From:

Computational Learning Theory and Natural Systems,

Chapter 18, "Cross-validation and Modal Theories", Vol. 3, MIT Press, 1995

Cross-Validation and Modal Theories

Timothy L. Bailey

Charles Elkan

Department of Computer Science and Engineering

University of California, San Diego[1]

October 1993

ABSTRACT: Cross-validation is a frequently used, intuitively pleasing technique for estimating the accuracy of theories learned by machine learning algorithms. During testing of a machine learning algorithm (FOIL) on new databases of prokaryotic RNA transcription promoters which we have developed, cross-validation displayed an interesting phenomenon. One theory is found repeatedly and is responsible for very little of the cross-validation error, whereas other theories are found very infrequently which tend to be responsible for the majority of the cross-validation error. It is tempting to believe that the most frequently found theory (the "modal theory") may be more accurate as a classifier of unseen data than the other theories. However, experiments showed that modal theories are not more accurate on unseen data than the other theories found less frequently during cross-validation. Modal theories may be useful in predicting when cross-validation is a poor estimate of true accuracy. We offer explanations

[1]For correspondence: Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California 92093-0114, (619) 534-8187, tbailey@cs.ucsd.edu.

for these phenomena based on PAC learning theory.

One goal of machine learning algorithms is to learn rules for classifying unseen examples from sets of training examples. A training set is generally much smaller than the total number of possible examples and may be noisy as well. An obviously desirable quality for the learned rules is that they perform well (*i.e.* "generalize") on unseen examples. The generalization accuracy of the learned rules must be estimated. One popular and intuitively attractive method for estimating the accuracy of learned rules on unseen examples is cross-validation [Breiman *et al.*, 1984].

Cross-validation repeatedly splits the training examples into two subsets and learns a theory (synonymously, a rule) from the first subset and tests the theory ("cross-validates" it) by seeing how well it classifies the other subset. It can happen that many of the theories learned during cross-validation are identical. (Two theories are identical if they classify each possible example in the same way.) When this occurs, the most frequently occurring theory may be referred to as the "modal theory." Modal theories occurred frequently when cross-validation was used to estimate the accuracy of theories learned by FOIL on datasets of examples of RNA transcription promoters.

This paper reports on experiments to explore the properties of modal theories. An initial hypothesis that they might be more accurate than the other theories learned during cross-validation turned out to be false but to shed some light on cross-validation. Modal theories do appear to be useful in predicting when cross-validation estimates may be biased.

Section 18.1 describes cross-validation in more detail. Section 18.2 talks more about modal theories. Section 18.3 describes the RNA transcription promoter datasets. Section 18.4 describes FOIL, the learning algorithm used in the experiments. Section 18.5 gives descriptions of the experiments on modal theories and their results. Section 18.6 offers explanations for the results.

1 Cross-validation

Cross-validation is a method for estimating the true error rate of a classification rule. True error rate—the probability that the rule will incorrectly classify an example drawn randomly from the same distribution (F) as the training set—is probably the most commonly used criterion for the goodness of a rule. True error rate is sometimes referred to as the generalization error rate of the rule since it measures how well the rule is expected to generalize to examples some or most of which the learning algorithm did not see.

To quantify the concept of true error rate, we make the following definitions which are essentially the same as those in [Efron, 1983]. An example $x = (t, y)$ where t stands for the features of the example and y is the true classification of the example. Suppose a learning algorithm constructs prediction rule $\eta(t, \mathbf{X})$ from training set $\mathbf{X}$. Let $\eta_i = \eta(t_i, \mathbf{X})$ be the prediction on example x_i and let $Q[y_i, \eta_i]$ be the error of the learned rule on that example. We define Q as

$$\begin{aligned} Q[y_i, \eta_i] &= 0 \text{ if } \eta_i = y_i \\ &= 1 \text{ if } \eta_i \neq y_i \end{aligned} \tag{1}$$

We then define the true error rate (Err) as the probability of incorrectly classifying a randomly selected example $X_0 = (T_0, Y_0)$, that is, the expectation

$$Err = E_F\, Q[Y_0, \eta(T_0, Y_0)] \tag{2}$$

Cross-validation estimates Err by reserving part of the training set for testing the learned theory. In general, v-fold cross-validation (randomly) splits the training set into v equal-sized subsets, trains on $v - 1$ of them and tests on one of them. Each subset is used once as the test set (*i.e.* is left out of the training set.) A common choice for v is the size of the original training set. Since each subset contains one element, this is called "leave-one-out" (or n-fold) cross-validation. The experiments described later in this chapter all use "leave-one-out" cross-validation.

The "leave-one-out" cross-validation estimate of Err is defined as

$$\hat{E}rr_{(CV)} = 1/n \sum_{i=1}^{n} Q[y_i, \eta(t_i, \mathbf{x}_{(i)})]$$

where $\mathbf{x}_{(i)}$ is the training set with x_i removed and $\eta(t_i, \mathbf{x}_{(i)})$ is the rule learned from $\mathbf{x}_{(i)}$.

For a given amount of training data, leave-one-out cross-validation allows the learning algorithm to learn on the largest possible number of examples, while still providing a means of estimating the accuracy of the learned rules on unseen data. Intuitively, the true error of concepts learned on $n-1$ examples during leave-one-out cross-validation should be close to what we are trying to estimate—the true error of the theory learned on a particular n examples. By this argument, other methods which use smaller subsets for training can be expected to be poorer estimates of Err when the number of training examples available is small. v-fold cross-validation where $v < n$ falls into this category, as does the random-split method discussed in [Kononenko and Bratko, 1991].

2 Modal theories

Each time v-fold cross-validation is used to estimate the error of a theory learned by a learning algorithm from a dataset, the learning algorithm is run v times. This results in v "different" theories, some or all of which may actually be identical in terms of how they classify all possible examples. The modal theory of a cross-validation run is defined as the theory found most frequently by the learning algorithm during the run. The frequency of the modal theory is the proportion of times it is found during the cross-validation run.

It is often difficult to determine if two theories are identical if they are not syntactically identical. The learning algorithm described in this chapter, FOIL, often produces theories during cross-validation which are syntactically identical. Syntactic identity is very easy to test for, and is the actual measure used for identity in the experiments described below. In cases where at least 50% of theories found during cross-validation are syntactically identical, it is clear that the syntactically modal theory must be the modal theory. When the syntactically modal theory has a frequency lower than 50%, there is some chance that it is not the

Name	*Class*	*Sequence*
p001	promoter	GATGCGGCGCAAAATCGGAAATGGAGATG ...
n001	nonpromoter	ATCGCAACAAGGCAACCAAAACGAGACTC ...
p002	promoter	GAACAGCCGGCACCGGAGTTGGCATCAAT ...
n002	nonpromoter	CCGGCTGTTGATTCCGATTCGGTGGCAAT ...
p003	promoter	GGCGAAGCCAGCTCTTGAGCCGTGATAAA ...
n003	nonpromoter	ACCATTGATTTTGGGCCTCAGTTGGGAGC ...

Table 1: Excerpt from a typical promoter dataset.

modal theory, but we ignore this possibility in what follows. In most cases, the syntactically modal theory had a frequency near or above 50%.

3 The promoter datasets

Several researchers have used the problem of recognizing RNA transcription promoters in DNA sequences as a test of various learning algorithms [Towell *et al.*, 1990]. An RNA transcription promoter is a section of a DNA molecule which binds with a particular protein which "promotes" the transcription of a gene from DNA to RNA. DNA molecules consist of sequences of four different nucleotides called bases. The bases are abbreviated by the letters A, C, G and T. A section of a DNA molecule can thus be represented as a string over the alphabet "A C G T". Table 1 shows a portion of a typical promoter dataset.

We have prepared a number of new data sets by extracting sequences from a database of eukaryotic promoters [Bucher, 1991]. Our data sets each consist of either promoters from a single species or promoters from genes encoding the same product (*i.e.* enzyme or protein.) We used all of the promoters in the eukaryotic promoter database for each species or gene-product where all bases in the DNA sequence were known from 300 base positions upstream to 5 positions downstream from the start of transcription.

Database	*Positive*	*Negative*
G. Gallus	46	34
R. Norvegicus	70	60
M. Musculus	127	97
chorion	37	37

Table 2: The composition of promoter databases.

Three of the datasets consist of promoters from a single species and one consists of promoters from genes known for one protein family. Each example is a string over the alphabet "A C G T" of length 106, labeled with its class (either "promoter" or "non-promoter". The string for promoters is positions -100 to $+6$ relative to the start of transcription of a known gene. Negative examples consist of 106 characters of DNA from -300 to -195 from one of the same known genes. In constructing the datasets, all of the genes of the particular type present in the database were used except genes which contained unknown bases in any of the positions used in the training examples. Table 2 lists the number of positive and negative examples in each of the datasets.

4 The learning algorithm—FOIL

The FOIL [Quinlan, 1990] machine learning algorithm learns from data encoded as relations and outputs concepts in the form of simple Datalog programs. Precisely, the output format is sets of function-free Datalog-with-negation clauses. FOIL is given input relations defined extensionally as lists of "ground" tuples.[2] One or more of the relations is designated as the "target" relation, and FOIL attempts to learn an intensional definition for it in terms of the

[2]A ground tuple is a tuple of constants. FOIL requires the user to define enumerated data types. Each relation has a name and a list of the data types of its arguments. For example:

$$can-get-to(node, node)$$

relations it knows about.

For example, FOIL might be given the relations *linked-to*(X, Y) and *can-get-to*(X, Y) defined extensionally and asked to find an intensional definition of *can-get-to*(X, Y). If the extensional definitions of the two relations are such that *can-get-to*(X, Y) is the transitive closure of *linked-to*(X, Y), FOIL may succeed in finding the extensional definition

$$\begin{aligned} \textit{can-get-to}(X, Y) &\leftarrow \textit{linked-to}(X, Y) \\ \textit{can-get-to}(X, Y) &\leftarrow \textit{linked-to}(X, Z), \textit{linked-to}(Z, Y) \end{aligned}$$

FOIL uses a greedy algorithm that builds rules one clause at a time. Each clause is built one literal at a time, trying all possible variablizations of each possible relation, and adding the one with the highest "information gain" to the clause. There is some limited within-clause backtracking. That is, if no literal can be found with positive information gain, the algorithm backs up by removing the last literal added to the clause and replacing it with another candidate with positive (but lower) gain.

There are many possible encodings for a given classification problem. The particular one chosen can greatly affect the efficiency of FOIL and whether or not a rule is found at all. In addition, background knowledge not present in the examples can be provided in the form of additional relations that can be used in forming the rule.

The promoter training data must be converted into relations before it can be used by FOIL. Recall that the promoter datasets consist of examples which have three fields: name, class and sequence. The simplest way to encode the data for FOIL is as a single relation over three attributes: *sequence*(*name*, *position*, *base*). Here *name* is a data type which ranges over the names of the examples, *position* is a data type that ranges over the positions in the sequences (-100 to $+6$) and *base* ranges over the letters A, C, G and T. Each example

defines a relation of two arguments, both of which are of type *node*, where the data type *node* must be enumerated by the user. For example:

$$node : n1, n2, n3, n4, n5$$

specifies that the data type *node* can only take the values $n1$, $n2$, $n3$, $n4$ or $n5$. When FOIL learns a definition for a relation, it learns it in terms of literals which contain *no* constants, only variables.

in a promoter dataset becomes a set of tuples in the *sequence* relation—one tuple for each position in the sequence.

Unfortunately, due to intrinsic limitations of the algorithm, FOIL is not able to discover any concepts when promoter data is encoded is a single sequence relation. To overcome this, the data is instead encoded as a set of unary relations, one for each possible combination of *Position* and *Base*. These relations are named $1\_A(Name)$, $1\_C(Name)$, $1\_G(Name)$, $1\_T(Name)$, $2\_A(Name)$ and so on. The promoter "ACGCG" with name "human", for example, is encoded as

$$\begin{array}{l} 1\_A(human) \\ 2\_C(human) \\ 3\_G(human) \\ etc. \end{array}$$

Here, "human" is a constant, not a variable.

With this encoding of the promoter datasets, FOIL learns theories of the form:

$$\begin{array}{rcl} promoter(X) & \leftarrow & 25\_A(X), 26\_G(X), 31\_C(X) \\ promoter(X) & \leftarrow & 5\_G(X), 6\_A(X) \\ & \vdots & \end{array}$$

This theory means "X is a promoter if the the letter A occurs at position 25, the letter G occurs at position 26 and the letter C occurs at position 31, or if the letter G occurs at position 5 and the letter A occurs at position 6." It turns out that the clauses learned by FOIL given this encoding of the promoter data never contain any other variables than X, so the theories it learns are really propositional—one can ignore the X in $26\_G(X)$. So, in the experiments described in this chapter, FOIL is really learning propositional theories in disjunctive normal form.

5 Experiments on modal theories

The experiments described in this section investigate three questions.

- Are modal theories more accurate on the "left-out" example during "leave-one-out" cross-validation?
- Are modal theories more accurate than non-modal theories?
- Is the frequency of the modal theory correlated with the bias of the cross-validation estimate of error?

All the experiments used "leave-one-out" cross-validation as the estimate of error, FOIL as the learning algorithm, the four promoter datasets as the learning samples and the encoding of the DNA data for FOIL described in the previous section. In all cases, FOIL returned theories with extremely low (usually zero) apparent error.[3]

5.1 Modal theories are more accurate on the "left-out" example

"Leave-one-out" cross-validation with FOIL was run on the four promoter databases. The cross-validation estimate of error was computed, *i.e.* the number of times the learned theory was wrong on the left-out example divided by total number of examples. In addition, the fraction of times the (syntactically) modal theory was wrong on the left-out example was recorded, and the same was done for the non-modal theories.

The results of this experiment are shown in Table 3. FOIL was able to learn rules for recognizing promoters from each database. Except for the chorion database, the cross-validation estimate of error was quite high. The frequency of the (syntactically) modal theory was as high as 87.8%, and always above 42%. The modal theory always did extremely well at correctly predicting the class of the left-out example, while the non-modal theories were always wrong more than half the time. The table also shows the mean apparent error of the

[3]The apparent error of a theory is defined as the fraction of training set examples that the theory classifies incorrectly.

Database	*CV Error*	*Modal Freq.*	*Error on Left-Out Example*		*Modal Theory*
			Modal Theory	*Non-Modal Theories*	*Mean Apparent Error*
G. Gallus	30.0%	58.7%	2.1%	69.7%	2.5%
R. Norvegicus	33.8	42.3	0.0	58.7	0.8
M. Musculus	32.1	43.3	0.0	56.7	0.0
chorion	8.1	87.8	0.0	66.7	2.7

Table 3: FOIL frequently found a modal theory on the full promoter databases, and this theory was usually correct on the left-out example.

modal theory, which shows how well the modal theory tended to fit the training sets from which it was learned. In all cases, the modal theory was incorrect on at most two examples in the complete dataset.

5.2 Modal theories are not more accurate

The high accuracy of the modal theory on the left-out example encourages the (incorrect) hypothesis that the modal theory generalizes better on unseen data than the less frequently found theories. If this were true, then it would be profitable to always run FOIL several times on subsets of the training data until a modal theory emerged and use the modal theory as the final output of the learning algorithm. This approach might apply to other machine learning algorithms besides FOIL.

To confirm this hypothesis, it would be necessary to know the actual value of *true error* for theories learned by FOIL. Unfortunately, this is not possible (since the correct theory is not known.) To circumvent this difficulty, "test set estimation" was used to give an estimate of the error of the modal theory. Each dataset was randomly split into a training set and a test set of about equal size by placing each example from the dataset into either the test set with probability 1/2 or into the training set with probability 1/2. Then, cross-validation of

Database	*Modal Theory Test Error*	*Mean Test Error*
G. Gallus	40.0%	40.0%
R. Norvegicus	40.9	40.1
M. Musculus	35.4	33.8
chorion	12.5	11.6

Table 4: Modal theories are no better than average.

FOIL was done using just the examples in the training set. The test set was used afterwards to estimate the *true error* of the modal theories. The other (non-modal) theories were also tested on the test set. The fraction of examples in the test set correctly classified by a theory is called its "test error."

Modal theories, it turns out, are no better at generalizing to unseen data than non-modal theories, judging from the performance of the modal theories on the test set. The error of the modal theories on the test sets was often worse than the average error on the test sets of the all theories found during a run of cross-validation. Table 4 shows the results. It lists the test error of the modal theory and the average value of test error of all theories found during cross-validation.

To further investigate this phenomenon, FOIL was run on the full training sets (no example was left out) and the (single) theory learned was tested on the test sets for each database. The error of the theories generated in this way turned out, in every case, to be identical to the Modal Theory Test Error shown in Table 4. The reason proved to be that, for each database, the modal theory found during cross-validation on the training set was the *same as the theory found by* FOIL *on the entire training set.* These results will be further discussed in the discussion section of this chapter.

5.3 The frequency of the modal theory is correlated with the bias of cross-validation

Earlier experiments seemed to indicate a tendency of cross-validation to be downwardly biased as an estimate of *true error* when the frequency of the modal theory was high. (That is, cross-validation seemed to *underestimate* the *true error.*) To investigate this possibility, ten experiments were run on each promoter database. In each experiment, the dataset was randomly split into two (approximately) equal subsets as described in the previous subsection, and cross-validation was run on the first subset. Then, the first subset was used again to learn a single theory. Finally, that theory was tested on the second subset to measure test error as described above.

Each such experiment provided a single data point consisting of the cross-validation estimate of error of a theory learned on a dataset, the test error of the theory, and the frequency of the modal theory learned on the dataset. Figure 1 shows the results of all 40 experiments (ten experiments on each of the four datasets) plotted as cross-validation error minus test error vs. modal theory frequency. The correlation is reasonably strong (-0.41). If we assume that cross-validation error minus test error is an estimator of the bias of cross-validation, as the modal frequency increases, the (estimate of) bias goes from upward to downward. In other words, when the frequency of the modal theory is high, cross-validation tends to *underestimate* the *true error.* Conversely, when the modal theory frequency is low, cross-validation tends to *overestimate* the *true error.*

It is interesting to look at the results of the previous experiment individually for each of the datasets. These are displayed in Figure 2. One notices immediately that the tendency of cross-validation to underestimate *true error* increases with modal theory frequency for each of the datasets. The effect is consistent across all of the datasets. It is interesting to note that for the two smallest datasets, G. gallus (80 samples) and chorion (74 samples), most of the points lie above the line $y = 0$. This means that, for small datasets, cross-validation seems to be biased upwards (overestimate *true error.*) On the other hand, cross-validation appears to be unbiased or biased slightly downward for the largest databases M. musculus

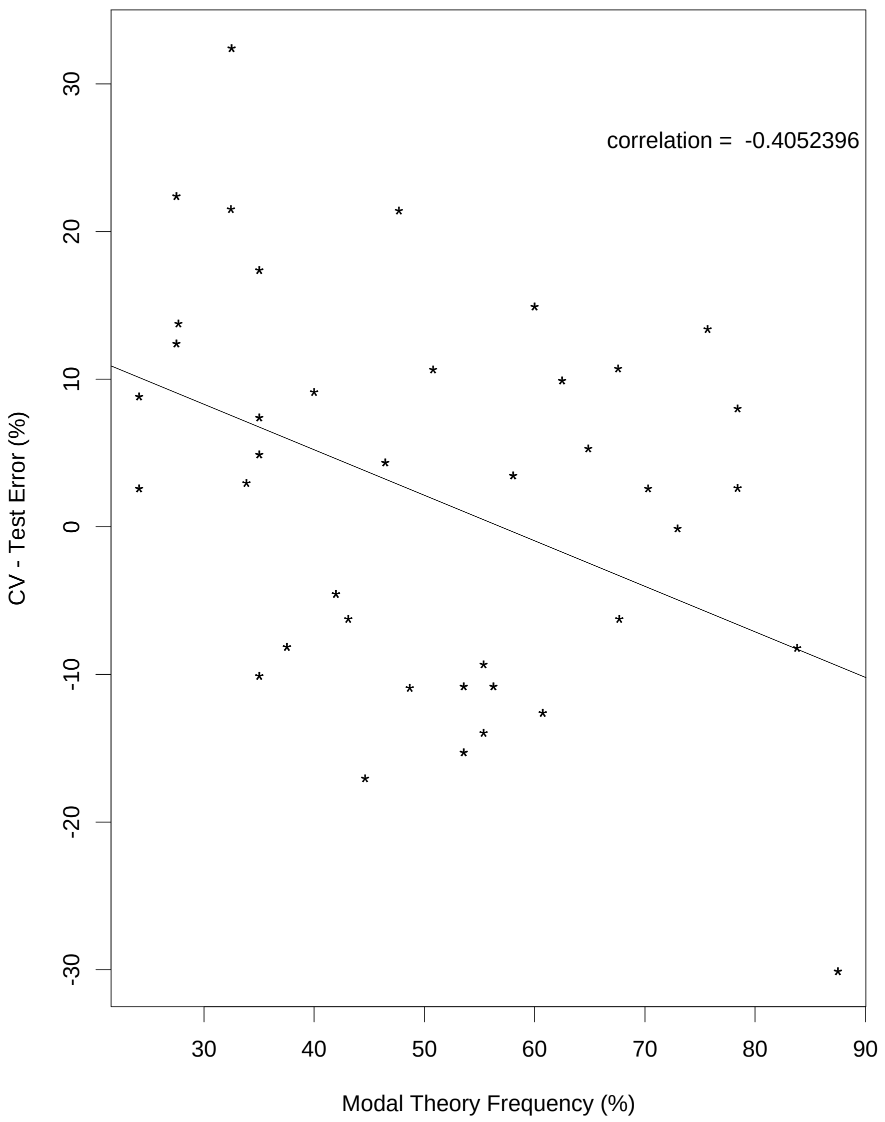

Figure 1: Scatter plot, least-squares fit and correlation coefficient of Modal Theory Frequency versus Estimate of Bias of Cross-validation for 10 trials on each promoter dataset. All 40 trials are combined.

(224 samples) and R. Norvegicus (130 samples.) An explanation for this effect is offered in the discussion section of the chapter.

6 Discussion

We investigated the accuracy of theories learned by FOIL on databases describing RNA transcription promoters. We discovered that, during leave-one-out cross-validation, FOIL often discovers the same theory repeatedly. We then showed that when this occurs, the cross-validation error may be much lower than the test error measured on a test set of unseen examples. So, high modal theory frequency appears to be a warning that cross-validation error may be misleading and other error measures should be used. Modal theories do not appear to be useful for improving the accuracy of learning algorithms, but do appear useful for determining when cross-validation error estimates may be erroneous.

The tendency of modal theories learned during "leave-one-out" cross-validation to do well on the left-out example can easily be explained. In the experiments described here, the learning algorithm produced theories which tended to be nearly perfect at classifying the training set examples. (In other words, the learned theories had little or no apparent error.) During "leave-one-out" cross-validation, the union of any two datasets contains all of the original examples. So, if the same theory is found twice (or more) and has no apparent error each time, it must be perfectly consistent with the entire dataset. This was the case with the M. musculus dataset in Table 3. The modal theory is found 97 times and fits the training set perfectly each time, so it must fit the entire dataset and have a CV error estimate of 0.0%.

A similar argument can be made when the modal theory is found many times and makes only 1 or 2 errors on the training set each time. This was the case for the other three datasets shown in Table 3. When a learning algorithm is able to achieve low apparent error, it is inevitable that modal theories discovered during cross-validation will tend to fit the dataset very well. This explains the high accuracy of modal theories on the left-out examples, as

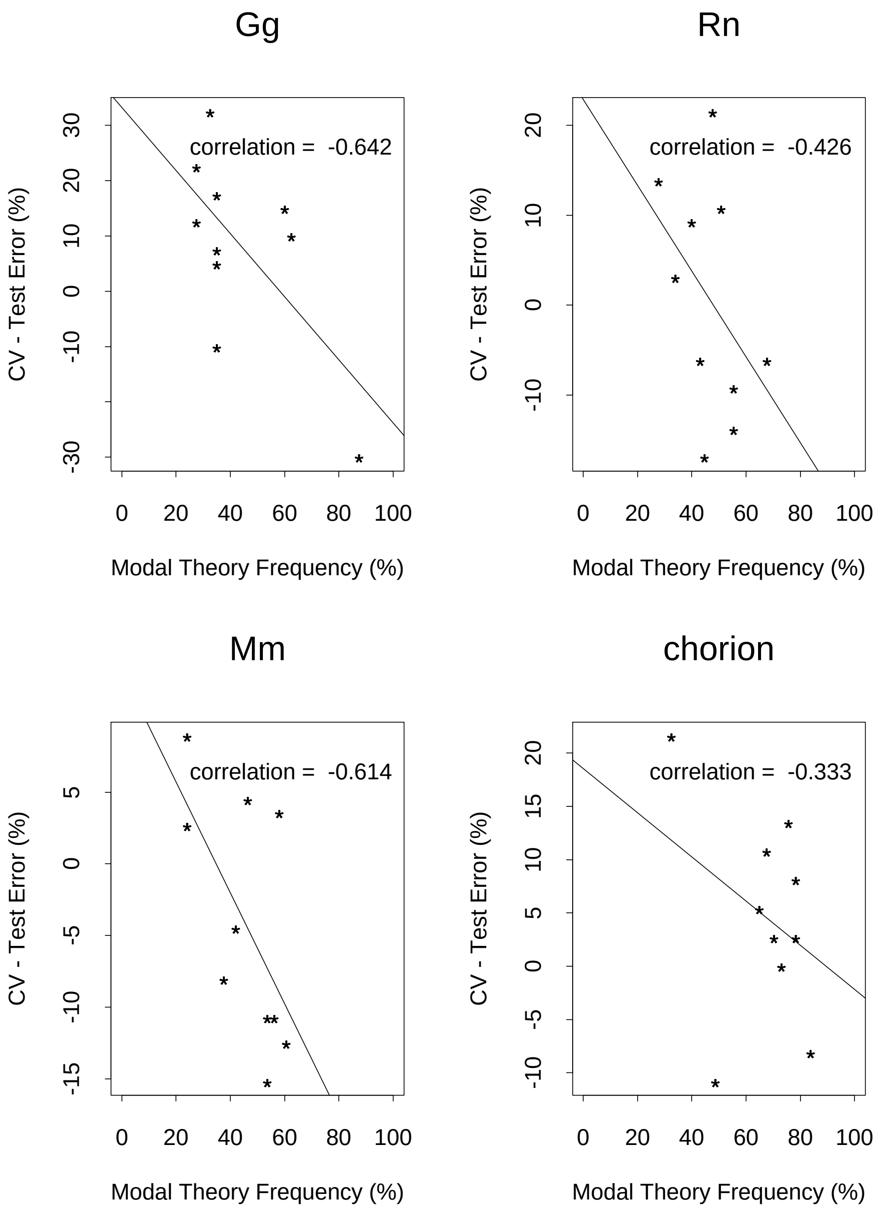

Figure 2: Scatter plot, least-squares fit and correlation coefficient of Modal Theory Frequency versus Estimate of Bias of Cross-validation for 10 trials on each promoter dataset.

well as why the modal theory tends to be identical with the theory on the entire dataset.

The last experiment raised the possibility that the frequency of the modal theory may be correlated with the bias of cross-validation as an estimate of *true error.* With all four promoter datasets, cross-validation tended to overestimate *true error* (or, at least, its proxy, test error) when the modal theory frequency was low and to underestimate it when the modal theory frequency was high.

To explain the fact that the cross-validation estimate of error is too high when the modal theory frequency is low, observe that cross-validation estimates the error of a theory learned from n samples by learning theories from $n-1$ samples. It is reasonable to assume that, in general, theories learned from n samples have lower *true error* than theories learned from $n-1$ samples. One might expect, therefore, that cross-validation would be biased upward when the samples size n is small. We noticed in the previous section that, for the smaller datasets, cross-validation tended to be biased upwards, whereas it was fairly unbiased for the larger datasets. This effect may explain some of the bias observed when the modal theory frequency is low.

To explain the fact that cross-validation underestimates *true error* when the modal theory frequency is high, we observe that, in all the experiments, the apparent error of the theories learned was extremely low—usually zero. The learning algorithm is able to find a theory that perfectly fits the dataset even though the theories it comes up with may be highly inaccurate.[4] As was mentioned earlier, when a theory with zero apparent error is found twice during cross-validation, it must perfectly fit the *entire* dataset. So, it is always correct on the left-out example and does not contribute to the cross-validation error estimate. This means that as the frequency of the modal theory approaches 1, the cross-validation error estimate will approach the apparent error rate, which will be almost zero (for algorithms like FOIL). Under these conditions, cross-validation will underestimate *true error.*

It seems wise, then, to consider high modal theory frequency as a warning flag that a learning algorithm may be finding a spurious theory. On the other hand, the correlation between modal theory frequency and bias is far from perfect. Modal theory frequency may be high precisely because the learning algorithm is finding an accurate theory.

[4]The hypothesis space searched by FOIL with the encoding used is essentially disjunctive normal form boolean formulas (DNF). The Datalog theories discovered by FOIL on the promoter databases tended to consist of at least $s = 2$ clauses each with at least $k = 2$ literals. So, the hypothesis space has Vapnik-Chervonenkis dimension (VCdim) as defined in [Haussler, 1988] of at least

$$\mathrm{VCdim}(H) \geq \left\lfloor ks \log \frac{n}{ks^{1/k}} \right\rfloor .$$

Since the number of attributes is $n = 106$, this gives a lower bound for VCdim of approximately 20. The sample complexity of FOIL on this problem is then given in [Haussler, 1988] as

$$(4 \log(2/\delta) + 8\,\mathrm{VCdim}(H) \log(13/\epsilon))/\epsilon ,$$

So for $\epsilon = \delta = .1$, the sample complexity is at least 11,370 samples, far more than the actual sample sizes of (order) 100 samples.

References

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

[Bucher, 1991] Philipp Bucher. *The Eukaryotic Promoter Database of the Weizmann Institute of Science. EMBL Nucleotide Sequence Data Library Release 29.* Weizmann Institute of Science, 1991.

[Efron, 1983] Bradley Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, June 1983.

[Haussler, 1988] David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[Kononenko and Bratko, 1991] Igor Kononenko and Ivan Bratko. Information-based evaluation criterion for classifiers' performance. *Machine Learning*, 6(1):67–80, January 1991.

[Quinlan, 1990] John R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[Towell *et al.*, 1990] G. G. Towell, J. W. Shavlik, and Michiel O. Noordewier. Refinement of approximate domain theories by knowledge-based artificial neural networks. In *Proceedings of the National Conference on Artificial Intelligence*, pages 861–866, 1990.