domains. CTRL.layout.3 and CTRL.layout.4 form a separate version derivation path because they are completely independent of the path formed by CTRL.layout.1 and CTRL.layout.2. Other relationships can be similarly deduced.

In summary, when an object is created, the system consults with the creating tool's TSD to establish the associated configuration relationship, and a backward traversal of the augmented derivation graph starting from the triggering object is initiated. During the traversal, associated version and equivalence relationships are created by checking each visited node. Finally, the equivalence relationships that could be inherited from the triggering object's immediate ancestor, if any, are located and established.

III. Conclusion

We apply the *immediate computation* paradigm as advocated by language-directed editor researchers [4] to engineering object management systems by treating each tool execution as an editor command, and the whole object base as an interrelated network of objects. Based on this idea we are able to establish interobject relationships incrementally and automatically. Three kinds of relationships supported by the Version Server are examined in greater detail. The semantics of design operations, captured by *tool semantic specifications*, are used to determine whether equivalence relationships can be inherited from the ancestor by the descendent version. The power of this paradigm scales with the amount of tool/object semantics that can be specified in advance. As more refined semantic specifications of tools and/or objects are made known to the system, we believe the proposed approach could build up more varieties of sophisticated meta-data in an automatic fashion.

References

[1] E. Chang, D. Gedye, and R. Katz, "The design and implementation of a version server for computer-aided design data," *Software—Practice and Experience*, vol. 19, no. 3, pp. 199-222, Mar. 1989.

[2] G. Clemm, D. Heimbigner, L. Osterweil, and L. Williams, "Keystone: A federated software environment," in *Proc. Workshop Software Eng. Environments for Programming-in-the-Large*, June 1985, pp. 80-89.

[3] S. E. Hudson and R. King, "Cactis: A self-adaptive, concurrent implementation of an object-oriented database management system," Dep. Comput. Sci., Univ. Arizona, Tech. Rep. 88-20, 1988.

[4] T. Reps, T. Teitelbaum, and A. Demers, "Incremental context-dependent analysis for language-based editors," *ACM Trans. Programming Languages and Syst.*, vol. 5, no. 4, pp. 449-477, July 1983.

[5] A. Skarra, S. Zdonik, and S. Reiss, "An object server for an object-oriented database system," in *Proc. Object-Oriented Database Syst. Workshop*, Sept. 1986, pp. 196-204.

Induction of Meta-Knowledge about Knowledge Discovery

B. R. Gaines and P. Compton

***Abstract*—A study is reported of the use of ripple-down rule induction to develop a meta-model of ten years of clinical data captured as part of the development of an expert system for thyroid diagnosis. The study shows how the suitability for inductive knowledge discovery of such real-world data can be characterized in terms of its stationarity, and how the best error rates achievable and the amount of data necessary to achieve them, can be estimated.**

***Index Terms*—Garvan thyroid database, induct, induction, knowledge discovery, machine learning, medical diagnosis, meta-modeling, meta-knowledge, ripple-down rules, rules with exceptions.**

I. Introduction

The inductive modeling of large databases is a practical technique for knowledge discovery and decision support through knowledge-based systems [9]. It is usually expected that the model will be *predictive*, and can be used to forecast attributes of future data. However, Hume has shown that there are no logical foundations for such predictions [6]. The use of past results as predictors of future events is intrinsically fallible and should always be questioned. An inductive modeling methodology should include a meta-modeling activity to estimate the consistency of the data generation process over time. What is the best outcome that could be obtained in inductive modeling of the historic data? What structural changes have occurred that reduce the projectibility of the model? How much data are necessary to build a model, or rebuild one after major changes in the data gathering situation? Such meta-knowledge acquisition also provides information on the applicability of inductive modeling to related situations.

An impediment to empirical research on meta-modeling is the limited availability of large-scale datasets which have been used in decision-support applications. An increasing number of expert systems are coming into routine use, but it takes time for a significant amount of back data to become available. The study reported here uses data from the Garvan ES-1 thyroid diagnosis system that has been in routine use since 1984 producing some 6000 interpretations a year [5]. Each case is characterized by 20 binary attributes, six numeric attributes, and one of 61 diagnostic categories. Data on over 47 000 cases for the decade from 1980 through 1990 have enabled a number of studies to be carried out to determine how inductive models would have performed had modern modeling techniques been available throughout the expert system development. Interpretation of the results is aided by the availability of the manually developed rule set for ES-1, and a recent reconstruction of this using ripple-down rules [8]. For the studies below, consistent interpretation of the 47 000 cases were derived by using the final 1990 version of ES-1.

This note commences with a brief overview of the inductive modeling techniques used, and then focuses on the meta-modeling of the large dataset in terms of its predictability and the amount of data necessary to achieve the attainable error rates.

II. Ripple-Down Rules

The concept underlying ripple-down rules is that people cope with the acquisition and maintenance of complex knowledge structures by making incremental changes to them within a well-defined *context* such that the effect of changes is locally contained [1], [2]. Standard production rule systems do not have this property. The modularity of the rules is not reflected in a modularity of the consequences of changes to those rules, making the development and maintenance of rule-based systems difficult [7]. The ripple-down rule technique creates a two-way dependency between rules such that rule activation is investigated only in the context of other rule activation. If the premise of a parent rule is true for a particular individual, then, if it has no dependents, its conclusion will be asserted for that individual. If, however, it has an "if-true" de-

Manuscript received January 15, 1993. This work was supported by the National Sciences and Engineering Research Council of Canada and by the Australian Research Council.

B. R. Gaines is with the Knowledge Science Institute, University of Calgary, Calgary, Alta. Canada T2N 1N4.

P. Compton is with the Department of Computer Science, University of New South Wales, Sydney, NSW 2033, Australia.

IEEE Log Number 9212801.

1041-4347/93$03.00 © 1993 IEEE

pendent, then that rule and its dependents will be tested, and the original conclusion will only be asserted if the premises of none of them are true for the entity. Conversely, if the premise of a parent rule is false for a particular individual, then not only will its conclusion not be asserted, but also, if it has an "if-false" dependent, then it and its dependents will be tested [4].

The Garvan ES-1 system was developed manually over some six years and eventually comprised over 700 non-disjunctive production rules which became increasingly difficult to maintain. The ripple-down rule technique was tested by constructing a new expert system from scratch using some 9500 recent cases. The new rule set was generated manually in a short time, had 550 rules, with similar performance to the original production rules. Studies have also been carried out on the induction of production rules from these cases using C4.5 [10] and of ripple-down rules using Induct [3]. The dataset available was split into 8000 sample cases and 1514 test cases. The 550 manually developed ripple-down rules had 2.8% errors on the test set compared with 515 rules, 2.7% errors for C4.5 [8], and 195 rules, 2.3% errors for Induct [4]. These results on manual and automatic rule generation were the starting point for the meta-modeling studies.

III. Meta-Modeling Studies

The availability of the extended dataset of 43 472 cases has made it possible to develop meta-models of the Garvan data characterizing its suitability for inductive knowledge discovery. As a first experiment the database was split into eight successive datasets of 5434 cases each, and Induct was run on the entire dataset and on the eight subsets. The resulting rules were then evaluated on every dataset to give the table shown in Fig. 1, in which the rows represent the evaluation of one set of rules on the full dataset and the eight successive subsets. The errors in the first row are exceptionally low because they are those of a model based on the complete data, and those along the diagonal are low because they are evaluated on the dataset on which the model is based. What is of primary interest is the performance of a model on the dataset to the immediate right of the diagonal since this indicates how well the model predicts the coming data. This error rate ranges through 4.34, 5.72, 8.15, 6.26, 3.85, 3.86, and 3.79%. Following the error rate to the right, it tends to deteriorate substantially indicating that the database is not stationary.

To attempt to gain some insight into the dynamics of the Garvan data, and the way that this would have been tracked if an inductive model had been developed on a continuous basis, a second analysis was undertaken in which the complete dataset was modeled for the first 500 cases, the first 1000 cases, the first 1500 cases, and so on, up to 43 000 cases. Each model was then tested on the next 500 cases (472 for the last model) and the error rate plotted as shown in Fig. 2. It can be seen that the 1979–80 data did not predict for 1981 very well. Then, an improving model develops down to a low error rate in late 1982, followed by a period of higher errors in 1983–84, with a major change in late 1984, followed by a steady improvement from 1985 through 1990. The major unpredictability in 1984 was coincidental with changes in procedures in the hospital introduced at the start of the expert system development.

Since from Fig. 2 the hospital procedures seem to have stabilized leading to a reasonably stationary distribution from 1985, a third study was undertaken to estimate the amount of data necessary to induce an effective expert system for the recent Garvan data. The thousand cases from 46 001 through 47 000 were used as a target test set, and Induct was run on the 500 preceding cases, the 1000 preceding cases, and so on, in steps of 500 cases. The error rate on the 1000 following test cases that had not been used in induction

Model \ Test	Total Dataset	First Subset	Second Subset	Third Subset	Fourth Subset	Fifth Subset	Sixth Subset	Seventh Subset	Eighth Subset
Total	**0.46**	0.26	0.42	1.31	0.52	0.52	0.17	0.26	0.24
First	9.38	**0.52**	4.34	6.81	10.38	14.13	13.75	12.68	12.27
Second	9.30	5.98	**0.61**	5.72	9.40	13.75	14.32	12.94	11.72
Third	9.87	14.08	5.13	**1.56**	8.15	12.86	12.75	12.62	11.80
Fourth	7.08	12.16	9.92	6.04	**0.77**	6.26	7.53	7.29	6.70
Fifth	10.96	21.16	26.13	19.80	7.99	**0.57**	3.85	4.32	3.86
Sixth	11.43	21.62	26.17	21.51	9.79	4.62	**0.52**	3.86	3.39
Seventh	12.37	22.23	30.31	21.77	10.20	5.65	4.38	**0.66**	3.79
Eighth	11.43	19.57	25.97	20.67	9.27	5.70	4.77	4.16	**1.21**

Fig. 1. Analysis of Garvan data over ten years in eight successive datasets. (Each row shows the percentage errors on rules modeled from the datasets indicated in the left column and tested on the complete dataset and the eight subsets.)

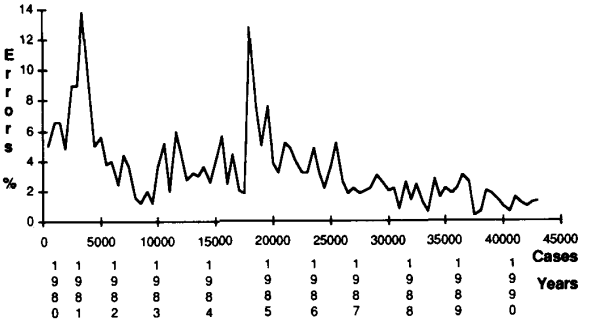

Fig. 2. Estimating the stationarity of the data: error rates on next 500 cases from models of Garvan database up to a certain time.

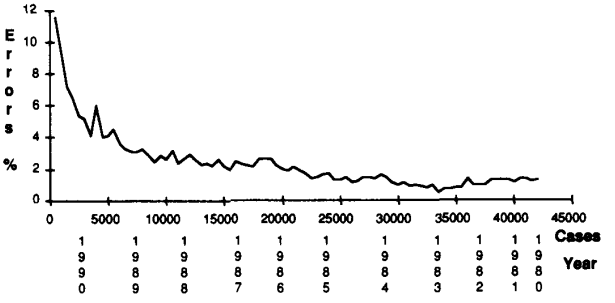

Fig. 3. Estimating the best error rate achievable and the amount of data required to do so: error rates on last 1000 cases from models of Garvan database from preceding cases.

was used as a measure of the quality of the expert system produced. Fig. 3 is a plot of the error rate against the number of cases used in induction. The error rate drops to, and remains below, 2.7% after 11 000 cases or three years of data. It drops to, and remains below, 1.4% after 29 000 cases or seven years of data.

These studies provide a meta-model of the Garvan data. It shows two major changes in its characteristics in mid 1981 and late 1984 sufficient to undermine preceding knowledge discovery and decision support systems based on it. It is stationary from late 1984 through 1990. A reasonably predictive model can be induced from the last three years data, but the error rate can be substantially improved using seven years of data.

IV. Conclusions

A study is reported of the use of ripple-down rule induction to develop a meta-model of ten years of clinical data captured as part of the development of an expert system for thyroid diagnosis. The study shows how such real-world data can be characterized in terms

of its stationarity, and hence suitability for inductive knowledge discovery, and how the best error rates achievable and the amount of data necessary to achieve them, can be estimated.

References

[1] P. Compton and R. Jansen, "Knowledge in context: A strategy for expert system maintenance," in *Proc. AI'88, 2nd Australian Joint Artificial Intell. Conf.*, Adelaide Austrailia, C. J. Barter and M. J. Brooks, Ed. Berlin: Springer, 1990, pp. 292-306.

[2] ——, "A philosophical basis for knowledge acquisition," *Knowledge Acquisition*, vol. 2, no. 3, pp. 241-258, 1990.

[3] B. R. Gaines, "An ounce of knowledge is worth a ton of data: Quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction," in *Proc. 6th Int. Workshop Machine Learning*. San Mateo, CA: Morgan Kaufmann, pp. 156-159.

[4] B. R. Gaines and P. Compton, "Induction of ripple-down rules," In: A. Adams and L. Sterling (eds.): AI '92. Proceedings of the 5th Australian Joint Conference on Artificial Intelligence. Hobart, Tasmania, World Scientific, Singapore 1992, pp. 349-354.

[5] P. J. Horn, P. J. Compton, L. Lazarus, and J. R. Quinlan, "An expert system for the interpretation of thyroid assays in a clinical laboratory," *Australian Comput. J.*, vol. 17, pp. 7-11, 1985.

[6] D. Hume, *A Treatise of Human Nature*. Oxford, England: Clarendon, 1888.

[7] X. Li, "What's so bad about rule-based programming?," *IEEE Software*, pp. 103-105, 1991.

[8] Y. Mansuri, J. G. Kim, P. Compton, and C. Sammut, "A comparison of a manual knowledge acquisition method and an inductive learning method," in *Proc. 1st Australian Workshop Knowledge Acquisition for Knowledge-Based Systems*. Sydney: Univ. Sydney, 1991, pp. 114-132.

[9] G. Piatetsky-Shapiro and W. Frawley, Ed., *Knowledge Discovery in Databases*. Cambridge, MA: MIT, 1991.

[10] J. R. Quinlan, "Simplifying decision trees," *Int. J. Man-Machine Studies*, vol. 27, no. 3, pp. 221-234, 1987.

Discovery of Inexact Concepts from Structural Data

Lawrence B. Holder and Diane J. Cook

***Abstract*—Concept discovery in structural data requires the identification of repetitive substructures in the data. We describe a method for discovering substructures in data using an inexact graph match. A previous implementation of our SUBDUE system discovers substructures based on the psychologically-motivated criteria of cognitive savings, compactness, connectivity and coverage. However, the instances in the data must exactly match the discovered substructures. We describe a new implementation of SUBDUE that employs an inexact graph match to discover substructures which occur often in the data, but not always in the same form. This inexact substructure discovery can be used to formulate fuzzy concepts, compress the data description, and discover interesting structures in data that are found either in an identical or in a slightly convoluted form. Examples from the domains of scene analysis and chemical compound analysis demonstrate the benefits of the inexact discovery technique.**

***Index Terms*—Chemical analysis, data compression, inexact graph match, scene analysis, substructure discovery.**

I. Introduction

Automated knowledge discovery is essential for extracting information from databases [2]. However, complex databases have moved from the simplistic attribute-value representation assumed by recent knowledge-discovery systems to a structural data representation that reflects the relationships among objects. Discovering concepts in this structural data requires the identification of common substructures within the data. The motivation for this process is not merely to find substructures capable of compressing the data by abstracting instances of the substructure, but also to identify conceptually interesting substructures that enhance the interpretation of the data. Substructure discovery is the process of identifying concepts describing interesting and repetitive substructures within structural data. Once discovered, the substructure concept can be used to simplify the data by replacing instances of the substructure with a pointer to the newly discovered concept. The discovered substructure concepts allow abstraction over detailed structure in the original data and provide new, relevant attributes for interpreting the data. Iteration of the substructure discovery and replacement process constructs a hierarchical description of the structural data in terms of the discovered substructures. This hierarchy provides varying levels of interpretation that can be accessed based on the goals of the data analysis.

Discovering interesting concepts requires a cognitive evaluation component and the ability to consider instances of the concept that do not exactly match the concept definition. We describe the SUBDUE system that utilizes psychologically motivated heuristics and an inexact graph match to discover substructures which occur often in the data, but not always in the same form. This inexact substructure discovery can be used to formulate fuzzy concepts, compress the data description, and discover interesting structures in data that are found either in an identical or in a slightly convoluted form. Examples from the domains of scene analysis and chemical analysis demonstrate the benefits of the discovery technique.

Manuscript received October 1, 1992; revised June 15, 1993.

The authors are with the Department of Computer Science Engineering, University of Texas at Arlington, Box 19015, Arlington, TX 76019.

IEEE Log Number 9212802.

II. SUBDUE

The SUBDUE system [3] discovers substructure in structured data based on four psychologically motivated heuristics: cognitive savings, connectivity, compactness and coverage. The *cognitive savings* of a substructure represents the net reduction in complexity when considering both the reduction in complexity of the input data after replacing each substructure instance by a single conceptual entity, and the gain in complexity associated with the definition of the new substructure. Substructure *compactness* is defined as the ratio of the number of edges to the number of nodes in a graph representation of the substructure. *Connectivity* measures the amount of external connection in the instances of the substructure. *Coverage* measures the fraction of structure in the input data described by the substructure. SUBDUE evaluates a substructure based on the product of these four heuristic values, each weighted by a user-supplied exponent.

The substructure discovery algorithm used by SUBDUE is a computationally constrained best-first search guided by the heuristics. SUBDUE represents structured data as a directed graph. The algorithm begins with the substructure matching a single node in the graph. Each iteration through the algorithm selects the heuristically best substructure and expands the instances of the substructure by one neighboring edge in all possible ways. The algorithm searches for the heuristically best substructure until all possible substructures have been considered or the amount of computation exceeds a given limit. Holder *et al.* [4] describe the SUBDUE system in more detail.

1041-4347/93$03.00 © 1993 IEEE