Working Paper 93-19
Statistics and Econometrics Series 15
September 1993

Departamento de Estadística y Econometría
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (341) 624-9849

ON THE AUTOMATED EXTRACTION OF REGRESSION KNOWLEDGE FROM DATABASES

Jorge Muruzábal*

Abstract

The advent of inexpensive, powerful computing systems, together with the increasing amount of available data, conforms one of the greatest challenges for next-century information science. Since it is apparent that much future analysis will be done automatically, a good deal of attention has been paid recently to the implementation of ideas and/or the adaptation of systems originally developed in machine learning and other computer science areas. This interest seems to stem from both the suspicion that traditional techniques are not well-suited for large-scale automation and the success of new algorithmic concepts in difficult optimization problems.

In this paper, I discuss a number of issues concerning the automated extraction of regression knowledge from databases. By regression knowledge is meant quantitative knowledge about the relationship between a vector of predictors or independent variables (x) and a scalar response or dependent variable (y). A number of difficulties found in some well-known tools are pointed out, and a flexible framework avoiding many such difficulties is described and advocated. Basic features of a new tool pursuing this direction are reviewed.

Key Words
Knowledge Discovery; Flexibility; Robustness; Low Structure.

*Departamento de Estadística y Econometría, Universidad Carlos III de Madrid.

On the automated extraction of regression knowledge from databases

Jorge Muruzábal[1]
Departamento de Estadística y Econometría
Universidad Carlos III de Madrid

Abstract

The advent of inexpensive, powerful computing systems, together with the increasing amount of available data, conforms one of the greatest challenges for next-century information science. Since it is apparent that much future analysis will be done automatically, a good deal of attention has been paid recently to the implementation of ideas and/or the adaptation of systems originally developed in machine learning and other computer science areas. This interest seems to stem from both the suspicion that traditional techniques are not well-suited for large-scale automation and the success of new algorithmic concepts in difficult optimization problems.

In this paper, I discuss a number of issues concerning the automated extraction of regression knowledge from databases. By regression knowledge is meant quantitative knowledge about the relationship between a vector of predictors or independent variables (x) and a scalar response or dependent variable (y). A number of difficulties found in some well-known tools are pointed out, and a flexible framework avoiding many such difficulties is described and advocated. Basic features of a new tool pursuing this direction are reviewed.

Keywords: Knowledge Discovery, Flexibility, Robustness, Low Structure.

1 Introduction

Computers are changing society. Due to the speed and accuracy with which they retrieve, process and store information, computers not only have freed humans from tedious, error-prone tasks, but also have become indispensable research tools in many scientific disciplines. In recent years, the impact of the computer revolution has been particularly noticeable in the areas of information processing and data analysis, where the increasing affordability of computing power has opened new horizons at the level of both research and application.

The widespread availability of computers has also brought some new challenges. For example, large databases are routinely collected and stored in both scientific and business environments. Uncovering the useful regularities hidden inside such databases is of course a natural goal, but one for which we may not have enough time! Indeed, it has been estimated that the total volume of raw information in the world doubles every twenty months [Frawley et al. 1991], so the risk of being literally flooded by data is a serious one. Also, because of the inherent complexity introduced by sheer volume, it seems unlikely that most of these data will ever be seen by human eyes. Instead, the search for regularities will

[1] This article is based on the author's Ph. D. dissertation; an earlier version appears in the Proceedings of COMPSTAT 92, Physica-Verlag, Germany. Support for this research has been provided in part by grant NSF/DMS-8911548-02 (U.S.A.), by the Economics Research Program at the Santa Fe Institute, Santa Fe, New Mexico, and by grant DGICYT PB92-0246 (Spain).

be most likely taken up by *autonomous* computer programs. Despite the evident statistical content of this quest, the first organized body of research has been developed without much input from classical statistics, cf. [Piatetsky-Shapiro and Frawley 1991]. In contrast, attention has often shifted to machine learning models and techniques.

A little reflection shows that this phenomenon should not be very surprising. Many standard statistical methods start off by imposing a relatively rich structure of distributional assumptions on the data, then rely heavily on user input to refine such assumptions and finally deliver a *compact*, high-level model as summary of the analysis. To meet the challenge of automated inference in large, little understood databases, these methods seem inadequate. First, from the point of view of the information being handled, not only rich structure will often be hard to elicit, but also a high-level model may go well beyond the scientist's most realistic expectations (who will be initially more interested in the synthesis of *local* regularities than in the development of a full-fledge theory). Further, prior available knowledge involving symbolic or structural information is usually hard to incorporate. Second, from a computational viewpoint, mechanization of these methods must resolve lots of sophisticated processing and decision making, and thus poses formidable difficulties. In addition, they tend to suffer from the so-called *curse of dimensionality* (they do not scale up well as the problem's basic dimensions increase). Some of these issues are further discussed in section 2 below.

In view of these apparent shortcomings, alternative methods must be developed. Such methods should (i) impose weaker distributional assumptions; (ii) avoid high-level decision-making; (iii) exhibit an exploratory rather than confirmatory nature; (iv) support the expression of as many kinds of nonnumeric information as possible; and (v) be parsimonious in their use of basic resources. Interpretability and analytical tractability seem additional desirable features.

While the previous remarks are relevant in various settings, in this paper we are concerned with the problem of regression, that is, the problem of inferring the relationship between a stimulus or vector of predictors (x) and a scalar response (y) on the basis of a sample of observed pairs D= $\{(x_i,y_i)\}$. Without loss of generality, all predictors are hereafter assumed boolean and y is taken to vary over the unit interval (0,1). Regression knowledge is usually expressed in the form of one or more conditional density functions f(y/x) or one or more conditional expectations E(y/x), the latter being far more common in the literature. Individually, these objects convey (predictive) information in that they can be used to anticipate the response associated with new stimuli. Collectively, they may also provide some insights on the overall effect of single predictors or suggest interactions among two or more predictors.

The rest of the paper is organized as follows. Expanding on the previous remarks, section 2 contains a general discussion on automated data analysis. Two specific tools are reviewed in sections 3 and 4. Section 5 presents a new framework and section 6 sums up and further comments on the main ideas.

2 On automated data analysis

In recent years, artificial intelligence (AI) models and methods have begun to be explored by statisticians and other workers interested in the development of *intelligent* tools for data analysis. It is interesting to note that the process of adapting an AI methodology to form the basis of a purely data-analytic tool is not frequently seen in the statistical literature (an example is provided in [Phelps and Musgrove 1986]). Instead, a traditional source of motivation has been the recognition of an important caveat in standard statistical packages:

They supply advanced numerical processing but little guidance about its proper use and interpretation. As a result, much attention has been cast to the problem of mechanizing statistical strategy using AI machinery, and many ideas on expert systems providing various kinds of automated *assistance* have been discussed, cf. [Hahn 1985], [Tukey 1985,1986], [Thisted 1986], [Huber 1986], [Bates and Chambers 1987], [Defays 1989].

A well-known working example of this kind of expert systems is REX [Gale 1986]. REX operates in the context of multiple linear regression and implements a particular strategy developed by its creators (Gale and Pregibon). REX's suggestions are based on a set of predefined cut points that determine the distinction between mild and severe problems. It can provide explanations to back up its suggestions and it maintains in general a constructive dialogue with the user, who always has the last word during the analysis.

A different issue is how to carry out the analysis automatically. With this purpose, an obvious approach is to grant REX the ability to make decisions on its own. Indeed, a number of commercial systems following this idea are available. However, it may be unreasonable to assume that all important intermediate steps involved in this kind of model-building can be taken on the basis of statistical information alone. For example, one often needs to consider specific knowledge of both the subject matter and the data collection process, cf. [Lauritzen and Spiegelhalter 1988], [Schaffer 1989]. Even if such information is integrated (which is by no means a trivial step), the principles used by human analysts may be highly context dependent in complicated ways. Thus, the existence of a clear-cut flow chart to guide the process has been seriously questioned by some authors (see eg. [Huber 1986]).

The kind of inferential framework used in REX and other systems is likely to lie at the root of these problems. As mentioned earlier, standard model-based statistical methods rely heavily on the human analyst, who is in charge of selecting a suitable initial model, evaluating it in the light of the data and introducing subsequent refinements (that is, changing the representation). Automating the sophisticated reasoning behind the latter tasks is difficult because, among other things, knowledge is often expressed *globally*, so the consequences of any change in representation spread everywhere (consider, for example, the effect of deleting one variable in a nonorthogonal regression model). This complexity may be tolerated by the human analyst, but it would seem overwhelming for an algorithm.

The solution is not simply to avoid expressing knowledge globally: even when knowledge is expressed locally, different components may be highly constrained by the model, in which case manipulation tends to be awkward. The problem can be seen, for example, when trying to modify the set of local characteristics of a Markov random field, cf. [Geman and Geman 1984]. It is also an issue in tree-oriented inference, see below.

It might thus seem that the most promising approach not only expresses knowledge locally, but also avoids placing strong distributional or structural constraints among the components of the model. In this case, each component is relatively independent of the rest, reasoning may be restricted to at most a few components at a time, and individual components can be manipulated in a more straightforward manner. Since no global adjustments are needed in principle, substantial computational savings are likely to obtain. Such a knowledge representation is called *disintegrated*. An example of disintegrated representation (along with further discussion) is provided below in section 5.

The lack of rich structure provides also an alternative path to robustness. As is well-known, adopting a rich framework entails the risk of introducing an adverse, permanent bias in the learning process, be it automated or not. Hence, robustness is concerned with the study of the extent to which conclusions depend on prior assumptions: methods that are highly sensitive to small perturbations in the postulated distributions should be avoided in favor of more robust alternatives. In contrast, methods capable of proceeding in the absence of strong constraints follow the premise that automated learning systems should be permitted to evolve freely under milder *inductive biases*. In general, an inductive bias is loosely understood as any built-in mechanism that curbs induction. In machine learning practice, inductive biases appear in many different forms, such as (soft) constraints on representation, evaluation functions or heuristic operators. None of them is very restrictive, nor they typically carry long-term implications. Moreover, heuristic operators often exhibit a substantial amount of randomization. The bias introduced by strong distributional assumptions can be viewed as an extreme case: it drives induction along a relatively narrow path.

Because the acquired knowledge will often need to be reexpressed or qualified during the learning process, flexibility of representation is a crucial design issue. By this I mean two things: first, the system should be able to represent a *large* class of patterns; second, it should be easy to shift from one representation to another as data accumulates. It is useful to illustrate the above ideas in terms of two specific systems: BACON.6 and FIRM. Both systems express knowledge locally and both proceed in the absence of distributional assumptions; however, both impose relatively strong structural constraints among model components. This is argued to limit their flexibility and, therefore, their power as vehicles for learning. In order to support these claims, a fairly detailed discussion of each system is provided in turn.

3 BACON.6

Closely related to linear model theory, the system BACON.6 [Langley et al. 1986] provides an environment for automatic discovery of empirical laws summarizing numerical data. BACON.6 is able to rediscover, among others, the ideal gas law relating pressure, volume and temperature. Other discovery systems have been developed to complement and extend BACON.6, see [Langley et al. 1986] and [Zytkow and Baker 1991].

BACON.6 is designed to work on predictors with a finite number of quantitative levels (this excludes the case of boolean predictors, but let us focus on the inferential engine). A single-replicate full factorial data tree is assumed available for analysis. The system searches through a space of general laws by nesting pairwise relationships as follows: By fixing the first k-1 predictors at their lowest levels, BACON.6 first finds the best fitting functional form between the levels of the last predictor and the response. Only a relatively small set of parametric functional forms (supplied by the user) is considered at this or any subsequent stage; this set may include forms like $y=ax+b$, $y^{-1}=ax+b$, and others. Hence, the search through the space of laws essentially reduces to a (hierarchical) parameter estimation problem. For each form, parameters are fitted so they maximize the correlation between predicted and actual data values, and both the best fitting form(s) plus the optimizing values for its (their) parameters are stored at the corresponding node of the data tree.

BACON.6 next considers in turn all parallel nodes exploring the same relationship at the remaining values of predictor k-1 (keeping the first k-2 predictors at their lowest levels as before). In principle, the functional forms selected at each node may or may not include a common element, but BACON.6 proceeds by discarding all unsuitable forms found at any time during the scan. As a result, as soon as there remains exactly one suitable form, it will be automatically fitted to all data at the remaining nodes. Hence, the system "redefines its problem space in the light of its previous experience, so that considerably less search results", [Langley et al. 1986; p. 434]. However, why the system should decide on the global unsuitability of a given form on the basis of a *single* projection and how exactly its unsuitability is measured are not explicitly discussed in the paper.

At any rate, a unique functional form is adopted between predictor k and the response. Fitted parameter values at this level play the role of the response at the next level, in which the system explores similarly the relationship between each of them and the corresponding values of predictor k-1. The process continues by backward chaining until all intermediate parameters are estimated and all intermediate nodes filled. At this point, the resulting functional form between predictors and response is untangled. To illustrate, from $y=ax_1+b$, $a=2x_2$ and $\log(b)=x_2$, one obtains the law $y=2x_1x_2 + \exp(x_2)$.

In addition to previous criticisms (for example, Schaffer [1989] has argued that discoveries reached by BACON.6 are in fact gauged by critical system parameters specified *ad-hoc* by the programmers), the point can also be made that BACON.6 fails to comply adequately with the basic requirement of flexibility. For example, the system is restricted to a prespecified set of functional forms. More importantly, no hint of backtracking in the space of laws is provided, so knowledge can only spread along a tight one-way lane. Thus, decisions made early during any of the various decision-making processes (starting with the order in which predictors are arranged) may have an irreversible impact on the quality of the inferred relationship. It is also important to note that the system ignores two sources of uncertainty, namely uncertainty related to form selection (presumably needed to combine information from parallel nodes) and uncertainty related to parameter estimation (as fitted parameters are used regardless of their inherent variability).

4 FIRM

The last decade has seen the confirmation of *recursive splitting* or *tree-oriented* algorithms as useful tools for automated regression, see [Breiman et al. 1984] for a thorough exposition and [Quinlan 1990] or [Crawford 1990] for general reviews. Although these systems were not conceived to analyze large databases, they provide a useful contrast for the framework to be introduced in the next section. We first focus on a specific tree-oriented system called (CON)FIRM, [Hawkins 1990]. In addition, a generalization due to Friedman [1991] is briefly introduced to help clarify some points on flexibility.

FIRM proceeds in *batch* mode (that is, examining all data at once) and recursively creates an exhaustive, top-down partition of the training sample D. At step 0, the root node contains the whole data set D. For each available predictor, FIRM considers the standard one-way significance level based on the F distribution with one degree of freedom. Depending on the value of the most significant predictor, D is split into two subsets which form the two descendant nodes of the root node. This completes step 1; we have a small tree with three nodes. The procedure is now repeated separately at each of the lower nodes; in particular, splits at sibling nodes may be based on different predictors. Any lower node is split unless its most significant predictor does not reach some prespecified level of significance. The growing phase ends (and the resulting tree is output) when no further splitting is possible at any terminal node. When a new x arrives, one simply follows the branch of the tree to which x belongs down to the corresponding terminal node, where the mean and variance of the corresponding subset of the training sample lead respectively to a point prediction and variability estimate for the associated response.

This *forward selection* criterion to build the output tree departs from other recursive-splitting systems in that it never backtracks. For example, both ID3 [Quinlan 1986] and CART [Breiman et al. 1984] create first an overgrown tree that fits the training data tightly. Because of the presence of unnecessary splits due to random associations in the training sample (overfitting), this tree will tend to make undue mistakes on test data. To solve this problem, these systems carry out a *pruning* phase: splits are reevaluated in a bottom-up manner and splits may be merged sequentially to yield simpler trees. However, early decisions affecting decisively the distribution of the data throughout the nodes are still likely to be rarely revised (this is also the case for *incremental* tree-oriented systems not discussed here, see eg. [Van de Velde 1990]). A related problem is that sometimes the same exact subtree is replicated in parallel along various branches (Quinlan has devised an algorithm that translates the knowledge expressed in the final tree into a set of standard production rules that avoid this unpleasant redundancy). These difficulties may be seen as part of the price that must be paid to maintain a clean, mutually exclusive organization.

Friedman [1991] has extended the recursive-splitting methodology in various ways to achieve, among other things, a higher degree of flexibility. The upgrades in Friedman's new regression technique, called MARS (Multivariate Adaptive Regression Splines), can be summarized as follows. First, splitting a node no longer means its removal as a bearer of predictive information: both parent and children nodes contribute to determine the predicted response at the children nodes (the knowledge structure returned by MARS is no longer a tree and hence "node" is used in a generalized sense here). Second, splitting is not restricted to children nodes: the same parent may be split several times and their children from different splits may coexist with one another. This permits easy induction of main-effect models, which is unnatural in conventional tree-oriented methods. Third, a low-degree spline, rather than a constant, is fitted at each node. Lastly, splitting and deleting decisions are based on the global fit of linear combinations of splines, not on local information at the individual nodes. In particular, any node may be selected for splitting and any node (including those created early during the growing phase) may be deleted during the pruning phase. To sum up, by fitting different types of splines and combining splines from overlapping nodes, it is evident that a much wider variety of local patterns can be represented.

5 PASS

This section describes the basic ideas in PASS (Predictive Adaptive Sequential System), a simple classifier system (CS) for the automatic extraction of regression knowledge. For a general introduction to CSs, see [Holland 1986,1989]. An early, qualitative CS approach to the problem of learning in growing knowledge bases is proposed in [Holland 1980]. A general learning theorem in a markovian framework is proved in [Holland 1986b]. A more comprehensive account of the ideas underlying CSs is laid out in [Holland et al. 1986], which includes links with psychological and other kinds of learning. For a detailed description of PASS and experimental results, see [Muruzábal 1992,1993]. A closely related problem involving a similar knowledge representation is tackled in [Packard 1989].

In contrast to the systems discussed earlier, PASS processes data pairs (x,y) one at a time. Starting with the first pair, it always tries to improve on the most suitable knowledge structure to express the regularities found in the data stream. In practice, data will be sampled (and possibly resampled) from a finite matrix in a random or fixed manner; in simulations, the data stream is generated as a sequence of independent replicates from some prespecified joint distribution. Since no data are permanently stored, the amount of memory required by PASS depends only on system parameters.

The model entertained by PASS is a collection of unstructured classifiers (that is, the model uses a disintegrated representation). The system enjoys a great deal of freedom to handle its own resources -- it can, for example, create multiple copies or variations of a useful classifier (compare to FIRM and MARS as discussed above). Classifiers themselves are somewhat limited for convenience of design. They consist of four parts: (i) The standard schema s, a subset of stimulus space determined by some common coordinates. (ii) A predictive distribution d. This is implemented as an evolving discrete probability distribution with small convex support over an arbitrary number of equally-sized subintervals on the unit interval. (iii) The standard strength S, a measure of the classifier's current utility. And (iv) The set of exceptions E, a form of short-term, local memory. The first two components reflect each a tentative expression of a perceived regularity; the third measures the confidence the system has in the classifier (regularity); the last one constitutes the basic information on which the system tries to improve the knowledge base.

The system follows the standard stimulus/response/reinforcement cycle. At each step, it first sees the stimulus x, which triggers some subset of the the current population of classifiers. The associated elementary predictions d are merged to configure the system's predictive distribution for the unseen y, say f. The actual y is then provided and used with various purposes: (i) to change the strength of triggered classifiers according to their individual d; (ii) to monitor the system's overall progress by means of some increasing function of f(y), the predictive probability assigned to the observed response; and (iii) to be eventually studied (along with other data values in E) by the system's heuristic operators (inductive biases). The first two uses do not require the knowledge nor the storage of the precise value of y, nor is every y always stored in some exception set E. Once reinforcement and evaluation are completed and (possibly) heuristic operators have performed some changes in the population, the system is ready for a new (x,y).

Individual classifiers are not only to tune their more relevant segments (s,d) to exploit every regularity in the data stream; they may also evolve to associate with other classifiers and form *emergent* knowledge structures. The latter feature is more rarely seen as it seems to depend on computational details in ways that are only beginning to be understood. Some studies on the emergence and stability of *default hierarchies* in general CSs are provided by [Riolo 1987,1989] (a default hierarchy is a list of classifiers with nested, increasingly more specific schemata). Considering perhaps that inductive biases based on strength alone are too mild to guide emergence effectively, some authors have introduced partial structure in the population of classifiers, cf. [Shu and Schaeffer 1991].

Since each classifier in PASS can be interpreted in isolation, the system has been shown to reconstruct some simple linear regression models. However, structure may integrate itself in a way that may not be always obvious to a human outsider. Some kind of integration of knowledge may be much helpful as the number of predictors or classifiers increase. For example, compressing routines as those in [Zhou 1990] may be appropriate.

Operators act on the knowledge base by both modifying the strength, schema or distribution of existing classifiers and introducing new ones. These mechanisms depend largely on local assessment (measured by strength and other summaries), but it may be profitable to make them depend also on global assessment (as measured by performance). A typical example of the latter dependence is to decrease the system's overall exploration rate as the learning curve estabilizes. System performance surprisingly stabilizes despite the fact that only a minor fraction of classifiers stay for longer periods of time.

As it regards performance, it is not hard to find system parameters that achieve satisfactory learning rates in various simulated testbeds. In particular, PASS is able to detect fairly diffuse regularities. It is also able to generalize appropriately even under high noise/contamination rates. Finally, it achieves only slightly worse results than FIRM, yet it does never consider big subsets of data at once. However, scale-up factors are vague yet; due to the significant amount of randomization in PASS, computational complexity is difficult to analyze.

6 Summary and concluding remarks

This paper presents first some general comments on automated inference, then goes on to discuss various approaches to the problem of extraction of regression knowledge from databases. Although these approaches express knowledge locally, they exhibit markedly different data processing styles, inference engines, and output structures. Compared to BACON.6 and FIRM, PASS seems flexible, robust, memory-wise parsimonious and potentially more informative. It does also seem to constitute a rich experimental framework where many computational ideas, often cognitively inspired, can be tested. Basic research should be conducted to precise scale-up factors and clarify the role of certain system parameters. Possible extensions include replacing schemata with other types of subsets in stimulus space, introducing alternative families of predictive distributions, and imposing loose structural linkage among classifiers.

References

[Bates and Chambers 1987]
D. M. Bates and J. M. Chambers. *Statistical Models as Data Structures.* Statistical Research Report # 42, AT&T Bell Labs, Murray Hill, NJ.

[Breiman et al. 1984]
L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, CA.

[Crawford 1990]
S. L. Crawford. *Extensions to the CART algorithm.* In B. R. Gaines and J. H. Boose (Eds.) Machine Learning and Uncertain Reasoning. Academic Press, New York.

[Defays 1989]
D. Defays. *Statistics and Artificial Intelligence.* In E. Diday (Ed.) Data Analysis, Learning Symbolic and Numeric Knowledge. Nova Science, New York.

[Frawley et al. 1991]
W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus. *Knowledge Discovery in Databases: An Overview.* In G. Piatetsky-Shapiro and W. J. Frawley (Eds.) Knowledge Discovery in Databases. MIT Press, Cambridge, MA.

[Friedman 1991]
J. H. Friedman. *Multivariate adaptive regression splines.* The Annals of Statistics, 19.

[Gale 1986]
W. A. Gale. *REX Review.* In W. A. Gale (Ed.) Artificial Intelligence and Statistics. Addison-Wesley, Reading, MA.

[Geman and Geman 1984]
S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.* IEEE Trans. on Pattern Analysis and Machine Intelligence, 6.

[Hahn 1985]
G. J. Hahn. *More Intelligent Statistical Software and Statistical Expert Systems: Future Directions.* The American Statistician, 39.

[Hawkins 1990]
D. M. Hawkins. *Formal Inference-based Recursive Modeling.* Technical Report # 542, University of Minnesota, Minneapolis, MN.

[Holland 1980]
J. H. Holland. *Adaptive Algorithms for Discovering and Using General Patterns in Growing Knowledge Bases.* International Journal of Policy Analysis and Information Systems, 4.

[Holland 1986a]
J. H. Holland. *Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems.* In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.) Machine Learning: An Artificial Intelligence Approach II. Morgan Kauffman, San Mateo, CA.

[Holland 1986b]
J. H. Holland. *A Mathematical Framework for Studying Learning in Classifier Systems.* Physica D, 22.

[Holland 1989]
J. H. Holland. *Using Classifier Systems to Study Adaptive Nonlinear Networks.* In D. L. Stein (Ed.) Lectures in the Sciences of Complexity: Proceedings of the 1988 Complex Systems Summer School, Santa Fe Institute. Addison-Wesley, Reading, MA.

[Holland et al. 1986]
J. H. Holland, K. J. Holyoak, R. E. Nisbett and P. R. Thagard. *Induction: Processes of Inference, Learning and Discovery.* MIT Press, Cambridge, MA.

[Huber 1986]
P. J. Huber. *Environments for Supporting Statistical Strategy*. In W.A. Gale (Ed.) Artificial Intelligence and Statistics. Addison-Wesley, Reading, MA.

[Langley et al. 1986]
P. Langley, J. M. Zytkow, H. A. Simon and G. L. Bradshaw. *The search for regularity: four aspects of scientific discovery*. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.) Machine Learning: An Artificial Intelligence Approach II. Morgan Kauffman, San Mateo, CA.

[Lauritzen and Spiegelhalter 1988]
S. L. Lauritzen and D. J. Spiegelhalter. *Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems*. Journal of the Royal Statistical Society (B), 50.

[Muruzábal 1992]
J. Muruzábal. *A machine learning approach to a problem in exploratory data analysis*. Ph. D. thesis. University of Minnesota, Minneapolis, MN.

[Muruzábal 1993]
J. Muruzábal. *PASS: a simple classifier system for data analysis*. Submitted for publication.

[Packard 1989]
N. H. Packard. *A Genetic Learning Algorithm for the Analysis of Complex Data*. Technical Report, Center for Complex Systems Research and the Physics Department, University of Illinois at Urbana, IL.

[Piatetsky-Shapiro and Frawley 1991]
G. Piatetsky-Shapiro and W. J. Frawley (Eds.) *Knowledge Discovery in Databases*. MIT Press, Cambridge, MA.

[Phelps and Musgrove 1986]
R. I. Phelps and P. B. Musgrove. *Artificial Intelligence Approaches in Statistics*. In W. A. Gale (Ed.) Artificial Intelligence and Statistics. Addison-Wesley, Reading, MA.

[Quinlan 1986]
J. R. Quinlan. *Induction of Decision Trees*. Machine Learning, 1.

[Quinlan 1990]
J. R. Quinlan. *Probabilistic Decision Trees*. In Y. Kodratoff and R. S. Michalski (Eds.) Machine Learning: An Artificial Intelligence Approach III. Morgan Kauffman, San Mateo, CA.

[Riolo 1987]
R. L. Riolo. *Bucket brigade performance II: Simple default hierarchies*. In J. Grefenstette (Ed.) Proceedings of the Second International Conference on Genetic Algorithms and Their Applications. Lawrence Erlbaum, Hillsdale, NJ.

[Riolo 1989]
R. L. Riolo. *The emergence of default hierarchies in learning classifier systems*. In J. D. Schaffer (Ed.) Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kauffman, San Mateo, CA.

[Schaffer 1989]
C. Schaffer. *BACON, Data Analysis and Artificial Intelligence*. In A. M. Segre (Ed.) Proceedings of the Sixth International Workshop on Machine Learning. Morgan Kauffman, San Mateo, CA.

[Shu and Schaeffer 1991]
L. Shu and J. Schaeffer. *HCS: Adding Hierarchies to Classifiers Systems*. Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kauffman, San Mateo, CA.

[Thisted 1986]
R. A. Thisted. *Knowledge Representation for Expert Data Analysis Systems*. In D. M. Allen (Ed.) Computer Science and Statistics: The Interface. North-Holland, New York.

[Tukey 1985]
J. W. Tukey. In the discussion of [Hahn 1985].

[Tukey 1986]
J. W. Tukey. *An Alphabet for Statisticians' Expert Systems*. In W. A. Gale (Ed.) Artificial Intelligence and Statistics. Addison-Wesley, Reading, MA.

[Van de Velde 1990]
W. Van de Velde. *IDL, or Taming the Multiplexer*. In K. Morik (Ed.) Proceedings of the Fourth European Working Session on Learning. Morgan Kauffman, San Mateo, CA.

[Zhou 1990]
H. H. Zhou. *CSM. A Computational Model of Cumulative Learning*. Machine Learning, 5.

[Zytkow and Baker 1991]
J. Zytkow and J. Baker. *Interactive Mining of Regularities in Databases*. In G. Piatetsky-Shapiro and W. J. Frawley (Eds.) Knowledge Discovery in Databases. MIT Press, Cambridge, MA.