Theory Learning as Informed Stochastic Program Synthesis

Lucas E. Morales {lucasem}

1 Introduction

Theories, as formal logical expressions in the language of thought which represents laws as patterns, are intuitively developed by children. I expand on an algorithm which performs stochastic search on theory space and the then-generated space of models and explanations given experiences and observations [1]. The grammatical formalism of a language of thought gives the agent generative capacity for a hierarchical Bayesian framework of theory and model spaces on which a Monte-Carlo Markov-chain (MCMC) stochastic search algorithm is performed. Annealing is done so MCMC will visit all theories with probability proportional to their posterior probability.

The algorithm I propose guides stochastic search by sampling observations as data corresponding to the given model, using experience and a combination of exploration and exploitation of reality upon agent action. There are two phases to the algorithm: *wake*, and *sleep*. In the *wake* phase, the agent is given observations and a probing mechanism to analyze them, giving the agent ability to explore and exploit understanding. In the *sleep* phase, the agent retrospectively performs stochastic search and uses the observations to determine a more fitting theory and model for the next *wake* phase.

2 Theories and Models

A theory may be synthesized as a program in the "language of thought" (LoT) system, formalized with probabilistic lambda calculus, to utilize the compositional necessities of functional languages and the probabilistic behaviors of the natural world [2].

Theories take a set of parameters which yield a model to fill the knowledge gap between observation and higher-level structural understanding.

For clarity of concept, I will occasionally use the example of determining what Euclidean functions best describe a set of particular 2D graphs. For a simple non-probabilistic parabola centered at the origin, the theory T may solely be the exponentiation operator. This theory, in total, has two open placeholders R (expression 1) — the two operands of exponentiation for the theory (expression 2). For the parabola we're trying to model, there is one parameter $\theta_1 = 2$ (expression 3) with the corresponding parameter-mapping PM (expression 4). Finally, the remaining placeholders are dimensions of independent variability. In this case, the domain X of a single dimension of real numbers (expression 5) which has the domain-mapping DM on the remaining open placeholders (expression 6). Thus, this particular model can be sufficiently described as follows:

$$R := (r_1, r_2) \tag{1}$$

$$T(R) := (\texttt{pow}\ r_1\ r_2) \tag{2}$$

$$\Theta := (\theta_1) = (2) \tag{3}$$

$$PM : (\theta_1) \to (r_2) \tag{4}$$

$$X := (x_1 \in \mathbb{R}) \tag{5}$$

$$DM : (x_1) \to (r_1) \tag{6}$$

This is not the only model which describes the observed function, however. Another one which would be equally correct has no parameter, and $T(R) = (\texttt{mul}\ r_1\ r_2)$ with the single dimension of the domain mapped to both

placeholders $DM : (x_1, x_1) \to (r_1, r_2)$.

If the pattern wasn't exactly the same between observations, such as parabolas centered anywhere and not necessarily the origin, then probabilistic language primitives such as `uniform` and `gauss` may be used. In this scenario, the theory may be

$$T(R) := (\texttt{sum}\ (\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2)\ (\texttt{gaussian}))$$

where the gaussian samples are made for each particular model.

3 Model Inference

Bayes' theorem provides a scoring mechanism for a particular model M which implements theory T given observations D:

$$P(M|D,T) \propto P(D|M)P(M|T)$$

The model likelihood $P(D|M)$ comes from an approximation margin of the model with observation. The model likelihood uses samples of observation to calculate numerical error of a model's predictions. The performance of a model in this regard yields implicit negative evidence to discredit poor models [3]. The likelihood is thus the product of coefficients of determination using the sum of squares over residuals as compared to the variance of each observation. For a set of observations D where each observation i holds a domain vector x and range vector y,

$$P(D|M) = \prod_{i \in D} \left(1 - \frac{\sum_{x,y \in i}(y - M(x))^2}{\sum_{x,y \in i}(y - \bar{y})^2}\right)$$

The model prior $P(M|T)$ adheres to the principle that theories should closely pertain to their output with little dependence on mediating factors. Models that have fewer parameters $|\Theta|$ are more likely a priori, as they have fewer degrees of freedom and thus prevent over-fitting. This means that choosing a theory is often more important than choosing the model, because

parameter count is heavily contingent on the theory and the domain space — both of which are independent of the model [4]. Models are preferential to dependence on more observed parameters $|X|$. Models are also more likely a priori if the dimension of the range of the domain-mapping $|DM(X)|$ is greater than that of the domain itself (i.e. the domain accounts for more open placeholders), and likewise for the parameter-mapping. Using the notation from the section on theories and models:

$$P(M|T) \propto \frac{|DM(X)|\,(1 + |PM(\Theta)|)}{(1 + |\Theta|)^2}$$

4 Theory Inference

Bayes' theorem provides a scoring mechanism for a theory T in the theory space U (governed by the infinite space of the language of thought) given observations D:

$$P(T|D, U) \propto P(D|T)P(T|U)$$

Because theories act indirectly with the data they help model, the theory likelihood $P(D|T)$ is expanded to a sum of model likelihoods:

$$P(D|T) = \sum_M P(D|M)P(M|T)$$

The potentially infinite model space for a particular theory is contracted when determining the theory likelihood by fixing observations as parameters to the theory and considering the model parameters Θ as the variable domain. Sampling from the model space can be done by maximizing coherence (i.e. approximate equality) between parameters in different fixed observation settings. This yields likely models which serve as representatives for the theory. The model likelihoods and priors are computed according to the model inference section above.

The theory prior $P(T|U)$ is a lexically-determined score for a theory, holding two assumptions:

1. relevance of the size principle [5], having lower probability for more complex hypotheses.

2. recursive complexity, that recursive theories are less probable.

A syntactic prior defines a measurable prior for a probabilistic context-free Horn clause grammar [6] by evaluating the product of probabilities for each item in the parse tree. A probabilistic context-free grammar can be generated by expanding lambda expressions to their primitive functions called in a consistent manner, where functions can only be applied to valid arguments. This first assumption is implemented as the prior according to the rational rules model: $P_{RR}(T)$. For the lexical prior to assume recursive lexicons are less likely a priori, a free parameter γ exists:

$$P(T) \propto \begin{cases} \gamma \cdot P_{RR}(T) & \text{if } T \text{ uses recursion} \\ (1-\gamma) \cdot P_{RR}(T) & \text{otherwise} \end{cases}$$

Stochastic search is done in theory space using a grammar-based Metropolis Hastings (MH) inference algorithm as is implemented in the Church language [7]. The algorithm essentially follows a Markov chain on the space of valid lexical structures. A tentative new theory T' is sampled from the proposal distribution which constructs a new theory based on its predecessor. It is generated with a distribution on steps of removing a lexicon (expression 7) or adding a lexicon as a new higher-order composition (expression 8), as a compositional substitute for an existing lexicon (expression 9), or as a substitute for an open parameter (expression 10). There are finitely many possible steps for transition, so a sample can be correctly taken from the space. More will be discussed on this transition generation in the stochastic program synthesis section of this paper.

$$(\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2) \longrightarrow (\texttt{pow}\ r_1\ r_2) \tag{7}$$

$$(\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2) \longrightarrow (\texttt{sum}\ (\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2)\ (\texttt{gaussian})) \tag{8}$$

$$(\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2) \longrightarrow (\texttt{pow}\ (\texttt{mul}\ r_1\ (\texttt{gaussian}))\ r_2) \tag{9}$$

$$(\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ r_2) \longrightarrow (\texttt{pow}\ (\texttt{sub}\ r_1\ (\texttt{gaussian}))\ (\texttt{mul}\ r_2\ r_3)) \tag{10}$$

The acceptance probability of a sample is computed as

$$a = \frac{P(T'|D,U)}{P(T|D,U)} \cdot \frac{Q(T|T')}{Q(T'|T)} = \frac{P(T'|D,U)}{P(T|D,U)}$$

where $P(T|D,U)$ is the posterior probability described above and $Q(T'|T)$ is the proposal density. The proposal density serves to add directionality to theory sampling in MH. The lexical structure can go either way in simplicity or complexity, which the posterior accounts for, and thus the proposal density is deemed symmetric and not a factor in the acceptance probability. To aid convergence, this acceptance probability is slightly exponentiated as time progresses and more stochastic steps are taken.

The Metropolis Hastings algorithm is used as a Markov chain Monte Carlo method to emulate the stochastic computational search that's hypothesized to be done in theory learning and acquisition in young children. It frequents more probable theories but doesn't disregard *out-of-the-box* thinking and random innovation.

5 Observation and Action

In the *wake* phase of the algorithm, the agent makes observations. For a particular observation, the agent chooses a small number of representative samples. The agent can select samples at a similar concentration to ones it already has to attempt to reaffirm its current theory and model (exploitation),

or it can select samples that are unlike the those it already has (exploration). These data will be used for inference and stochastic progression in the *sleep* phase. In the 2D graph example, this step is like being given a function which you can only interact with by taking samples.

Let a normal distribution on previously observed samples serve as a default exploitation-based probabilistic model. The determination of the agent as to whether to exploit or explore is thus quantified as deviation from the mean. Making the decision for a particular sample is organized with the likelihood of the agent's current model, $P(D|M)$. More likely models are met with a higher probability of larger deviations in sampling. Less likely models are intuitively met with exploitation to help determine how accurate the model is, which will result in the model having either increased or decreased likelihood (thus effecting the next iteration of sampling and of stochastic program synthesis).

For model likelihood $\alpha = P(D|M)$, the deviation is sampled from the distribution with tunable parameter β:

$$D(\alpha) = \alpha^{\beta} \cdot \text{uniform} + (1 - \alpha)^{\beta} \cdot \text{gauss}$$

Sampling observations for data in this manner ensures stochastic annealing by selecting data points probabilistically for improved theory and model accuracy assurance.

6 Program Specification

The language on which programs are synthesized is a probabilistic lambda calculus. The selection of primitives is designed to specify parametrizable functions with multi-dimensional domains and ranges, though many primitives are isolated to a single dimension of output. These primitives are not disjoint in their capabilities and may be derivable from other primitives.

The selection of primitives corresponds to the built-in functions and capabilities of Church, a functional probabilistic programming language.

New functions can be learned using these language primitives. When a pattern of functions is used frequently, a small finitely-sized space of learned function definitions is populated with a new assignment for the pattern.

```
(define (isItem item) (lambda (otherItem) (if (eq? item otherItem) 1 0)))
```

After a utility such as `isItem` is defined, a theory for counting falsehoods in a boolean observation system (where many true/false values represent a single input, and integer output) may now be easily constructed (where r_1 is assigned `#f` in the model parameterization):

$$T(R) := (\texttt{sum}\ (\texttt{map}\ (\texttt{isItem}\ r_1)\ r_2))$$

This creation of new simpler functions from frequent complex primitive patterns allows for a more accurate theory prior $P(T|U)$ and improves the capabilities of the Metropolis Hastings sampler. Constructs like `isItem` are commonplace, and should thus be easily permitted by stochastic program synthesis in the MH theory sampler.

7 Stochastic Program Synthesis

The generation of new programs in the Metropolis Hastings sampler is nontrivial for such a high level computationally universal language such as Church. To obtain the proposal distribution $P(T'|T)$ on which samples are taken for MH theory search, every transition step s is associated a frequency $f(s)$. A set of frequency-explicit rules composes a transition step.

By default, every symbol change is given a frequency value of 1. Use of custom-defined functions has a particular tunable frequency value (e.g. 5). Different frequency rates may be applied to whether the step is a removal, a self-composition, a lexicon substitution, or an open-parameter substitution. To maintain ease of certain common code paradigms such as conditionals, map-reduce, and lambda expressions, they are assigned a low frequency factor (e.g. 0.1). Each of these rules r is assigned a frequency $f(r)$ (notation here is that f is defined on single items as well as sets of those items).

When multiple frequency-defined rules apply (such as self-composition with a function), the harmonic mean of applicable rules' frequencies is taken. For step $s = \{r_1, \ldots, r_n\}$,

$$f(s) = \frac{n}{\sum_{r \in s} \frac{1}{f(r)}}$$

Some functions, such as `map` must be expanded with function arguments in order to operate. These higher-order functions are expanded into immediate child possibilities, with the resulting step frequency ultimately being linearly halved for each higher-order step to a small pre-defined maximum depth (e.g. 2).

Normalizing over all transition steps for a particular program yields the proposal distribution for stochastic program synthesis in the Metropolis Hastings theory sampler.

8 Discussion

I have specified an algorithmic model for theory learning, based on the biphasic dichotomy of observation (*wake*) and learning (*sleep*). It uses a hierarchical Bayesian framework and stochastic theory generation in the space of a language-of-thought program to acquire theories, and uses theory confidence (sample model likelihood) to make informed samples to better the agent's future theory learning.

I believe the major step to improve this algorithm is abstract the two scales of theories and models into a hierarchy of any scale, which may have recursive qualities. If Θ parameters could be non-constant functions like those used in theories and could have their own lower-scale hierarchical state for parameters of those functions, then theory structures would be recursively learned rather than only flat low-level theories themselves tailored to a particular task/set of observations. This would primarily be useful if there were many different theories being learned, so such a paradigm is intuitive and serves a better computational model for intelligence and cognition.

References

[1] Ullman, T. D., Goodman, N. D., & Tenenbaum, J. B. (2012). Theory learning as stochastic search in a language of thought. *Cognitive Development* 27 (4), 455-480.

[2] Goodman, N. D., Tenenbaum, J. B., & Gerstenberg, T. (2014). Concepts in a probabilistic language of thought. *Concepts: New Directions*, MIT Press.

[3] Tenenbaum, J. B. & Griffiths, T. L. (2001). The rational basis of representativeness. *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* 1036-1041.

[4] Katz, Y., Goodman, N. D., Kersting, K., Kemp, C., & Tenenbaum, J. B. (2008). Modeling semantic cognition as logical dimensionality reduction. *Proceedings of the 13th Annual Conference of the Cognitive Science Society.*

[5] Tenenbaum, J. B. (1999). A bayesian framework for concept learning. *Ph.D. thesis, Massachusetts Institute of Technology.*

[6] Goodman, N. D., Tenenbaum J. B., Feldman J., & Griffiths, T. L. (2008b). A rational analysis of rule-based concept learning. *Cognitive Science* 32 (1), 108154.

[7] Goodman, N. D., Mansinghka, V. K., Roy, D. M., Bonawitz, K., & Tenenbaum, J. B. (2008a). Church: a language for generative models. *Uncertainty in Artificial Intelligence.*