Process-Oriented Estimation of Generalization Error

Pedro Domingos
Artificial Intelligence Group
Instituto Superior Técnico
Lisbon 1049-001, Portugal
pedrod@gia.ist.utl.pt
http://www.gia.ist.utl.pt/~pedrod

Abstract

Methods to avoid overfitting fall into two broad categories: data-oriented (using separate data for validation) and representation-oriented (penalizing complexity in the model). Both have limitations that are hard to overcome. We argue that fully adequate model evaluation is only possible if the search process by which models are obtained is also taken into account. To this end, we recently proposed a method for *process-oriented evaluation (POE)*, and successfully applied it to rule induction [Domingos, 1998b]. However, for the sake of simplicity this treatment made a number of rather artificial assumptions. In this paper the assumptions are removed, and a simple formula for error estimation is obtained. Empirical trials show the new, better-founded form of POE to be as accurate as the previous one, while further reducing theory sizes.

1 Introduction

Overfitting avoidance is a central problem in machine learning. If a learner is sufficiently powerful, whatever representation and search methods it uses, it must guard against selecting a model that fits the training data well but captures the underlying phenomenon poorly. Current methods to address this problem fall into two broad categories. *Data-oriented evaluation* uses separate data to learn and validate models, and includes methods like cross-validation [Breiman *et al.*, 1984; Stone, 1974], the bootstrap [Efron and Tibshirani, 1993], and reduced-error pruning [Brunk and Pazzani, 1991]. It has several disadvantages: it is often computationally intensive, reduces the data available for learning, can be unreliable if the validation set is small, and is itself prone to overfitting if a large number of models is compared [Ng, 1997]. *Representation-oriented evaluation* seeks to avoid these problems by using the same data for training and validation, but *a priori* penalizing some models. Bayesian approaches in general fall into this category (e.g., [Chickering and Heckerman, 1997]). Representation-oriented measures typically contain two terms, one reflecting fit to the data, and one penalizing model complexity (e.g., [Rissanen, 1978]). This approach is only appropriate when the simpler models are truly the more accurate ones, and there is mounting evidence that this is typically not the case [Domingos, 1998a; Jensen and Cohen, 1998]. Structural risk minimization [Vapnik, 1995; Shawe-Taylor *et al.*, 1996] and PAC learning [Kearns and Vazirani, 1994] are representation-oriented methods that seek to bound the difference between training and generalization error using a function of the model space's (effective) dimension. This typically produces bounds that are overly broad, and requires severely restricting the model space.

We believe the limitations of representation-oriented evaluation stem from ignoring the search process by which candidate models[1] are obtained. A learner with an unlimited model space can avoid overfitting as long as it attempts only a limited number of models (even if it is not possible *a priori* to predict which). Intuitively, the more search has been performed to obtain a model, the higher its expected generalization error for a given training-set error. In a recent paper [Domingos, 1998b] we made this intuition precise and applied the resulting formulas to the CN2 rule learner [Clark and Niblett, 1989], obtaining systematic improvements in generalization error and theory size. However, for the sake of simplicity the treatment in [Domingos, 1998b] made two rather artificial assumptions: that all error rates are *a priori* equally likely, and that a model's generalization error can be roughly estimated by treating all previously-generated models as having similar generalization errors. In this paper we remove these two assumptions, interpret the result, and successfully apply it to CN2.

2 Process-Oriented Evaluation

Suppose learner L_m consists of drawing m hypotheses at random (independently) from some model space, and returning the one with lowest error on a training sample S. Let $h_{m,i}$ be the ith hypothesis generated by L_m. If $h_{m,i}$'s true error rate is $\epsilon_{m,i}$ and S consists of n independently

[1] By "model" we mean model structure *and* parameter values.

drawn examples, the number of errors $e_{m,i}$ committed by $h_{m,i}$ on S is a binomially distributed variable with parameters n and $\epsilon_{m,i}$:

$$\begin{aligned} p(e_{m,i}|n,\epsilon_{m,i}) &= b(e_{m,i}|n,\epsilon_{m,i}) \\ &= \begin{pmatrix} n \\ e_{m,i} \end{pmatrix} \epsilon_{m,i}^{e_{m,i}} (1-\epsilon_{m,i})^{n-e_{m,i}} \end{aligned} \tag{1}$$

Let $B(e_{m,i}|n,\epsilon_{m,i})$ be the probability that the number of errors is greater than $e_{m,i}$:

$$B(e_{m,i}|n,\epsilon_{m,i}) = \sum_{i=e_{m,i}+1}^{n} b(i|n,\epsilon_{m,i}) \tag{2}$$

Notice that this notation is the opposite of the usual notation for a cumulative distribution function (i.e., $B(e|n,\epsilon) = 1 - \text{Binomial\_cdf}(e|n,\epsilon)$). It will be more convenient for what follows.

The probability of L_m returning a hypothesis h_m that misclassifies e_m training examples is the probability that at least one of the m hypotheses $h_{m,i}$ makes e_m errors, and all the others make e_m or more errors. Equivalently, it is the probability that all hypotheses $h_{m,i}$ make more than $e_m - 1$ errors, minus the probability that they all make more than e_m errors:

$$\begin{aligned} p(e_m|n,\vec{\epsilon}_m) &= \prod_{i=1}^{m} B(e_m - 1|n,\epsilon_{m,i}) \\ &\quad - \prod_{i=1}^{m} B(e_m|n,\epsilon_{m,i}) \end{aligned} \tag{3}$$

where $\vec{\epsilon}_m = (\epsilon_{m,1},\ldots,\epsilon_{m,i},\ldots,\epsilon_{m,m})$. By Bayes' theorem:

$$p(\vec{\epsilon}_m|n,e_m) \propto p(\vec{\epsilon}_m)\, p(e_m|n,\vec{\epsilon}_m) \tag{4}$$

Let $h_{m,c}$ be the hypothesis with lowest error (i.e., the "chosen" hypothesis, so that learner L_m returns $h_{m,c}$ and $e_m = e_{m,c}$). Our goal is to predict $h_m = h_{m,c}$'s true error rate $\epsilon_m = \epsilon_{m,c}$ from e_m. For this purpose, we marginalize Equation 4 over all the $e_{m,i}$ save $e_{m,c}$:[2]

$$p(\epsilon_{m,c}|n,e_m) \propto \int_{\vec{\epsilon}_{m\backslash c}} p(\vec{\epsilon}_m)\, p(e_m|n,\vec{\epsilon}_m)\, d\vec{\epsilon}_{m\backslash c} \tag{5}$$

where the integral is multiple, over all components of $\vec{\epsilon}_m$ save $\epsilon_{m,c}$. The expected value of $\epsilon_{m,c}$ can now be computed by integration:

$$E[\epsilon_{m,c}|n,e_m] = \frac{\int_0^1 \epsilon_{m,c}\, p(\epsilon_{m,c}|n,e_m)\, d\epsilon_{m,c}}{\int_0^1 p(\epsilon_{m,c}|n,e_m)\, d\epsilon_{m,c}} \tag{6}$$

[2]This is where we previously assumed that $\forall_i\ \epsilon_{m,i} = \epsilon_{m,c}$ and dropped the prior $p(\vec{\epsilon}_m)$.

Let:

$$f = \int_0^1 p(\epsilon_{m,i})\, b(e_m|n,\epsilon_{m,i})\, d\epsilon_{m,i} \tag{7}$$

$$F = \int_0^1 p(\epsilon_{m,i})\, B(e_m|n,\epsilon_{m,i})\, d\epsilon_{m,i} \tag{8}$$

$$E_b[\epsilon_{m,c}] = \frac{\int_0^1 \epsilon_{m,c}\, p(\epsilon_{m,c})\, b(e_m|n,\epsilon_{m,c})\, d\epsilon_{m,c}}{\int_0^1 p(\epsilon_{m,c})\, b(e_m|n,\epsilon_{m,c})\, d\epsilon_{m,c}} \tag{9}$$

$$E_B[\epsilon_{m,c}] = \frac{\int_0^1 \epsilon_{m,c}\, p(\epsilon_{m,c})\, B(e_m|n,\epsilon_{m,c})\, d\epsilon_{m,c}}{\int_0^1 p(\epsilon_{m,c})\, B(e_m|n,\epsilon_{m,c})\, d\epsilon_{m,c}} \tag{10}$$

Substituting Equation 3 into 5 and 5 into 6, using the assumption of independent hypotheses, and assuming the same prior $p(\epsilon_{m,i})$ for all hypotheses, we obtain the following expression:

$$\begin{aligned} E[\epsilon_{m,c}|n,e_m] &= \frac{f(F+f)^{m-1}}{(F+f)^m - F^m} E_b[\epsilon_{m,c}] \\ &\quad - \frac{F[(F+f)^{m-1} - F^{m-1}]}{(F+f)^m - F^m} E_B[\epsilon_{m,c}] \end{aligned} \tag{11}$$

For all but the smallest n, $F \gg f$ (Equations 7, 8, 1 and 2). Thus, using the binomial expansion of $(F+f)^m$ we obtain that $(F+f)^m - F^m \simeq mfF^{m-1}$, $(F+f)^{m-1} - F^{m-1} \simeq (m-1)fF^{m-2}$, and $(F+f)^{m-1} \simeq F^{m-1}$. Substituting these into Equation 11 and simplifying, we obtain:

$$E[\epsilon_{m,c}|n,e_m] = \frac{E_b[\epsilon_{m,c}] + (m-1)E_B[\epsilon_{m,c}]}{m} \tag{12}$$

Let $\epsilon_{m,c}^{ML} = e_m/n$ be the maximum likelihood estimate of $\epsilon_{m,c}$. For sufficiently large n, $E_b[\epsilon_{m,c}] \simeq \epsilon_{m,c}^{ML}$ (Equation 9, given a well-behaved prior $p(\epsilon_{m,c})$, i.e., as long as $p(\epsilon_{m,c}) \neq 0$ in the neighborhood of $\epsilon_{m,c} = e_m/n$). Let $\epsilon_{m,c}^{Prior} = \int_0^1 \epsilon_{m,c} p(\epsilon_{m,c})\, d\epsilon_{m,c}$ be the prior expected value of $\epsilon_{m,c}$. Suppose a beta or similarly bell-shaped prior is used [Bernardo and Smith, 1994]; this is what makes intuitive sense for error rates. In general e_m/n (the inflection point of $B(e_m|n,\epsilon_{m,c})$ as a function of $\epsilon_{m,c}$) will fall below $\epsilon_{m,c}^{Prior}$ (the peak of the prior), since e_m will tend to zero as more hypotheses are generated and the one with lowest error selected. Then, for sufficiently large n, $B(e_m|n,\epsilon_{m,c}) \simeq 1$ over the entire range where $p(\epsilon_{m,c})$ is significantly greater than zero (leaving out only the left tail of the distribution), and $E_B[\epsilon_{m,c}] \simeq \epsilon_{m,c}^{Prior}$ (Equation 10). Making these substitutions we finally obtain (omitting the c indexes, since $\epsilon_m = \epsilon_{m,c}$):

$$E[\epsilon_m|n, e_m] = \frac{\epsilon_m^{ML} + (m-1)\epsilon_m^{Prior}}{m} \quad (13)$$

This formula is quite similar to the well-known Laplace correction or m-estimate [Cestnik, 1990]. Its role for the number of hypotheses is similar to the m-estimate's role for the number of examples. The m-estimate gradually changes from the maximum likelihood estimate to the prior as the number of examples decreases; similarly, Equation 13 gradually uncovers the prior as the number of hypotheses generated increases. The intuitive meaning of Equation 13 is clear: when a learner generates a series of hypotheses and returns the one with lowest training-set error, the more hypotheses it generates the less sure we are that the observed error corresponds to the true error, and the more weight should be given to the *a priori* expected error.

This result is intuitively satisfying, because it gives a mathematical basis for increasing model uncertainty as the amount of search performed increases. However, Equation 13 as it stands is of limited practical use, because it converges very rapidly to ϵ_m^{Prior} as more independent hypotheses are generated. As a result, for all but the earliest few hypotheses, the error estimate $E[\epsilon_m|n, e_m]$ is quite insensitive to the empirical error ϵ_m^{ML}. This effect, however, is at least partly due to the fact that hypothesis dependences are being ignored, and as a result the empirical error of one hypothesis carries no information about the true error of another. In particular, only the empirical error of the chosen hypothesis carries information about its true error, resulting in the chosen hypothesis' expected error being the unalloyed prior in all *a priori* possible situations where the minimum empirical error is not the chosen hypothesis' (Equation 3). In practical learners, on the other hand, the hypotheses generated are typically very strongly dependent. Thus, in general, all the empirical errors observed will carry information about the true error of the chosen hypothesis, and Equation 13 should converge correspondingly slower to the prior term ϵ_m^{Prior}. We propose to model this by replacing m in Equation 13 by a slower-growing function of m, which can be thought of as the "effective number of independent hypotheses attempted." For example, attempting ten hypotheses with given dependences between them may be equivalent (with respect to the convergence of Equation 13 to ϵ_m^{Prior}) to attempting two independent hypotheses. Thus, Equation 13 provides a simple way of combining data-oriented, representation-oriented and process-oriented information when estimating generalization error: ϵ_m^{ML} is the data-oriented component (the model's empirical error), ϵ_m^{Prior} is the representation-oriented component (a function of the model's form), and m is the process-oriented component (a function of the search process that led to the model).

3 Application to Rule Induction

Most rule induction systems employ a set covering or "separate and conquer" search strategy [Michalski, 1983; Clark and Niblett, 1989]. Rules are induced one at a time, and each rule starts with a training set composed of the examples not covered by any previous rules. A rule is induced by adding conditions one at a time, starting with none (i.e., the rule initially covers the entire instance space). The next condition to add is chosen by attempting all possible conditions. Conditions on symbolic attributes are typically of the form $a_i = v_{ij}$, where v_{ij} is a possible value of attribute a_i. Conditions on numeric attributes are typically of the form $a_i \leq v_{ij}$ or $a_i > v_{ij}$, where the thresholds v_{ij} are usually values of the attribute that appear in the training set. In the beam search process used by many rule learners, at each step the best b versions of the rule according to some evaluation function are selected for further specialization. AQ [Michalski, 1983] continues adding conditions until the rule is "pure" (i.e., until it covers examples of only one class). This can lead to severe overfitting. The latest version of the CN2 system [Clark and Niblett, 1989; Clark and Boswell, 1991] uses a simple and effective Bayesian method to combat this: induction of a rule stops when no specialization improves its error rate, and the latter is computed using a *Laplace correction* or *m-estimate*. If n_r is the number of examples covered by a rule r, and e_r is the number of those examples it misclassifies, the conventional estimate of the rule's error rate is e_r/n_r, but its m-estimate is:

$$\hat{\epsilon}_r = \frac{e_r + \mathsf{m}\epsilon_0}{n_r + \mathsf{m}} \quad (14)$$

where ϵ_0 is the rule's *a priori* error, which CN2 takes to be the error obtained by random guessing if all classes are equally likely: $\epsilon_0 = (c-1)/c$, where c is the number of classes. This prior value is given a weight of m examples (i.e., the behavior of Equation 14 is equivalent to having m additional examples covered by the rule, one of each class). CN2 uses m$=c$. As conditions are added, the rule covers fewer and fewer examples, and $\hat{\epsilon}_r$ tends to ϵ_0. Thus a rule making more misclassifications may be preferred if it covers more examples, causing induction to stop earlier and reducing overfitting. Clark and Boswell [1991] found this version of CN2 to be more accurate than C4.5 [Quinlan, 1993] on 10 of the 12 benchmark datasets they used for testing. However, this scheme ignores that, as more and more conditions are attempted, the probability of finding one that appears to reduce the rule's error merely by chance increases. This will lead the m-estimate to underestimate the chosen condition's true error, and CN2 to overfit. The upward correction made to ϵ_r should increase with the number of conditions attempted. The process-oriented evaluation framework described in the previous section allows us to do this in a systematic way, as follows.

Equation 13 can be used to compare the hypotheses returned by k learners $L_1, \ldots, L_m, \ldots, L_k$, and choose the one with lowest predicted error. It can also be used to compare successive stages of the same learner, by taking L_{m_2} to be the result of continuing the search of learner L_{m_1} ($m_1 < m_2$) with $m_2 - m_1$ more hypotheses. In

Table 1: Empirical results: error rates and theory sizes of default CN2 and CN2 with two versions of process-oriented evaluation (CN2-POE1 and CN2-POE2).

Dataset	Error rate			Theory size		
	CN2	CN2-POE1	CN2-POE2	CN2	CN2-POE1	CN2-POE2
Breast	30.0±1.4	29.7±1.4	30.3±1.3	114.5±2.4	58.7±2.6	104.9±2.6
Echocardio	32.7±1.2	32.3±1.3	31.2±1.1	42.9±1.2	35.4±2.1	39.2±1.3
Glass	39.0±1.5	38.3±1.7	39.1±1.4	51.8±1.0	54.7±1.1	45.2±1.0
HeartC	20.8±0.8	22.5±0.8	22.4±0.8	57.8±0.9	52.0±1.0	52.6±1.0
HeartH	22.4±1.1	21.8±1.3	21.9±1.1	69.2±1.5	60.3±1.4	58.9±1.1
Hepatitis	21.2±0.9	19.2±1.3	18.8±1.1	40.2±1.7	34.0±1.3	34.4±1.1
Lympho	21.4±1.1	24.1±1.1	23.4±1.2	39.5±0.7	38.7±1.0	32.8±1.1
Soybean	19.5±1.0	19.4±1.0	22.9±1.2	116.7±2.3	110.9±3.1	97.7±1.7
Thyroid	4.1±0.2	3.8±0.2	4.0±0.2	97.5±2.0	104.8±2.0	83.4±2.6
Tumor	60.1±1.0	65.1±1.3	60.0±1.2	302.8±4.6	273.9±4.4	241.6±3.9
Voting	4.8±0.4	4.3±0.3	4.3±0.3	61.7±2.9	49.6±2.5	33.2±1.7

particular, the successive stages can be the successive versions of a rule returned by CN2 or a similar "separate and conquer" rule learner. A natural choice for the prior expected error $\epsilon_{m,c}^{Prior}$ for all rule versions is the default error rate, obtained by always predicting the most frequent class in the training set. The choice of slower-growing function of m is less obvious. One possibility is $m' = \log m$ (for $m > 1$), based on an analogy with decision tree induction. When learning a tree using an algorithm like C4.5, each new hypothesis is obtained by modifying the previous one in only a fraction of the instance space (the fraction corresponding to the node currently being expanded), and this fraction becomes exponentially smaller as induction progresses. Only an entire new level of the decision tree corresponds to an entirely new hypothesis. Since the depth of the tree grows approximately with the logarithm of the number of nodes, we can take the equivalent number of independent hypotheses attempted m' to be proportional to the logarithm of the total number of hypotheses attempted m. Since a rule corresponds to a path through a decision tree, both in its content and in the way it is induced by a system like CN2, we can apply a similar line of reasoning to the number of rules attempted.[3]

Let each hypothesis be one version of the rule attempted during the beam search. Equation 13 does not need to be computed for every rule version generated during the beam search. This would introduce a preference for adding some conditions instead of others, which is unlikely to produce good results unless there is domain knowledge supporting such preferences. Instead, Equation 13 can be computed only once for each round. One round consists of generating every possible one-step specialization of each rule version in the beam, and selecting the b best. Thus, if there are a attributes and v is the maximum number of values of any attribute (in the worst case, $v = n$ for numeric attributes), one round corresponds to $O(bav)$ rule versions. Let m_k be the total number of rule versions generated up to, and including, round k. Round 1 consists of the initial rule with no conditions, and $m_1 = 1$. Induction stops when $E[\epsilon_{m_k}|n_{m_k}, e_{m_k}] \geq E[\epsilon_{m_{k-1}}|n_{m_{k-1}}, e_{m_{k-1}}]$, for $k > 1$.

4 Experiments

In order to test the effectiveness of process-oriented evaluation, default and process-oriented versions of CN2 were compared on the benchmark datasets previously used by Clark and Boswell [1991] .[4] The process-oriented versions were implemented by adding the necessary facilities to the CN2 source code. Details of the earlier version of POE and its implementation can be found in [Domingos, 1998b]. CN2's Laplace estimates are still used to choose the best b specializations in each round. This is preferable to using uncorrected estimates, since as implemented POE has no preference between hypotheses within the same round, and this is also a factor in avoiding overfitting. However, the Laplace correction distorts the value of ϵ_m^{ML} used in Equation 13. This will be particularly pronounced when there are many classes, since CN2 uses $\mathsf{m} = c$. In order to minimize this problem, $\mathsf{m} = 2$ was used with POE.[5]

The experimental procedure of [Clark and Boswell, 1991] was followed. Each dataset was randomly divided into 67% for training and 33% for testing, and the error rate and theory size (total number of conditions) were measured for default CN2, CN2-POE1 (the earlier version) and CN2-POE2 (the version described in this paper). This was repeated 20 times. The average results and their standard deviations are shown in Table 1;[6]

[3] In the experiments described below, the results were not sensitive to the base of the logarithms used. Base 2, base e and base 10 all yielded practically indistinguishable error rates and theory sizes. The results reported are for base 2.

[4] With the exception of pole-and-cart, which is not available in the UCI repository [Blake *et al.*, 1998].

[5] Simply changing $\mathsf{m} = c$ to $\mathsf{m} = 2$ in default CN2 does not change its performance on the datasets used.

[6] There are some differences between CN2's results and those reported in [Clark and Boswell, 1991]. This may be due

the results for CN2 and CN2-POE1 are from [Domingos, 1998b].

Compared to CN2-POE1, CN2-POE2 roughly maintains accuracy (lower error in five datasets, higher in five, same in one; 0.2% lower error on average) while reducing theory size in most datasets (lower in seven, higher in four, 4.5 fewer conditions on average). This indicates that Equation 13 is successfully deleting unnecessary conditions that the previous method retained. Being in closed form, Equation 13 is also much more efficient to evaluate than the integrals in [Domingos, 1998b]. Experiments on two larger UCI databases (shuttle and letter) showed CN2-POE2 learning faster than CN2, while maintaining accuracy and reducing theory size.

5 Related Work

The literature on model selection and error estimation is very large, and we will not attempt to review it here. Several pieces of previous work take into account the number of hypotheses being compared, and so can be considered early steps towards process-oriented evaluation. This includes notably systems that use Bonferroni corrections when testing significance (e.g., [Gaines, 1989; Jensen and Schmill, 1997]; see also [Klockars and Sax, 1986]). A key difference between these systems and what is proposed here is that they require a somewhat arbitrary choice of significance threshold, while this paper directly attempts to optimize the end goal (expected generalization error). Also, the Bonferroni correction does not take hypothesis dependencies into account, while the present framework offers (at least in principle) a way of doing so.

Quinlan and Cameron-Jones's [1995] "layered search" method for automatically selecting CN2's beam width can also be considered a form of process-oriented evaluation. While layered search and the approach proposed here have similar aims, their biases differ: layered search limits the search's width, while the present method limits its length. The latter may be more effective in reducing the fragmentation and small disjuncts problems [Pagallo and Haussler, 1990; Holte *et al.*, 1989]. The assumptions made here are also clearer than those implicit in Quinlan and Cameron-Jones's [1995] measure.

Freund [1998] recently proposed a form of process-oriented evaluation that is closer to the PAC-learning framework. It is an extension of the statistical query model [Kearns, 1993] that attempts to obtain tighter bounds on generalization error by considering the tree of queries that the learner could make. While the general algorithm to obtain these bounds has exponential computational cost in the number of queries made, Freund proposes a specialized version for algorithms based on local search (e.g., CN2) that is more efficient, at the price of loosened bounds. How tight the bounds will be in either case is still an open question; no empirical testing of Freund's [1998] method has been carried out so far. These bounds could be used for model selection by preferring the model with the lowest upper bound (for given parameters). However, as with Bonferroni corrections, the result will in general depend on the choice of parameters, for which there is no clear criterion. While the approach proposed in the present paper directly obtains an estimate of the generalization error, it would also be useful to have a confidence interval for it, and Freund's [1998] method may be a path to it.

Evaluating models that are the result of a search process, not just of fitting the parameters of a predetermined structure, has traditionally not been a concern of statisticians. However, this is beginning to change [Chatfield, 1995].

6 Conclusion

Two main types of model selection are currently available. In *data-oriented evaluation*, a hypothesis's score does not depend on its form or how the hypothesis was found, but only on its performance on the data. In *representation-oriented evaluation*, the score depends on the data and on the hypothesis's form, but not on the search process that led to it. Recently [Domingos, 1998b] we argued that the latter cannot be ignored, and proposed *process-oriented evaluation* (POE). However, in [Domingos, 1998b] we assumed that all models searched had similar true error rates, and that all error rates were equally likely *a priori*. In this paper we removed these assumptions, and derived a simple approximation for the generalization error of the returned hypothesis as a function of the number of hypotheses searched. This approximation is a weighted average of the maximum likelihood estimate of the error and the prior expected error, that increasingly favors the prior as more models are attempted. This approximation gives a mathematical basis to the intuition that model uncertainty should increase with the amount of search conducted.

In the future we plan to: study the statistical properties of Equation 11, in particular when the sample size is not large enough to approximate it by Equation 13; compare the method proposed here with other forms of process-oriented evaluation (e.g., Bonferroni corrections and layered search); apply it to other learners; and study methods for accurately estimating the growth of the effective number of hypotheses m' in each of these learners.

References

[Bernardo and Smith, 1994] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory.* Wiley, New York, NY, 1994.

[Blake *et al.*, 1998] C. Blake, E. Keogh, and C. J. Merz. UCI repository of machine learning databases. Machine-readable data repository, Department of Information and Computer Science, University of California at Irvine, Irvine, CA, 1998. http://-www.ics.uci.edu/~mlearn/MLRepository.html.

to the fact that the default version of CN2 uses a beam size of 5, whereas Clark and Boswell used $b = 20$. The distribution version of CN2 may also differ from the one used in [Clark and Boswell, 1991].

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

[Brunk and Pazzani, 1991] C. Brunk and M. J. Pazzani. An investigation of noise-tolerant relational concept learning algorithms. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 389–393, Evanston, IL, 1991. Morgan Kaufmann.

[Cestnik, 1990] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, pages 147–149, Stockholm, Sweden, 1990. Pitman.

[Chatfield, 1995] C. Chatfield. Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society A*, 158, 1995.

[Chickering and Heckerman, 1997] D. M. Chickering and D. Heckerman. Efficent approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29:181–212, 1997.

[Clark and Boswell, 1991] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proceedings of the Sixth European Working Session on Learning*, pages 151–163, Porto, Portugal, 1991. Springer-Verlag.

[Clark and Niblett, 1989] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[Domingos, 1998a] P. Domingos. Occam's two razors: The sharp and the blunt. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 37–43, New York, NY, 1998. AAAI Press.

[Domingos, 1998b] P. Domingos. A process-oriented heuristic for model selection. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 127–135, Madison, WI, 1998. Morgan Kaufmann.

[Efron and Tibshirani, 1993] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, NY, 1993.

[Freund, 1998] Y. Freund. Self bounding learning algorithms. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, Madison, WI, 1998. Morgan Kaufmann.

[Gaines, 1989] B. R. Gaines. An ounce of knowledge is worth a ton of data: Quantitative studies of the tradeoff between expertise and data based on statistically well-founded empirical induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 156–159, Ithaca, NY, 1989. Morgan Kaufmann.

[Holte *et al.*, 1989] R. C. Holte, L. E. Acker, and B. W. Porter. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 813–818, Detroit, MI, 1989. Morgan Kaufmann.

[Jensen and Cohen, 1998] D. Jensen and P. R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 1998. To appear.

[Jensen and Schmill, 1997] D. Jensen and M. Schmill. Adjusting for multiple comparisons in decision tree pruning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 195–198, Newport Beach, CA, 1997. AAAI Press.

[Kearns and Vazirani, 1994] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

[Kearns, 1993] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth ACM Symposium on the Theory of Computing*, pages 392–401, New York, NY, 1993. ACM Press.

[Klockars and Sax, 1986] A. J. Klockars and G. Sax. *Multiple Comparisons*. Sage, Beverly Hills, CA, 1986.

[Michalski, 1983] R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–161, 1983.

[Ng, 1997] A. Y. Ng. Preventing "overfitting" of cross-validation data. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 245–253, Nashville, TN, 1997. Morgan Kaufmann.

[Pagallo and Haussler, 1990] G. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 3:71–99, 1990.

[Quinlan and Cameron-Jones, 1995] J. R. Quinlan and R. M. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1019–1024, Montréal, Canada, 1995. Morgan Kaufmann.

[Quinlan, 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[Shawe-Taylor *et al.*, 1996] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. Technical Report NC-TR-96-053, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1996.

[Stone, 1974] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.