Error-Driven Stochastic Search for Theories and Concepts

Owen Lewis (olewis@mit.edu), **Santiago Perez** (spock@mit.edu), **Joshua Tenenbaum** (jbt@mit.edu)
Department of Brain and Cognitive Science, 43 Vassar Street
Cambridge, MA 02139 USA

Abstract

Bayesian models have been strikingly successful in a wide range of domains. However, the stochastic search algorithms generally used by these models have been criticized for not capturing the error-driven nature of human learning. Here, we incorporate error-driven proposals into a stochastic search algorithm and evaluate its performance on concept and theory learning problems. Compared to a model with random proposals, we find that error-driven search requires fewer proposals and fewer evaluations against labelled data.

Keywords: Bayesian inference; algorithmic level; concepts and categories

Introduction

From infancy, humans impose structure on the world with an impressive array of abstractions, conceptual categorizations and intuitive and formal theories. Characterizing these structures and explaining how they might be learned from data are formidable challenges for both cognitive science and artificial intelligence. Over the past decade, a class of probabilistic Bayesian models has emerged as a promising and unifying account of how a learner could acquire concepts and theories (Tenenbaum, Kemp, Griffiths, & Goodman, 2011). These models cast learning as statistical inference: the learner's goal is to approximate a posterior distribution over the class of structures to be learned, weighing a candidate structure according to its ability to account for the observed data and its probability according to the learner's prior beliefs. This probabilistic framing allows these models to capture both rule-like and graded aspects of human concepts and theories.

Bayesian models are able to discover good abstractions in a number of domains, and in many of these cases they make predictions that agree qualitatively or quantitatively with experimental data from human learners. For many Bayesian models, though, such predictions are confined to the Marr computational level of analysis: they predict *which* structures a learner will discover or prefer, namely those with high posterior probability, but they are largely agnostic about the algorithmic details of how the learner makes these discoveries.

Internally, most Bayesian models in cognitive domains approximate the target posterior distribution using stochastic search. The most widely used family of search algorithms, which includes the Metropolis Hastings algorithm and simulated annealing, has the following iterative *propose* and *accept* structure. Given a current candidate structure, the algorithm perturbs it, generating a new candidate called a *proposal*. The proposal is then evaluated, and if it is *accepted* it displaces the previous candidate as the current hypothesis. Usually, a proposal is accepted deterministically if it has higher posterior probability than the current hypothesis, and stochastically if it has lower posterior probability. Algorithms of this kind are simple, robust, and effective, but it has been unclear how they relate to the processes of human learning.

Recently, though, researchers have started to address this issue (Griffiths, Vul, & Sanborn, 2012). For instance, Ullman et al. (2012) examine a collection of theory learning tasks, showing that a stochastic search model can qualitatively reproduce the dynamics of human learning across several domains. Bonawitz et al. (2011) connect approximate Bayesian inference to earlier algorithmic-level models of human concept learning, and construct sequential approximation schemes that are able to capture aspects of human performance on a trial-by-trial basis.

Despite these successes, criticisms of stochastic search as a process model of human learning remain. One of the most powerful of these criticisms, made by L. Schulz (2012), hinges on the proposal mechanism by which new candidates are produced. In most existing stochastic search models, including the process models of Ullman et al. and Bonawitz et al., proposals are made *randomly*; Schulz argues that human learning is more structured. Specifically, human learning is *error-driven*: learners make proposals that fix specific deficiencies in their current hypothesis.

Efficient, error-driven search may hold the answer to another criticism of stochastic search. Humans (Feldman, 2000), even young children (Bonawitz et al., 2012), are able to learn remarkably quickly and efficiently, but existing search models are often slow. For instance, (Bonawitz et al., 2012) shows that children are able to learn a theory of magnetism in a matter of minutes, but computational models take many hours to solve similar problems. Relatedly, human learning performance scales remarkably well with problem complexity (Feldman, 2000), while computational models struggle as search spaces become larger. We present a concrete implementation of error-driven search and show that it can help close this gap. By considering only those hypotheses that fix specific problems, an error-driven learner can avoid irrelevant parts of the search space and converge to a good solution quickly.

A rich tradition of error-driven learning models exists in the classical literature on symbolic learning in AI and cognitive science. For instance, version space learning (Mitchell, 1978), FOIL (Quinlan, 1990) and explanation-based learning (Mitchell, Keller, & Kedar-Cabelli, 1986) all explore the idea of iteratively modifying hypotheses to account for specific observations. However, despite enjoying some notable successes, these models lack some of the capabilities of Bayesian models, for instance the ability to account for gradedness in human learning, and for humans' ability to learn from noisy data.

The contribution of this paper is to synthesize ideas from this earlier tradition of error-based learning with contemporary Bayesian models. We present simple error-driven proposal mechanisms for two concept and theory learning domains in which Bayesian modeling has been successfully applied in the past. We show that these error-driven algorithms are significantly more efficient than the purely random ones that have appeared in the literature so far, making them both more widely applicable and closer in capability and character to human learning.

Modeling Framework

The family of stochastic search algorithms discussed in this paper share the following abstract form:

```
$h \leftarrow$ random hypothesis
repeat
    $h' \sim Q(h' \mid h)$
    $r \leftarrow \frac{P(D|h')P(h')}{P(D|h)P(h)}$
    if $r > 1$ then
        $h \leftarrow h'$ determinisitcally.
    else
        $h \leftarrow h'$ with probability $r^{\frac{1}{T}}$
    end if
until convergence
```

Here, the hypothesis h represents the current estimate of the target theory or concept, and D is a set of observed data. T is a "temperature" parameter that controls the algorithms tendency to accept proposals with smaller posterior than the current hypothesis.

From the point of view of this paper, the key component of this algorithm template is the proposal distribution $Q(h' \mid h)$. As discussed in the introduction, standard choices of this distribution are purely random; the main algorithmic contribution of this paper is to supplement these random proposals with error-driven ones that correct mistakes made by the current hypothesis. In each of the two domains we study, we present two proposal distributions, a random one $Q_{rand}(h' \mid h)$, and an error-driven one $Q_{err}(h' \mid h, e)$, which is conditional on the example e whose prediction is to be corrected.

In order to get a fine-grained picture of the benefits of error-driven proposals, we study a parametrized family of models whose proposal distribution is a mixture of Q_{err} and Q_{rand}. We introduce a parameter p_{rand} as the mixture weight:

$$Q = p_{rand} \cdot Q_{rand} + (1 - p_{rand})Q_{err}$$

Thus, $1 - p_{rand}$ can be interpreted as the average "error-drivenness" of a model.

Apart from the proposal distribution, two other components of this template are important problem-dependent choices: the prior distribution $P(h)$, the likelihood $P(D \mid h)$. In the remainder of this section, we describe algorithms for two learning problems: rule-based concepts and a simplified theory of magnetism. For each of these problems we define prior distributions and likelihoods, and two different proposal distributions, one random and one error-driven. The search algorithms we create in this way are problem-specific and distinct from one another, but error-driven proposals play the same role in both.

Concept Learning

Our concepts are defined over objects represented as collections of features. Each object has feature dimensions $f_1, f_2, \ldots, f_{n_{dims}}$, each of which takes values in the set $v_1, v_2, \ldots, v_{n_{vals}}$. For concreteness, one may think of the feature dimensions as attributes like *shape*, *color* and *size*, and values as instantiations of these attributes, such as *triangle*, *green*, *small*.

We define a concept as a rule expressed in disjunctive normal form (DNF), specifying values for some or all of the features. For example,

$$c = (color = green \wedge shape = triangle) \vee (color = blue)$$

Such a concept induces a function from the set of objects to the set $\{True, False\}$. For instance, if an object e_1 is a blue circle, then $c(e_1) = True$.

We construct a concept learning problem by first generating a target concept c_T, and then generating a set of example $\{e_i\}_{i=1}^{n_{ex}}$, together with their labels according to the target concept, $\{c_T(e_i)\}_{i=1}^{n_{ex}}$. With these labelled examples as inputs, the concept learner's task is to recover the original target concept, or an approximation to it.

With the problem setup in place, we move to the definition of the components of the search algorithm.

Prior: Goodman et al. (2008) present a Bayesian analysis of concept learning centered on a stochastic *DNF grammar* over concepts, similar to the one below:

$$\begin{array}{lcl} S & \rightarrow & D \\ D & \rightarrow & C \vee D \\ & | & False \\ C & \rightarrow & P \wedge C \\ & | & True \\ P & \rightarrow & f_i = v_j \end{array}$$

The language defined by this grammar consists of all and only the well-formed DNF formulae with primitive propositions of the form $f_i = v_j$. Given this generative model, the prior definition is automatic: the prior probability of a concept is its probably of being produced by the DNF grammar. Because concepts with more conjuncts or disjuncts require more grammar productions, they are penalized by the prior; the prior implements a simplicity bias.

Likelihood: Given a set dataset $D = \{(e, c_T(e))\}$, we can define the likelihood of a hypothesized concept h on any subset $S \subset D$. Following Goodman et al. (2008) , we define

$$P(S \mid c) = e^{-b|w|} \qquad w = \{s \in S \mid c_T(s) \neq c(s)\},$$

Goodman et al. (2008) found $b = 4$ to give good agreement with human data; this is the value we use for all experiments. The definition of likelihood in terms of a possibly proper subset of D differs from Goodman et al. (2008), and is deliberate; we leave $|S| := n_{eval}$ as a model parameter. This choice is partly motivated by cognitive plausibility, specifically the intuition that people modify their hypotheses "on the fly," rather than holding them fixed during an exhaustive enumeration. More concretely, as we will show later, using a smaller evaluation set can lead to efficiency gains. In particular, error-driven proposals in the concept domain are generally of sufficiently high quality that they do not need to vetted on the entire dataset.

Random proposals: In Goodman et al. (2008), proposals are generated by randomly selecting a non-terminal node in the tree representation of a concept and regrowing the subtree below it. The random proposals we use differ in two specifics.

First, the binary tree representation produced by the DNF grammar introduces asymmetries within the sets of conjuncts and disjuncts. In particular, nodes closer to the root of the tree are less likely to be modified than nodes closer to the leaves, since these lower nodes are contained in a greater number of subtrees. This asymmetry is undesirable, since it makes mistakes high in the tree difficult to correct. We correct this issue by randomly permuting the tree before each proposal.

Second, Goodman et al. (2008) uses a proposal distribution symmetrized with the Metropolis correction. With this modification, the algorithm obeys the detailed balance requirement, and therefore benefits from theoretical guarantees associated with Metropolis Hastings algorithms. In this paper we are interested in optimization rather than sampling; we are satisfied with a single high-probability candidate, and do not require a full characterization of the posterior distribution. So we leave the proposals in both the random and error-driven model asymmetric. While optimization of this kind is adequate, and indeed perhaps preferable, for many applications, it will also be interesting to study symmetric error-driven models in future work.

Error-driven proposals: An incorrect prediction $h(e)$ is either a false negative or a false positive. In the case of a false negative, the proposed correction h' must be a *generalization* of h. Generalization can be accomplished by either of the following two operators.

- `add-or` adds a disjunct to h, choosing a random assertion $f_i = v_j$ from the feature representation of e, returning $h' = h \wedge f_i = v_j$.

- `del-and` removes one more conjuncts from h. From the set of all disjuncts containing at least one feature true of e, `del-and` chooses one element at random, and removes from it all conjuncts *not* true of e.

For the case of a false positive, we have two *specialization* operators, dual to the generalization ones.

- `del-or` removes from h all disjuncts true of the negative example e.

- `add-and`, works by finding all disjuncts that are true of e, and adding to each a conjunct *not* true of e.

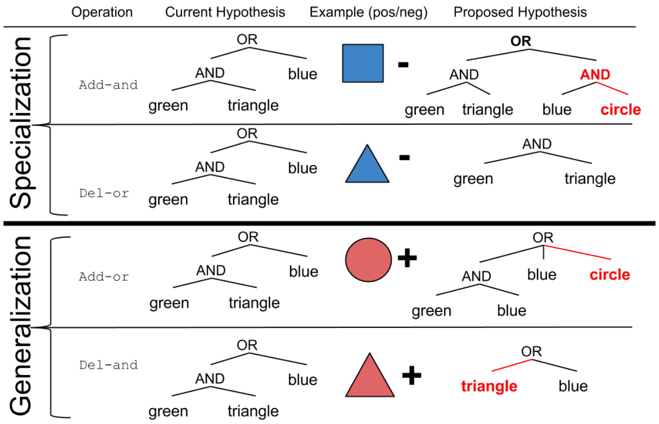

Figure 1: Example applications of specialization and generalization.

An example application for each operator is shown in figure 1.

Given specialization and generalization operators, making error-driven proposals is straightforward: find a misclassified example e, and apply a randomly chosen specialization or generalization operator, as appropriate.

Theory learning

In our theory learning problem, adapted from Ullman et al. (2012), we imagine a learner confronted with a collection of objects, some of which are plastic, some of which are magnets, and others of which are metallic but not magnetic. These objects interact in the expected way: magnets interact with metals and other magnets, and no other pairs of objects interact. The learner observes some collection of interactions and non-interactions, and must infer a theory that compactly describes and predicts these observations. Importantly, the objects are indistinguishable by their surface characteristics, so the learner is not aware *a priori* how, or that, the objects should be grouped into types. Thus, the theory learner is faced with a chicken and egg problem: she must infer causal laws of interaction stated in terms of latent kinds at the same time that she infers the definition and extension of the kind terms.

Formally, a theory takes the form of a collection of Horn clauses. For example, the correct theory for the magnetism domain is:

$$\begin{aligned}
interacts(X,Y) &\leftarrow magnet(X), magnet(Y)\\
interacts(X,Y) &\leftarrow metal(X), magnet(Y)\\
not\_interacts(X,Y) &\leftarrow plastic(X), plastic(Y)\\
not\_interacts(X,Y) &\leftarrow plastic(X), metal(Y)\\
not\_interacts(X,Y) &\leftarrow plastic(X), magnet(Y)
\end{aligned}$$

We call these Horn clauses *rules*. We provide the learner with the knowledge that interaction is symmetric, meaning that permutations of these rule need not be learned.

We call predicates like *magnet* and *plastic kinds*. A complete theory requires, in addition to rules, *assignments* specifying the extensions of the kinds. These take the form *metal(a)*, *magnet(b)*, etc., where the lowercase letters refer to specific objects.

As we did for concept learning, we specify search algorithms by defining a prior over the theories, a likelihood, and random and error-driven proposal distributions. Also like concept learning, the point of departure for our approach is a grammar-based Monte Carlo model, of the kinds presented in (Katz, Goodman, Kersting, Kemp, & Tenenbaum, 2008; Ullman, Goodman, & Tenenbaum, 2012). For theory learning, though, we stray further from this existing model than we did for concepts.

Prior: Like in concepts, priors on theories measure the probability that a theory was generated by a grammar, this time one that generates conjunctions of Horn clauses.

$$\begin{array}{lcl}
S & \rightarrow & R \wedge S \quad | \quad Stop \\
R & \rightarrow & H \Leftarrow B \\
H & \rightarrow & interacts(X,Y) \quad | \quad not\_interacts(X,Y) \\
B & \rightarrow & K_1 \wedge K_2 \\
K_1 & \rightarrow & kind_1(X) \;|\; \ldots \;|\; kind_n(X) \;|\; kind_{new}(X) \\
K_2 & \rightarrow & kind_1(Y) \;|\; \ldots \;|\; kind_m(Y) \;|\; kind_{new}(Y)
\end{array}$$

Here, the $kind_i$ are the kinds already used in the derivation, and $kind_{new}$ is a fresh kind symbol.

Likelihood: As with concepts, a theory makes predications about the observed data which can be compared with ground truth. In the theory of magnetism, predictions take the form of assertions that pairs of objects do or do not interact. Note that, unlike in the concept domain, a hypothesized theory can fail to make a prediction about an observation; this occurs, for instance, when objects have not yet been assigned kinds. As in concepts, we define likelihood as

$$P(S \mid c) = e^{-b|w|} \qquad w = \{s \in S \mid c_T(s) \neq c(s)\},$$

where w is now the set of incorrect and missing predications. We used $b = 5$, and as before, $|S| := n_{eval}$.

Random proposals: Given h, we produce a proposal h' by both randomly regrowing one of h's subtrees, and reassigning the kinds of a geometric number of objects, using only those kinds that appear in the modified tree.

Error-driven proposals: As in concept learning, an error-driven proposal produces a hypothesis that fixes a specific incorrect prediction, for example $e = interacts(a,b)$. The proposal is produced by the following two steps.

1. Choose a subset of size N of $\{a,b\}$ and assign its members to either existing or new kinds. N is geometrically distributed.

2. Update the rules to reflect the new assignments. After this process the hypothesis correctly predicts each interaction involving either a or b.

MH-Gibbs: In existing literature on stochastic search for theories, a third search strategy is most prevalent (Katz et al., 2008; Ullman et al., 2012). We present it briefly here for purposes of comparison; for a more complete discussion, see (Katz et al., 2008; Ullman et al., 2012). This strategy separates the learning of rules and assignments. The search for rules takes the form of a Metropolis Hastings search in which proposals are generated from a Horn clauses grammar, similar to the DNF grammar for concepts presented above. Assignments are found conditional on a set of rules: given a proposal at the rule level, an "inner loop" estimates good assignments using Gibbs sampling. This Gibbs sampler randomly reassigns one object at time, sampling the assignment conditioned on all of the other existing assignments.

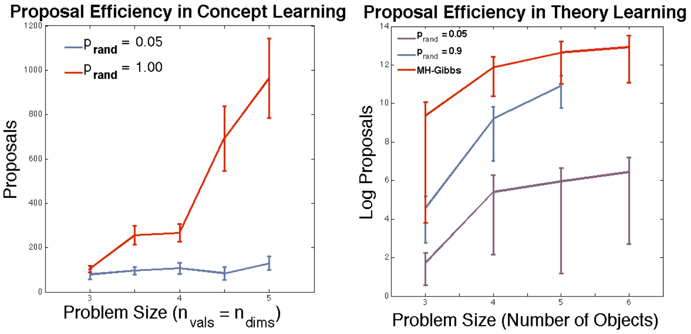

Figure 2: **Left.** Performance of error-driven and random concept learning on problems of different sizes. On the x-axis is a single problem size parameter, $s := n_{dims} = n_{vals}$. On the y-axis is the log of the number of proposals the model made before it converged. We ran each model 30 times on each of 18 target concepts and selected for each model the concept on which it achieved median performance; the plotted values are the mean and standard error of the 30 runs of the model on this concept.
Right. Performance of error-driven and random theory learning. Here, problem size is number of objects, and we ran each model 30 times on the correct theory of magnetism. All runs had one magnetic, two metallic, one plastic and a randomized assortment of other objects. For the MH-Gibbs model we counted the number of data accesses used before the optimal assignment was reached. It was computationally infeasible to compute $p_{rand} = 0.9$ at 6 objects.

For all models, before evaluating a proposal, we remove contradictory rules and rules that do not apply given the current assignments.

Simulation results

In the preceding section, we introduced error-driven learning algorithms for theory and concept learning. Here, we test the claim that these algorithms represent an improvement over their purely random counterparts. We present two experiments, designed to test different notions of efficiency. The first defines efficiency as the ability to find a good theory concept using a small number of proposals, and the second as the ability to find a good solution using a small number of queries to the observed data.

Experiment One: Proposal Efficiency

If error-driven proposals are more effective than random ones, error-driven search should have to consider a smaller number of proposals before convergence, an advantage that should

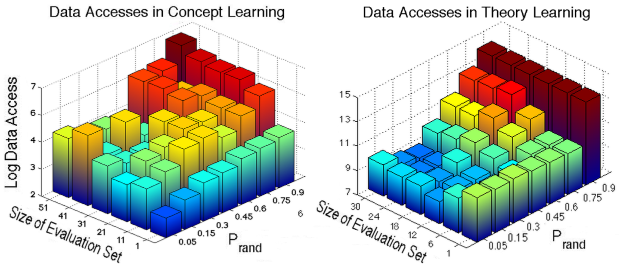

Figure 3: Number of data-accesses before convergence for concept learning (left) and theory learning (right), as p_{rand} and n_{eval} vary. The results for concept learning were obtained with $n_{dim} = n_{vals} = 5$, and 30 examples. The values shown are averages of 300 runs, three for each of 100 concepts. For theories, the results used six objects, and are averaged over 50 runs. We terminated all simulations after 100,000 proposals, a ceiling which $p_{rand} = 0.9$ consistently hit.

grow with the size of the search space. This is the intuition tested by this first experiment.

The tasks in this experiment are generated by creating a target concept or theory and a set of observed data consistent with it. We then measure the ability of our models to recover the target structure from the data. Specifically, we say that a search algorithm has *converged* when it discovers a theory or concept at least as good as the one actually used to generate the data, where goodness is measured by posterior probability. The efficiency of a model is the number of proposals it generates before convergence. Using this measure of efficiency, we evaluate error-driven and random models as the size of search space grows. The details of the experimental setup are as follows.

Concepts: Concept learning problems have three parameters: n_{dims}, the number of feature dimensions possessed by each object, n_{vals}, the number of values possible in each dimension, and n_{ex} the number of examples in the dataset. In this experiment, we make the constraint $n_{dims} = n_{vals} := s$, and fix $n_{ex} = 60$, leaving s as the one free problem-size parameter.

Here we compare two models: an error-driven one with $p_{rand} = 0.05$, and a completely random one, with $p_{rand} = 1$. For each of these models, we evaluate likelihoods on the full evaluation set; $n_{eval} = n_{ex}$. To set the temperature parameter, we ran a cross-validation experiment with a different set of data, and chose for each of the models the temperature that maximized its efficiency. To compute the efficiency of each of the two models on each problem size, we ran them 30 times for each of 18 target concepts, and plotted the statistics for the problem with median mean difficulty.

The results, showing efficiency for the random and error-driven models on problems of different sizes, are given in figure 2. As expected, error-driven proposals are markedly more efficient than random ones. In particular, we draw attention to the scaling properties of the two algorithms. As the problem size grows, the number of samples required for convergence remains essentially constant for the error driven learner, but grows dramatically for the random learner.

Theories: In the theory learning version of the experiment, we fix the target theory to be the true theory of magnetism. The problem difficulty parameter is now the number of objects that the learner observes. We assume that the learner has access to a the complete set of pairwise interactions, making a total of $\binom{n}{2}$ observations for n objects, accounting for symmetry.

In addition to the completely random and completely error-driven models, we look at a variant of the MH-Gibbs model from (Katz et al., 2008; Ullman et al., 2012). The results are in figure 2. As with concepts, error-driven theory learning delivers marked efficiency gains, requiring several orders of magnitude fewer samples before convergence for all problem sizes.

Experiment Two: Data Efficiency

The previous experiment showed that using error-driven search can reduce the number of proposals needed to reach convergence, but it ignored the cost associated with making and evaluating each proposal. This is a significant omission: in many problem settings, acquiring and accessing data entail significant costs, meaning that models that access data less frequently are at a significant advantage. In this experiment, we evaluate data-accesses directly, defining (in)efficiency as the number of times a model has to “look at” a datapoint.

For both random and error-driven models, data accesses occur during proposal evaluation, when the likelihood of a proposal is calculated. The number of accesses required to evaluate the likelihood of a proposal is controlled by the parameter n_{eval} introduced in the previous section. Specifically, if a model generates p proposals before converging to a solution, its total number of data accesses due to evaluation is

$$d = p * n_{eval}.$$

The parameter n_{eval} represents a tradeoff between sample efficiency and data efficiency. At one extreme, setting $n_{eval} = n_{ex}$ will (on average) minimize the number of proposals needed, because a complete likelihood evaluation is the most reliable basis on which to decide whether a sample should be accepted. On the other extreme, setting $n_{eval} = 1$ gives a model that is maximally efficient *per proposal*, but one that will tend to accept bad proposals, resulting in an increased requirement of proposals before convergence. By the nature of error-driven models, we expect them to make proposals that are, on average, higher quality than random ones. It should therefore be possible to get away with evaluating these proposals on a smaller set than required for a random model. This is the hypothesis tested in this experiment. In this experiment, we do not vary the problem complexity. Instead, we fix a problem instance, and examine model performance for a range values of p_{rand} and n_{eval}.

In concept learning, likelihood evaluation is the only source of data accesses, and the number of data accesses per proposal is the same for error-driven and random models. For theory learning, though, an error-driven proposal triggered by an error e requires access to each interaction including the objects involved in e. Thus, in general, error-driven proposals require more accesses than random ones.

Results for the two learning domains are shown in figure 3. For concepts, we see that, by and large, smaller evaluation sets do lead to improved performance. In particular, for each value of p_{rand}, the fewest data accesses was achieved with $n_{eval} = 1$. For theories, though, smaller evaluation sets only represent an improvement for the larger values of p_{rand}, reflecting the greater cost of error-driven proposals.

In both theory and concept learning, smaller values of p_{rand} generally lead to better performance, but it is interesting to note that for theory learning the smallest value $p_{rand} = 0.05$ actually fails to be optimal. This is likely due to the fact that purely error-driven models can get caught on theories with high likelihood but low prior. In such states, there are few remaining errors to trigger new error-driven proposals, so a random proposal is required to move to a new hypothesis.

Discussion and Conclusion

We argued that Bayesian models of concept theory learning, already successful by many metrics, can be made more efficient and cognitively plausible by the use of error-driven proposals. We showed that in both concept learning and theory learning, error-driven algorithms are more efficient than purely random ones, both in terms of the number of proposals they consider before converging to a solution, and in terms of their use of observed data. These promising results raise some further questions.

A first major question concerns comparison with human data. As we argued in the introduction, the increased efficiency of error-driven models already represents an important improvement in cognitive fidelity: human learning is fast and scalable, so correct models of it should be as well. But while a plausible level of efficiency is a necessary condition for a cognitive model to be completely correct, it is not a sufficient one, and further experiments are indicated. One such experiment could replicate the test shown in figure 2 with human learners, examining how learning time scales with problem complexity. In addition, error-driven learning predicts a recency bias: the learner will tend to be preferentially correct on the last example it examined. Online learning paradigms could be used to test for such biases in humans, though it is important to note that memory limitations may lead to similar effects.

Second, algorithmic questions also remain. While the requirements of data and time made by an error-driven learner are more modest than those made by random search, they still seem excessive by the standards of human learning. What accounts for this discrepancy? A first important note is that the error-driven proposal mechanisms presented in this paper are far from the only ones possible. Indeed, the specific proposals we used were chosen as much for simplicity as for absolute performance: it seems likely that more thoroughly optimized choices will result in much more efficient models. Relatedly, since our main interest was in the relative performance of error-driven and random models, we applied few optimizations to either. For instance, other studies (Ullman et al., 2012) have found benefits from techniques like simulated annealing. Future work could determine if these methods help in the error-driven case as well.

Acknowledgments: This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

References

Bonawitz, E., Denison, S., Chen, A., Gopnik, A., & Griffiths, T. L. (2011). A simple sequential algorithm for approximating bayesian inference. In *Proceedings of the thirty-third annual conference of the cognitive science society* (pp. 2463–2468).

Bonawitz, E., Ullman, T., Gopnik, A., & Tenenbaum, J. (2012). Sticking to the evidence? a computational and behavioral case study of micro-theory change in the domain of magnetism. In *Icdl* (pp. 1–6).

Feldman, J. (2000). Minimization of boolean complexity in human concept learning. *Nature*, *407*(6804), 630–633.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–154.

Griffiths, T. L., Vul, E., & Sanborn, A. N. (2012). Bridging levels of analysis for probabilistic models of cognition. *Current Directions in Psychological Science*, *21*(4), 263–268.

Katz, Y., Goodman, N. D., Kersting, K., Kemp, C., & Tenenbaum, J. B. (2008). Modeling semantic cognition as logical dimensionality reduction. In *Proceedings of thirtieth annual meeting of the cognitive science society.*

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Association for the advancement of artificial intelligence* (Vol. 3, p. 5).

Mitchell, T. M. (1978). *Version spaces: an approach to concept learning.* (Tech. Rep.). DTIC Document.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine learning*, *1*(1), 47–80.

Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine learning*, *5*(3), 239–266.

Schulz, L. (2012). Finding new facts; thinking new thoughts. In *Advances in child development and behavior* (Vol. 43, p. 269-289).

Schulz, L. E., Goodman, N. D., Tenenbaum, J. B., & Jenkins, A. C. (2008). Going beyond the evidence: Abstract laws and preschoolers' responses to anomalous data. *Cognition*, *109*(2), 211–223.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*(6022), 1279–1285.

Tu, Z., & Zhu, S.-C. (2002). Image segmentation by data-driven markov chain monte carlo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(5), 657–673.

Ullman, T. D., Goodman, N. D., & Tenenbaum, J. B. (2012). Theory acquisition as stochastic search. In *Cognitive development* (p. 455-—480).