Evaluating and Comparing Classifiers: Complexity Measures

J. Kent Martin
jmartin@ics.uci.edu

October 26, 1994

Abstract

Relevant literature on Kolmogorov complexity measures and on trade-offs of classifier accuracy for reduced complexity is reviewed, seeking a pragmatic methodology for the practising applications analyst. Significant findings are that: (1) An accuracy/complexity trade-off is desirable; (2) Combined measures of accuracy/complexity are not practical due to difficulties encoding constraint satisfication, lack of sampling statistics and suitable tests of the null hypothesis, and practical difficulties of encoding complex functions and encoding across families of classifiers; (3) Therefore, a generalized version of the CART [5] 1-SE rule is recommended; (4) Kolmogorov complexity is not practically computable (see (2)); and, therefore, (6) Simply measuring response times on a target environment is the recommended measure of complexity.

1 Introduction

Most classification and clustering tools are intended to induce from a sample an efficient and accurate method for predicting class membership of future instances from the same distribution. Comparing the performance of the resulting classifiers can be quite problematic. There is a substantial body of literature dealing with the philosophical, theoretical, and pragmatic issues behind these questions. Among the most fundamental problems in this regard are:

1. Whether it is appropriate to make a trade-off of increased misclassification for reduced complexity and, if so, how this is to be done.
2. How to combine storage requirements and execution time into a single measure of complexity.

2 Precision versus Simplicity

Where the sample is the entire population, in one sense, the sample is its own best description, with two exceptions:

1. There is a computational model giving the class values exactly as a function of the attribute values, and this model is simpler (more tractable) than the set of instances.
2. The data are inconsistent (two or more instances with identical attribute values have different classes).

In the first case, certainly the model would be the preferred description. In the second case, if there were a model which covered all of the consistent instances correctly, the model would again be the preferred description. (Preferred in the sense of being equally informative and having lower cognitive burden).

When the available sample is not the entire population, especially when the data probably contain errors or when the population is potentially infinite, the model must be preferred even if it correctly covers only, say, 90% of the sample instances. The model provides a significant and necessary compression of both time and space. This implies some trade-off between apparent accuracy and description complexity.

There is a deeper question, whether any trade-off can be justified; and, if so, how? A common justification [4] invokes *Occam's Razor*, but Occam merely stated a principle to be applied when other considerations are equal. There is no necessary notion of a trade-off in Occam's principle.

Another common justification cites the results of some statistical analyses ([2, 4, 17, 22]) as providing a guarantee (or, at least, a high degree of certainty) that a simpler model will result in greater accuracy in classifying as yet unseen cases. This simply reads too much into the results of these analyses; often based on the misapprenhension that the sample itself, alone, can provide unequivocal evidence for choosing among alternative models which fit the sample data about equally well. Schaffer [20] has shown that it is easy to find large sets of counter-examples; and, therefore, that this preference for simpler models is a form of bias, whose appropriateness cannot be decided without reference to factors in addition to the particular data sample.

There is a another justification for this bias, rooted in the fundamental purpose and nature of categorization in human cognition. Categorization arises biologically from the organism's needs to impose order and simplicity on the infinite variety of situations and to match inputs to appropriate behaviors efficiently and in real-time. Rosch [19] makes similar arguments, viewing basic categories as those that strike an optimum balance between informativeness and cognitive load. That is, that the whole point of categorization is to achieve a proper trade-off of precision versus simplicity and efficiency.

3 Description versus Prediction

For non-categorical data, there are established methods for making this trade-off, based on analysis of variance, regression, and model building (see Beck and Arnold [3, pp 380-387], for instance); but what are the analogs of these methods for categorical data?

The basic idea is that the population variance can be partitioned into a part covered (predicted) by the model and a residual variance. If two models have equivalent residual variances, the simpler of the two is preferred; if the residuals are not equivalent, the model having the lower residual is preferred. The difference between residuals is compared to an error variance using the F-distribution ([1, pp 436-441]). For categorical data, the misclassification frequencies are analogous to the residuals, but the F-test is not applicable. The error rates are binomially distributed, and some test of significance for binomial means is required. Student's t-test (though not always appropriate) is typically used.

There have been several proposals (Wallace, *et al* [23, 10, 24], Muggleton *et al* [16]) for combining precision, storage, and run time into a single measure and criteria, variously described as minimum description length (MDL), or minimum message length (MML), or hypothesis proof (HP) compression, etc. All of these proposals share certain deficiencies:

1. How is a constraint satisfaction problem to be encoded? That is, if one model satisfies the constraints on misclassification rates and the other does not, how do we weight the description length to insure that the second, non-conforming, model loses?

2. How are misclassification costs to be incorporated? How many dollars per extra bit of model complexity will achieve the right trade-off?

3. What are the sampling statistics of these measures? That is, Muggleton, *et al* [16], for instance, propose that even one bit of compression is grounds for including an additional term in the model. Can that be right? (By their own theorem, random binary data can be compressed by k bits as often as 2^{-k} of the time, *i.e.*, this is a test at only the 50% confidence level.) If another sample were drawn from the population and the procedures repeated, would the same model win by this criterion (for that matter, would precisely the same models even be inferred)?

4. How is complexity to be encoded, particularly across different species of classifiers? Muggleton, *et al* [16], give a method for encoding Horn clause theories (models) and proofs on the input tape of a reference Turing machine. In principle, any classifier can be translated into a Horn clause representation, but this is not always a trivial task. And then it is not at all clear how, for instance, the translation from inputs to an output cell of a backpropagation network would be represented here; or that the resulting encoding would make sense.

If not by description length, then how? The following is a generalization of the CART (Breiman, *et al* [5, pp 78-80]) 1-standard-error (1-SE) rule, and Weiss' [25] reduced-complexity rule. Note that the initial and final steps appeal to common sense and knowledge of the problem domain; that is, to sources outside the sample data.

1. If any of the models satisfies the cost constraints, reject all that don't. If none do, use common sense and knowledge of the problem domain to decide whether to continue and choose among any of the models (and, if so, which). Of the remaining models, find the one having the lowest cost, and determine this cost's standard error.

2. Discard all models whose cost is more than t standard errors greater than the minimum (where $t = 1$ (CART [5]) or, preferably, Student's t statistic [1, pp 314-321], typically $t \approx 1.65$, for 95% confidence).

3. Of the remaining models, choose the one(s) having the lowest average complexity (see section 4). If there is a near-tie, use common sense and knowledge of the problem domain to choose.

4 Complexity

The generalized t-SE rule eliminates a major difficulty of the MML and compression techniques — how to express complexity in the same units as misclassification cost. The difficulty remains, however, of exactly what complexity means, and how it is to be measured.

Li and Vitányi [15] give an account of various proposed measures of complexity (Rissanen's minimum description length (MDL) [18], Fisher's maximum likelihood principle [9], Jayne's maximum entropy principle [12], Gold's paradigm for inductive inference [11], and even Valiant's [22] PAC-learning notion) in terms of Solomonoff's [21] ideas on inductive reasoning, viewing each of them as a particular means of approximating the noncomputable Kolmogorov [14] notion of information complexity which is the basis of Solomonoff's methods. Kolmogorov's notion is that the complexity of a set of data is defined as the length of the shortest universal Turing machine program that will generate the data. The notion is well-defined, but not practically computable. Chaitin [7] has recognized the elusiveness of theoretical universal Turing machine programs, and has proposed measuring the length of source programs in a special dialect of Lisp as a practical alternative.

The importance of the work of Kolmogorov, Solomonoff, Chaitin [6] and others tying together information theory and probability theory, and linking the various philosophic principles into a uniform idea of inductive reasoning cannot be overstated. (See Li and Vitányi [15] and Cover, *et al* [8] for reviews of this work.) While of great theoretical importance, and useful practically as a way of thinking about these issues, these ideas are immensely difficult to apply to real problems.

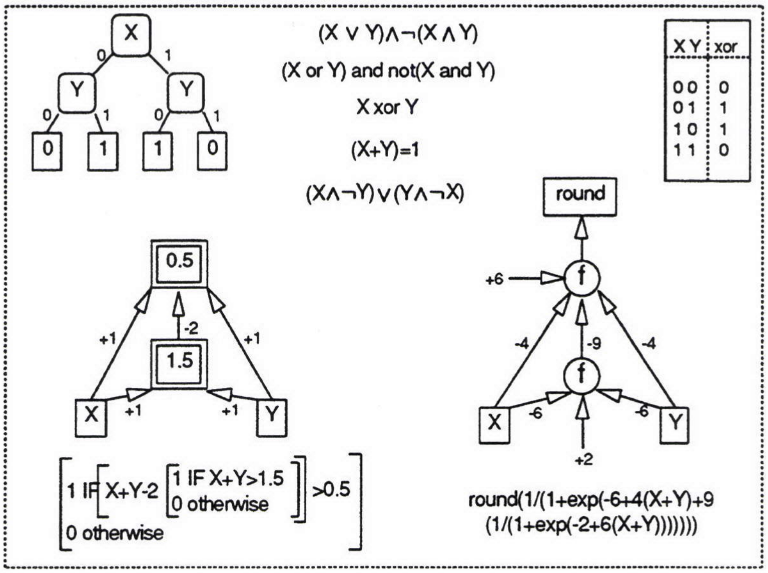
Figure 1 shows several ways of representing the same simple categorization (the exclusive-or of two binary inputs). This illustration plainly shows that the apparent complexity depends strongly on the representation (or implementation) language (see also Schaffer [20] on this point). This *apparent* complexity is deceiving:

- Ultimately, any classifier can be expressed as either a Boolean formula or a decision tree, interchangeably (because categories are discrete and mutually exclusive).
- The Boolean formula $(XxorY)$ appears simpler than $(X \wedge \neg Y) \vee (\neg X \wedge Y)$, but they might actually have equal execution times, depending on the hardware or software elements used for implementation.
- The neural network at the lower right of Figure 1 can also be represented as a decision tree with only one decision point and two leaf nodes. The internal calculation of the decision node would, however, be the nasty formula shown below the network. It is not only neural networks that involve complex math. Linear and quadratic discriminant classifiers also involve such computations. In principal, the decision nodes of a decision tree (or the conjuncts of a Horn clause) can involve arbitrarily complex computations. That they commonly do not is a design decision, not a necessity. In fact, limiting these to very simple tests, such as $X \leq c$, can seriously compromise their ability to express complex concepts.

A very apt metaphor for this situation is looking at an algorithm expressed as a flow chart, or modular breakdown diagram, or the source code, versus the executable. The diagrams and source code are abstractions, often deliberately obscuring details to gain (human) comprehensibility. What finally matters is the size and speed of the executable implementation on the target machine. How are time and space to be combined into a single measure? Should they be combined?

As noted earlier, there are several information theoretic schemes for combining the space and time complexity in a standardized manner. The reference Turing machine encoding of Prolog clause models proposal of Muggleton, *et al* [16, appendix, pp 344-346] is typical. Fundamental to these schemes is the assumption that the representation can be reduced to a set of primitive symbols, all of which have the same complexity. It is not at all clear or unequivocal what the complexity measures of transcendental functions (sin, exp, etc.) are relative to, say, polynomials of finite degree (much less to simple binary variables). Whatever they are, there is no reason to believe that they are captured by a Huffman (inverse frequency of symbol occurrence) encoding, or that a tabulation of the functions and bindings codes captures the 'proof complexity' of execution. MDL

Figure 1: Representations of $(XxorY)$



and related approaches have had success, however, in domains where the ground symbols do all have similar complexity (involve only simple logical and integer operations.

Prgamatically, all of this effort to find a theoretic combination of space and time complexity applicable to all the various representation styles in Figure 1 seems hard to justify and may be unnecessary:

1. The choice between a decision tree, a set of Horn clauses (rules), a set of discriminant functions, a neural network, *etc.* (and even the choice of which of these are even to be considered) is likely to be made on other, more subjective, grounds at any rate. Besides personal prejudices, a common consideration is the *perceived* complexity or opaqueness of the representation. Most users are extremely reluctant to rely on classifier systems whose representations they cannot understand, regardless of reported theoretical advantages. This is particularly true when the supposed advantages are expressed in terms as arcane (to most users) as the input tape of a reference Turing machine.

2. An analyst choosing between alternative classifiers is concerned with the cost-performance of actual implementations, not theoretical ones. If different platforms are involved, their cost, physical size, and power requirements, *etc.* may be more important than the computational complexity. Commonly, the platform has been previously specified based on those other grounds, and then it is a matter of comparing complexity of the actual implementations on that platform. In an age of cheap memory, 32-bit machines, and virtual memory, space is a secondary consideration. The impact of excessive memory requirements is, at any rate, reflected through the additional run-time requirements attendant on paging.

Table 1 illustrates many of the points made above. Note that the file sizes bear no necessary relationship to apparent complexity or to response time; using the primitive (hardware) XOR operator has a 5 : 1 speed advantage; the more efficient compiler has nearly a 2 : 1 advantage, increased to 3 : 1 using hardware floating point, and to 15 : 1 using hardware floating point for the nonlinear neural net. Pragmatically, the choice among these competing implementations would

Table 1: Implementations of *xor* compared

Model	Source File Sz (bytes)	Object File Sz (bytes)	.exe/.com File Sz (bytes)	Elapsed Time $10k \times 4$ Items Classified (sec)
Pascal Compiler, Software Floating Point				
$(X \vee Y) \wedge \neg(X \wedge Y)$	451	—	11,443	.49
X xor Y	266	—	11,303	.11
Linear ANN	428	—	11,471	1.99
Nonlinear ANN	419	—	11,529	102.42
C++ Compiler, Hardware Floating Point				
$(X \vee Y) \wedge \neg(X \wedge Y)$	348	661	6,531	.23
X xor Y	200	451	6,340	.05
Linear ANN	280	830	13,794	.71
Nonlinear ANN	346	1,068	16,042	9.62

be made based solely on run-time, and the space requirement would be considered only to break a tie or in the unlikely event that an implementation exceeded the address space of the targeted machine.

5 Summary

- Attempts to consolidate misclassification cost or rate and model complexity founder on issues of constraint satisfaction, trading dollars for complexity bits, lack of sampling statistics and suitable significance tests, and encoding complexity across different families of classifiers. Therefore, a generalized version of CART's 1-SE rule [5] is recommended.
- Information-theoretic measures of complexity are not practically computable, except within severely restricted families of classifiers. These measures are not useful for comparisons across families (*e.g.*, neural nets *vs.* the usual CART-style decision trees). The pragmatic measure of average response time on a target platform is recommended for expressing complexity.

References

[1] T. W. Anderson and S. L. Sclove. *The Statistical Analysis of Data.* The Scientific Press, Palo Alto, 2nd edition, 1986.

[2] A. R. Barron and T. M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory,* 37:1034–1054, 1991.

[3] J. V. Beck and K. J. Arnold. *Parameter Estimation in Engineering and Science.* John Wiley & Sons, New York, 1977.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters,* 24:377–380, 1987.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1984.

[6] G. J. Chaitin. *Algorithmic Information Theory.* Cambridge University Press, New York, 1987.

[7] G. J. Chaitin. Lisp program-size complexity, parts 1-4. *Applied Mathematics and Computation,* 49,52:(49)79–93, (52)103–126, 127–139, 141–147, 1992.

[8] T. M. Cover, P. Gacs, and R. M. Gray. Kolmogorov's contributions to information theory and algorithmic complexity. *Annals of Probability*, 17:840–865, 1989.

[9] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A*, 222:309–368, 1933.

[10] M. P. Georgeff and C. S. Wallace. A general selection criterion for inductive inference. In T. O'Shea, editor, *ECAI-84: Advances in Artificial Intelligence*, pages 473–482, Amsterdam, 1984. Elsevier.

[11] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[12] E. T. Jaynes. On the rationale of maximum entropy methods. *IEEE Proceedings*, 70:939–952, 1982.

[13] D. Kibler and P. Langley. Machine learning as an experimental science. In *Proceedings of the 3rd European Working Session on Learning*, London, 1988. Pittman.

[14] A. N. Kolmogorov. The logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, IT-14:662–664, 1968.

[15] M. Li and P. M. B. Vitányi. Inductive reasoning and Kolmogorov complexity. *Journal of Computer and Information Sciences*, 44:343–384, 1992.

[16] S. Muggleton, A. Srinivasan, and M. Bain. Compression, significance and accuracy. In *Proceedings of the 9th International Workshop on Machine Learning (ML-92)*, pages 338–347, San Mateo, CA, 1992. Morgan Kaufmann.

[17] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[18] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1982.

[19] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.

[20] C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10:153–178, 1993.

[21] R. J. Solomonoff. A formal theory of inductive inference, parts 1 and 2. *Information and Control*, 7:1–22,224–254, 1964.

[22] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[23] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 1:185–195, 1968.

[24] C. S. Wallace and P. R. Freeman. Estimation and inference by compact encoding. *Journal of the Royal Statistical Society, Series B*, 49:223–265, 1987.

[25] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods From Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann, San Mateo, CA, 1991.