** imperfections in the data (eg. due to voltage fluctuations in a transmission line, or due to incorrect instrument calibration). However, errors can also occur in the system's internal description of the world due to properties of the learning system itself (eg. poor search heuristics). We refer to these sources of error as **internal sources**. We discuss these two sources below.

2.1 External Sources of Error

2.1.1 Random Data Errors

Often, data contains random or 'noisy' components in it, caused by the inherent unpredictability of some events in nature. As a consequence, variations in elements of data may be observed which cannot be accounted for. Such random fluctuations may be inherent in the world being observed (eg. due to gusts of wind in a robot laboratory, imprecision in a robot arm) or in transmission of observations to the learning system (eg. imperfect measuring equipment, transcription errors).

Manago and Kodratoff [1] present a more detailed analysis of the sources of noise in data, in particular for the case where the 'measuring device' used to relay information about the concept to the system is a human. They subdivide the category of 'noisy components' of data into 'unreliable' (originating from the concept itself) and 'wrong' (originating from human-introduced mistakes). In addition, they extend the definition of noise to include a third category 'incomplete information' – here, however, we have used the term 'noise' in the slightly narrower sense of meaning 'random data errors' only, and included other categories of imperfect data such as incomplete information in category 2.1.3 below.

2.1.2 Systematic Data Errors

Sometimes, errors are introduced in a predictable rather than a random way – for example if an instrument is poorly calibrated. Unlike random errors, systematic errors can be accounted for by including some theory of the sources of the information, as well as a theory of that which the information is about.

2.1.3 Other External Sources

In addition to imperfect data, there are several other factors which may contribute towards errors in a system's theory of the world. We mention these briefly here:

- **Limited Description Language:** Events perceived by the system are described using statements in some description language. If some features of these events are unexpressable in this language, the features will not be observed by the system. Consequently, some processes may appear unexplainable to the system because the factors responsible for the observed variations cannot be expressed.

- **Incomplete Description:** An incomplete description can arise if some descriptive features have unknown values (eg. results of a particular medical test may be missing). Again, this can constrain the system's ability to form an adequate theory of its observations.

- **Limited Amount of Data:** If data is sparse, it may be difficult to reliably detect correlations in observations. This is a common problem in classification systems, where strong correlations between attributes and classes are not found simply because not enough data has been observed.

- **Intractable Data:** At the other extreme, if too much data is available to the system the problem of detecting correlations may be difficult or intractable. In other words, the problem of detecting relevant correlations may be exacerbated by the existence of data which is irrelevant to the particular learning task. The severity of this problem is related to the system's internal property of available computational power.

2.2 Internal Sources of Error

Most interestingly, there are several internal or 'subjective' sources of error in a system's ability to model the world it operates in. By subjective, we mean some observing systems will fail to perceive certain regularities in the environment, while other systems will succeed in perceiving them. Consequently, some systems may consider certain phenomena unexplainable whereas other systems may be able to form a theory to explain them. That some phenomenon is unexplainable, or best explainable as imperfections in the information source itself, is dependent on the system's ability to model correctly the underlying causes of events. If the system's model is deficient in some sense, the system will occasionally fail to predict correctly the outcome of some events.

Perception of unexplainability – in other words the absence of perceivable correlations – is dependent on the system's ability to model external events. Limits on the system's modelling power will constrain how much of the data it can account for and how much remains unexplained. Such limits can be affected not only by a system's limited conceptual apparatus, but also by limited computational power. The system may simply be unable to perceive certain correlations not because the conceptual apparatus is lacking certain concepts, but instead because there is insufficient computational power to locate an adequate theory. Computational power constrains the system in two ways in this respect – firstly in the task of locating a theory and secondly in the task of using the theory for prediction. In the latter case, if there is insufficient power to apply the theory the theory is said to be *intractable* [2] [3].

3 Problems Created by Imperfect Data

Although we have indicated different sources of error in a system's theory, both internal and external, it should be noted that there is a degree of overlap between these categories. However from the learning system's point of view the problem is the same – namely that there may be components of data which are apparently unexplainable, and the system needs to be able to distinguish them from other components. We shall now proceed to look at what these problems are and how they problems can be overcome.

3.1 The Perfect Data Assumption

The objective of many learning systems is to construct an internal model of the world which is completely consistent with observations (eg. Lex [4], ID3 [5]). Should a new observation be made which is inconsistent with the world, this acts as a cue for the system to alter its model. We shall call such systems **perfect modelling systems** obeying a **consistency requirement**. On first sight this requirement seems reasonable. However, this requirement is based (in part) on the assumption that the data the system has available is error-free – a 'perfect data' assumption.

In many applications this assumption is not valid. If certain elements in the data are in fact random it is impossible for the system to construct a completely predictive model. If we let the learning system continue though in the perfect modelling mode, the system will nevertheless try to form such a model. As a consequence, unnecessary components will be generated which, besides making no contribution to the predictive power of the system, will at the same time impose on it an extra computational burden. This situation is generally referred to as **overfitting**.

Instead, if the data is not perfect, the consistency requirement should be relaxed. This however raises the question of how we can detect that the data has been corrupted and how we can relax the consistency constraint. This point will be discussed later where we will describe some of the existing techniques for dealing with this problem.

3.2 Trade-offs between Simplicity and Consistency

Previous discussion seems to suggest that whenever the perfect data assumption holds, the consistency requirement should always be adhered to. Although the consistency requirement is certainly important, there are other criteria which may be important: simplicity of the overall model, adaptability, response time etc. It may happen that a slightly inconsistent and faster model is preferable to a model which is completely consistent but very slow. In order to take such criteria into account, the system should consider the overall goal of learning in terms of several metrics (eg. consistency, simplicity, adaptability) when forming its theory of the world.

4 Techniques for Dealing with Imperfect Data

We have classified errors in data into two kinds, random and systematic. We have also argued that imperfect data is not the only cause of a system holding an incorrect theory of the world, and described other external and internal causes. As it is undesirable for a system's theory to be incorrect, the central question we now address is how to distinguish between imperfect data and other causes of error in theory, and we review methods currently used in AI for making this distinction.

Techniques for dealing with random errors (noise) and for dealing with systematic errors differ. With noise, the issue is that of discerning whether genuine correlations exist in the data. With systematic errors, the issue is of discerning whether perceived correlations are due to the phenomenon of interest or due to properties of the information source from which information about the phenomenon is obtained. As a consequence, we deal with these two topics separately in the following section.

4.1 Techniques for Dealing with Random Errors

Systems working with real-world problems frequently have to address the issue of dealing with noise in the data, and there are essentially two categories of techniques:

- Transformation of the data, so that a 'perfect modelling' algorithm can then be applied
- Modify the learning algorithm

Each of these categories will be discussed in more detail below.

4.1.1 Transformation of Data

One method of dealing with noise is to try to remove it such that an algorithm making the 'noiseless domain' assumption can then be applied. Examples of this are :

- **Data filtering**
  Data is pre-filtered to extract only the most representative examples for feeding to the learning algorithm. For example the ESEL subsystem for the AQ11 induction program filters training data in this way [6].
- **Using "coarse-grained" attributes and classes**
  In many cases, it is not necessary to predict the world perfectly, but only to a certain accuracy. By using a larger grain size for attributes and classes, effects

of noise can be hidden from the learning system. For example, voltage could be considered as being ‘high’ or ‘low’, rather than taking a numerical value, resulting in small, noisy variations of voltage being invisible to the learning system. Then a learning algorithm making the ‘noiseless domain’ assumption can be effectively applied. Many qualitative modelling systems can hide noisy components in this way (another example of this is Kardio [7] where coarse grained attributes such as ‘heart rate’ hide part of the noise in data).

It should be noted that the task of selecting the appropriate grain size involves a trade-off, as coarse-grained attributes do not only hide the effects of noise but also decrease the precision of the model. Finding the best trade-off is potentially a difficult task for a learning system.

4.1.2 Modifying the Learning Algorithm

A second category of techniques for learning in the presence of noisy data is that of modifying the learning algorithm. Under the ‘perfect domain’ assumption, only models completely consistent with data need be explored. However in noisy domains, models slightly inconsistent with data also need to be explored and the selection criteria for ‘best model’ changed. Typically an additional measure of ‘reliability’ of the evidence supporting a model is introduced, and if this measure drops below a user-supplied threshold the model is considered inadequate and discarded. This process of threshold selection can be automated by the use of cross-validation techniques such as described by Brieman et al. [8] and Watkins [9].

Typically, systems assess reliability using statistically-based measures of generality and accuracy of model components. This can be viewed as the application of a weak kind of meta-knowledge to assess the quality of the knowledge itself. The extension of this is to employ more sophisticated meta-knowledge to assess reliability; for example Occam [10] uses additional meta-knowledge about which other hypotheses a particular hypothesis supports, whether there exist alternative competing hypotheses to the currently believed on etc. to assess confidence in a model component.

- **Pruned general-to-specific[1] search**
  One of the most common techniques for filtering out noisy components of data is to perform the search for a generalized description of observations in a general-to-specific manner, and then to halt the search before complete consistency with observation is reached. Such halting occurs when the number of examples supporting the current hypothesis falls too low. The general-to-specific search acts as a ‘covers lots of examples’-to-‘covers few examples’ search, which can be seen as a reliable-to-unreliable search that can be prematurely halted should a level of unreliability be passed.

  Several rule induction systems do this, the measure of unreliability being computed from the number and distribution of examples covered by the hypothesis in various ways eg. using percentage tests such as by King [11], Sequoia [12] and Plage [13]

[1] The word ‘general’ is applied of course to *components* of the model (eg. a production rule, a decision tree branch) rather than to the model as a whole - the latter’s generality (usually covering the entire space of examples) typically remains unchanged during search if generality is defined as the proportion of example space covered.

or using chi squared tests such as in CN2[2] [14]. The ID3 algorithm, performing a general-to-specific search already, can be similarly modified to cope with noise by halting branch growth earlier, as illustrated for example by Assistant-86 [15].

- **Rule truncation, or post-pruning**
  An alternative to prematurely halting a general-to-specific search, as above, is to allow it to run to completion (ie. form a model completely consistent[3] with observation) then to 'post-prune' the final model to remove components deemed unreliable. The advantage of this is that the quality of pruned and unpruned versions of a hypothesis can be directly compared rather than having to estimate the latter's quality during search.

  Again in rule induction systems this technique is common, for example in the most recent ID3 descendants C4 [16], Assistant-86 [15] and by Niblett and Bratko [17]. Niblett [18] gives a review of pre- and post-pruning techniques used for decision trees. AQ15 [19] employs a post-pruning technique for production rules (termed 'rule truncation'). The use of post-pruning of decision tree branches to generate production rules has been used by Corlett [20] and Quinlan [21].

- **Corroborative application of model components**

  Another technique to prevent unreliable model components degrading performance is to allow *all* components to contribute in some way during problem-solving, with different weights attached to their decisions. This avoids heavy reliance on a specific, possibly unreliable, part of the system, allowing noisy effects to be over-ridden and smoothed out by other model components. Statistical methods such as Bayesian techniques (eg. [22]) can be employed. AQ15 [19] performs weighted rule application in this way, and Quinlan [23] suggests how decision tree application can be made less brittle by introducing a degree of corroboration between decision tree branches.

- **The "exception-based" paradigm**

  For any system learning incrementally, noise presents the problem of when to ignore observations conflicting theory and when to modify the theory. Such systems are required to be robust (not changing with every new observation), but not so robust as to never change.

  A common technique for this is as follows. Should a new observation contradict expectations, then don't refine the model but instead note the contradiction (or 'expectation failure') in memory. Should the number of examples of exceptions to a model component rise above a certain threshold, then make a 'paradigm shift' and alter the model to account for as many exceptions as possible. This threshold is typically a certain number of failures (eg. 2), and has the effect requiring a certain weight of evidence before change can be justified. Examples of systems using this approach are Unimem [24], Alfred [25] and by Emde [26].

4.2 Techniques for Dealing with Systematic Errors

The task of dealing with systematic errors is related to that of dealing with random errors, in that it concerns the problem of finding an adequate model of observations. However,

[2] This system is an example of a modified search algorithm as well as modified search heuristics, where the basic AQ algorithm has been extended to search slightly inconsistent as well as consistent hypotheses

[3] Or as nearly consistent as possible if complete consistency is impossible.

unlike random errors, the particular problem is not of differentiating between explainable and unexplainable components of data but instead of distinguishing between explanations related to the phenomena of interest and explanations related to the source of knowledge itself through which information about the phenomena is being obtained. Such knowledge sources may vary in complexity, from a simple measuring tool to a complex system such as another active agent in the world (see [27] for a discussion of the latter).

The detection of systematic errors is partly dependent on a system's ability to model the source from which data is obtained. In situations where sources of knowledge are simple this can be straightforward, for example the existence of calibration errors can be hypothesised easily by allowing the system to consider the possibility of a linear offset in measurements it observes. However, where the knowledge sources are complex, such as is the case if the source is another active agent in the world, the theory of the knowledge source may be complex. For example Buggy, the tutoring system for correcting simple arithmetic problems [28], included the capacity to model the pupil's reasoning and hence explain the data it observes (namely the pupil's possibly incorrect solutions to problems) not only as random errors but also as complex systematic errors.

Buggy and other tutoring systems incorporating a model of the pupil simplify the problem of error detection because if the pupil gives an answer different to that expected, they assume that the pupil was always in error. It never considers the possibility that its own theory of, say, arithmetic is incorrect. However, in the general case where the system is not only a tutor but also required to learn, no such assumption can be made.

The problem of distinguishing between a systematic error in the source of information and an error in the learning system's own theory is difficult, and constitutes part of the credit assignment task in the 'incorrect theory' problem. Techniques similar to those for dealing with noise are required, namely the ability to assess the degree of belief in a particular theory and in observed evidence, and the ability to assess the quality of alternative 'patches' which can be used to correct the theory. This general problem of debugging incomplete and inadequate domain theories has been recently addressed in research in explanation-based learning (eg. by Hall [29], VanLehn [30] and in Disciple [31]), although there is still much research needed in addressing this difficult task.

5 Conclusion

Systems operating in the real world must address the problems presented by the possibility of errors in observations. In this paper we have described a framework with which to characterise such errors, and used this as a basis to discuss the problems caused by data imperfections. in data.

AI systems aim to form a model of the world they observe, to act as the basis for future decisions in problem-solving tasks. There are several sources which may lead to an error in such a model – some external, (including the presence of errors in the data), and some internal, due to the limitations of the system itself. We have examined one particular source of model error, namely imperfect data, in detail and distinguished between two types or errors in data, namely random errors (noise) and systematic errors. Most importantly, we have emphasised that the perception of both random and systematic errors depends on a system's ability to model the world it observes, hence introducing a subjective component to the decision of what does and does not constitute imperfection in data. Finally, we have explored some techniques developed in AI research for addressing these problems, allowing the detection and hence either the removal or modelling of such imperfections in observation.

6 Acknowledgements

This article originated from a panel discussion session at the second European Working Session on Learning (EWSL) entitled "Learning from Noisy Data and an Imperfect Teacher" (1987). We would like to thank Michel Manago, Joel Quinqueton and Stefan Wrobel for their contributions towards the ideas in this paper from the panel discussion and in subsequent correspondence, and Yves Kodratoff for comments on an earlier draft of this paper.

References

[1] Michel M. Manago and Yves Kodratoff. Noise and knowledge acquisition. In J. McDermott, editor, *IJCAI-87*, pages 348–354, Kaufmann, Ca, 1987.

[2] T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli. Explanation-based generalization: a unifying view. *Machine Learning Journal*, 1(1):47–80, 1986.

[3] P. V. Tadepalli. Learning in intractable domains. In T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, editors, *Machine Learning: A Guide to Current Research*, Kluwer, Lancaster, UK, 1986.

[4] T. M. Mitchell, P. Utgoff, and R. Banerji. Learning by experimentation: acquiring and refining problem-solving heuristics. In J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, editors, *Machine Learning, vol. 1*, Tioga, Palo Alto, Ca, 1983.

[5] J. Ross Quinlan. Learning efficient classification procedures and their application to chess endgames. In J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, editors, *Machine Learning, vol. 1*, Tioga, Palo Alto, Ca, 1983.

[6] R. S. Michalski and J. Larson. *Selection of most representative training examples and incremental generation of VL_1 hypotheses: the underlying methodology and the description of programs ESEL and AQ11.* Technical Report UIUCDCS-R-78-867, The University of of Illinois at Urbana-Champaign, Department of Computer Science, Urbana, 1978.

[7] Ivan Bratko, Igor Mozetic, and Nada Lavrač. Automatic synthesis and compression of cardiological knowledge. In *Machine Intelligence 11*, Ellis Horwood, Chichester, UK, 1986. (to be published).

[8] L. Brieman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.

[9] C. J. C. H. Watkins. Combining cross-validation and search. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, pages 79–87, Sigma, Wilmslow, UK, 1987.

[10] M. Pazzani, M. Dyer, and M. Flowers. The role of prior causal theories in generalization. In *AAAI-86*, 1986.

[11] Ross King. An inductive learning approach to the problem of predicting a protein's secondary structure from its amino acid sequence. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, Sigma, Wilmslow, UK, 1987.

[12] J. Haiech, J. Quinqueton, and J. Sallantin. Concept formation from sequential data. In *Proceedings of the 1st European Working Session on Learning (EWSL-86)*, Université de Paris-Sud, Orsay, France, 1986.

[13] O. Gascuel. Plage: a way to give and use knowledge in learning. In *Proceedings of the 1st European Working Session on Learning (EWSL-86)*, Université de Paris-Sud, Orsay, France, 1986.

[14] Peter Clark and Tim Niblett. Induction in noisy domains. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, Sigma, Wilmslow, UK, 1987.

[15] Bojan Cestnik, Igor Kononenko, and Ivan Bratko. Assistant 86: a knowledge-elicitation tool for sophisticated users. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, pages 31–45, Sigma, Wilmslow, UK, 1987.

[16] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus. Inductive knowledge acquisition: a case study. In *Proceedings of the second Australian Conference on the Applications of Expert Systems*, pages 183–204, New South Wales Institute of Technology, Sydney, 1986.

[17] Tim Niblett and Ivan Bratko. Learning decision rules in noisy domains. In *Expert Systems 86, Brighton, UK*, 1986.

[18] Tim Niblett. Constructing decision trees in noisy domains. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, pages 67–78, Sigma, Wilmslow, UK, 1987.

[19] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The AQ15 inductive learning system : an overview and experiments. In *Proceedings of IMAL 1986*, Université de Paris-Sud, Orsay, France, 1986.

[20] R. A. Corlett. Explaining induced decision trees. In *Expert Systems 83*, pages 136–142, 1983.

[21] J. Ross Quinlan. Generating production rules from decision trees. In J. McDermott, editor, *IJCAI-87*, pages 304–307, Kaufmann, Ca, 1987.

[22] D. Michie. Bayes, turing and the logic of corroboration. In *Machine intelligence and related topics: an information scientist's weekend book*, Gordon & Breach, NY, 1982.

[23] J. R. Quinlan. Decision trees as probabilistic classifiers. In P. Langley, editor, *Proc. 4th International Workshop on Machine Learning*, Kaufmann, Ca, 1987.

[24] M. Lebowitz. Concept learning in a rich input domain : generalization-based memory. In J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, editors, *Machine Learning, vol. 2*, Tioga, Palo Alto, Ca, 1986.

[25] C. Reisbeck. Failure-driven reminding for incremental learning. In *IJCAI-81*, 1981.

[26] W. Emde. Big flood in the blocks world; or non-cumulative learning. In *ECAI-86*, pages 569–575, 1986.

[27] Pavel Brazdil. Knowledge states and meta-knowledge maintenance. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (proceedings of the 2nd European Working Session on Learning)*, Sigma, Wilmslow, UK, 1987.

[28] Richard Burton. Diagnosing bugs in a simple procedural skill. In D. Sleeman and J. S. Brown, editors, *Intelligent Tutoring Systems*, pages 157–183, Academic Press, 1982.

[29] Robert Hall. *Learning by Failing to Explain.* Technical Report 906 (AI-TR-906), MIT AI Laboratory, 1986.

[30] K. VanLehn. Learning a domain theory by completing explanations. In T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, editors, *Machine Learning: A Guide to Current Research*, Kluwer, Lancaster, UK, 1986.

[31] Yves Kodratoff and Gheorghe Tecuci. What is an explanation in disciple? In P. Langley, editor, *Proc. 4th International Workshop on Machine Learning*, Kaufmann, Ca, 1987.