Approximate Bayesian Inference for PCFGs

Bonan Zhao

1 Overview

Previously, we introduced probabilistic context-free grammars (PCFGs) as a way to define a prior over the hypothesis space in causal generalization. Given observed data, we can update our beliefs about these hypotheses using Bayes' rule, and then make generalization predictions marginalizing over the posterior beliefs of our causal theories of the world.

Activity 1. *Circle the correct one:*
In terms of making generalization predictions, we care more about the prior / posterior *distribution.*

When the hypothesis space $\mathcal{H}$ is very large or infinite (think of recursion!), exact computation is typically impossible. In this course, we will go through methods for approximating the posterior distribution over hypotheses. These methods are known as **approximate Bayesian inference**, or approximate Bayesian computation (shortened as ABC).

By the end of this course, you should be able to:

1. Implement a Metropolis–Hastings sampler over hypotheses generated by a PCFG.
2. Implement a sequential Monte Carlo method, also known as a particle filter, for approximate online Bayesian learning.

2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a class of algorithms used to draw samples from a probability distribution. When a probability distribution is too complex or too high dimensional (like for our grammar trees), analytical tools often fall short. Luckily, we can approximate a distribution by drawing samples in smart ways.

2.1 Markov Chains and Stationary Distributions

Since we do not know our target distribution, we cannot directly sample from it. What can we do?

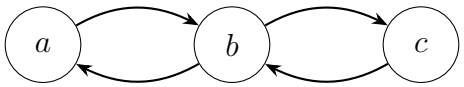
A useful analogy On the Tsinghua campus, students commute between dorm rooms and teaching buildings. On a Thursday morning, some moves from dorms to classrooms, some moves from classrooms to dorms. Individuals keep moving, but after enough time the fraction of people in each neighborhood may stabilize.

That stable population distribution is analogous to a *stationary distribution*.

Activity 2. *Why stationary distribution is useful to our problem?*

The students moving around scenario, under certain assumptions, forms a Markov chain.

Markov chains Imagine a particle moving among three states:



At each step it randomly moves to a neighboring state. If the particle is currently at b, the probability of its next move depends only on being at b, not on how it arrived there.

Formally, a sequence of random states $X_0, X_1, X_2, \ldots$ forms a *Markov chain*, if the next state depends only on the current state:

$$P(X_{t+1} = x' \mid X_t = x, X_{t-1}, \ldots, X_0) = P(X_{t+1} = x' \mid X_t = x).$$

This is called the *Markov property.*

Transition probabilities A Markov chain is specified by transition probabilities

$$T(x, x') = P(X_{t+1} = x' \mid X_t = x).$$

For a finite state space, these probabilities can be arranged into a transition matrix.

For example,

$$T = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.5 & 0.5 \end{pmatrix}.$$

The entry in row i, column j gives the probability of moving from state i to state j.

Activity 3. *Write the transition probabilities in this matrix onto the edges in the Figure above. If an edge is missing, draw it yourself, and add the corresponding transition probability.*

At any time, we may be uncertain about the chain's state. However, Let $p_t(x)$ denote the probability that the chain is in state x at time t. As the chain evolves, this distribution changes according to the transition probabilities. For example, if we start with all probability mass on state A, the distribution may gradually spread across the state space.

Stationary distributions A distribution π is called *stationary* if applying the transition rule leaves it unchanged:

$$\pi(x') \sum_x \pi(x) T(x, x').$$

In matrix notation,

$$\pi T = \pi.$$

Intuitively, if the chain starts with distribution π, it will still have distribution π after one step, two steps, or a million steps[1].

[1] This is only true when the Markov chain is finite, connected (i.e, for every state, it is possible to reach any other states), non-periodic (i.e, no oscillation between states).

Important idea. A stationary distribution is not a state. It is a probability distribution over states.

2.2 Metropolis–Hastings (MH)

Suppose we want to reason about a large set of possible hypotheses $h \in \mathcal{H}$. Since this space is enormous, it is impossible to enumerate all trees or sample directly from it. Metropolis–Hastings (MH) addresses this problem by constructing a Markov chain whose states are hypotheses and whose stationary distribution is the desired distribution.

The basic MH procedure. At each iteration, the algorithm performs the following steps:

1. Start with the current hypothesis h.
2. Propose a modified hypothesis h'.
3. Compare how well h' explains the data relative to h.
4. Accept the proposed hypothesis with a certain probability.
5. If the proposal is accepted, move to h'. Otherwise remain at h.

Proposal distribution is the probability of proposing h' when the current hypothesis is h:

$$q(h' \mid h).$$

The proposal distribution determines how the algorithm explores the hypothesis space.

Acceptance probability. After proposing h', we must decide whether to move there. The Metropolis–Hastings acceptance probability is

$$a(h \to h') = \min\left\{1, \frac{\pi(h')q(h \mid h')}{\pi(h)q(h' \mid h)}\right\},$$

where

- $a(h \to h')$ is the probability of accepting the proposed move.
- $\pi(h)$ is the target distribution evaluated at the current hypothesis.
- $\pi(h')$ is the target distribution evaluated at the proposed hypothesis.
- $q(h' \mid h)$ is the probability of proposing h' from h.
- $q(h \mid h')$ is the probability of proposing the reverse move.

Activity 4. *The acceptance probability is always between _____ and _____.*

Applying MH to Bayesian inference. In Bayesian inference, the target distribution is the posterior distribution:

$$\pi(h) = p(h \mid \mathcal{D}) \approx p(\mathcal{D} \mid h)p(h).$$

Activity 5. *In the MH acceptance probability, substitute the target distribution $\pi(h)$ with the posterior distribution $p(\mathcal{D} \mid h)p(h)$:*

$$a(h \to h') = \min\left\{1, \text{———————————}\right\}.$$

2.3 PCFG Hypotheses as Trees

Each of our hypotheses is essentially a derivation tree generated by a PCFG.

Tree-regeneration proposal A natural way to propose a new PCFG hypothesis is to modify part of its derivation tree. This is called a subtree-regeneration proposal.

Activity 6. *Without looking at the text below this activity, come up with at least two ways of hopping from one tree to another.*

Here is one recipe from Goodman et al. (2008): Given the current tree t:

1. Choose a node u in the tree.
2. Let the nonterminal at that node be A.
3. Delete the subtree rooted at u.
4. Regenerate a new subtree from nonterminal A using the PCFG.
5. Insert the new subtree back into the same location.

This produces a proposed tree t'.

Proposal probability Let $|t|$ be the number of selectable nonterminal nodes in tree t. Suppose the proposal chooses uniformly among all selectable nodes in t. If node u is chosen and its nonterminal label is A, then the proposal probability is

$$q(t' \mid t) = \frac{1}{|t|} p_{\mathcal{G}}(s'_u \mid A),$$

where:

- s'_u is the newly regenerated subtree;
- $p_{\mathcal{G}}(s'_u \mid A)$ is the PCFG probability of generating that subtree from A.

Activity 7. *The probability of regenerating the original subtree s_u from the newly regenerated subtree s'_u is*

$$q(t \mid t') =$$

Plugging these tree regeneration probabilities into the MH acceptance probability, eventually we get

$$a(t \to t') = \min\left\{1, \frac{p(\mathcal{D} \mid t')}{p(\mathcal{D} \mid t)} \frac{|t|}{|t'|}\right\}.$$

Summary Metropolis–Hastings replaces a difficult sampling problem

"Sample directly from $P(H \mid D)$"

with an easier one:

"Take many small random steps among trees."

By carefully choosing the acceptance rule, these local random steps eventually produce samples that approximate the distribution of interest.

Activity 8. *Optional: In your own words, why does this work?*

Activity 9. *Optional: MCMC sometimes accepts worse hypotheses, why?*

3 Particle Filtering / Sequential Monte Carlo

MCMC approximates a posterior after seeing all data at once. Particle filtering does the same thing *online*: it updates the posterior each time a new observation arrives.

Particles are weighted hypotheses that together approximate the posterior. Each particle j holds a hypothesis $h^{(j)}$ (a program, a causal rule, etc.) sampled from a PCFG, plus a weight $w^{(j)} \geq 0$ saying how plausible that hypothesis is. After seeing n observations:

$$\left\{(h_n^{(j)},\, w_n^{(j)})\right\}_{j=1}^{M}, \qquad \sum_{j=1}^{M} w_n^{(j)} = 1.$$

The weighted particle set approximates the posterior:

$$p(h \mid \mathcal{D}_n) \;\approx\; \sum_{j=1}^{M} w_n^{(j)}\, \delta_{h_n^{(j)}}(h),$$

where $\delta_{h^{(j)}}(h)$ is a point mass at $h^{(j)}$: it equals 1 if $h = h^{(j)}$ and 0 otherwise. The whole expression just says: put weight $w^{(j)}$ on the hypothesis $h^{(j)}$.

Initialisation Before seeing any data, sample each particle from the prior and weight them equally:

$$h_0^{(j)} \sim p(h), \qquad w_0^{(j)} = \tfrac{1}{M}.$$

Activity 10. *With $M = 4$ particles, fill in the weights for each particle in the table below:*

j	$h^{(j)}$	$w^{(j)}$
1	color(x) = ***red***	
2	shape(x) = ***square***	
3	color(x) = ***blue***	
4	shape(x) = ***triangle***	

Weight update When observation d_n arrives, evaluate how likely a particle is given data d_n (i.e., compute the likelihood $p(d_n \mid h_{n-1}^{(j)})$), and multiply this by each particle's current weight, then normalise:

$$\tilde{w}_n^{(j)} = w_{n-1}^{(j)}\, p(d_n \mid x_n, h_{n-1}^{(j)}), \qquad w_n^{(j)} = \frac{\tilde{w}_n^{(j)}}{\sum_\ell \tilde{w}_n^{(\ell)}}.$$

Activity 11. *Assume the our first observation d_1 = (**red square**, effect = 1). Assume a noisy likelihood such that a matching hypothesis predicts the positive label with probability* 0.95*, a non-matching one with probability* 0.05*. Filling the columns:*

j	$h^{(j)}$	$p(d_1 \mid h^{(j)})$	$w_1^{(j)}$
1	color = ***red***		
2	shape = ***square***		
3	color = ***blue***		
4	shape = ***triangle***		

Effective sample size (ESS) Over time weights can collapse: one particle ends up with nearly all the weight and the rest become negligible. A useful diagnostic is

$$\text{ESS} = \frac{1}{\sum_{j=1}^{M} (w_n^{(j)})^2}.$$

If all weights are equal, ESS $= M$. If one particle has weight 1, ESS $= 1$. We can set a threshold, for example $\frac{M}{2}$, and resample when ESS $< M/2$.

Activity 12. *Calculate the ESS for the table above. Do we resample?*

Activity 13. *Suppose we now see the second data point* $d_2 = (\texttt{red circle}, \textit{label=1})$. *Calculate the new weights for each particle and then calculate the ESS. Do we resample?*

Resampling The filter has essentially committed to particle 1, but we still have $M = 4$ slots. Resampling draws M new particles from the current weighted set at time n, giving high-weight particles more copies and discarding negligible ones. Draw ancestor indices

$$a_j \sim \text{Categorical}(w_n^{(1)}, \ldots, w_n^{(M)}),$$

then set

$$h_n^{(j)} \leftarrow h_n^{(a_j)}, \qquad w_n^{(j)} \leftarrow \tfrac{1}{M}.$$

In our example ($n = 2$), most resampled particles will be copies of color$(x) = \texttt{red}$.

Rejuvenation After resampling, many particles are duplicates—diversity is gone and the filter cannot explore. To fix this, run a few MH moves on each particle targeting the current posterior $p(h \mid \mathcal{D}_n)$. Concretely, apply subtree-regeneration and use the MH accept/reject step to get new particles. Applying several such moves to each particle. This preserves the target distribution while mutating copies into nearby alternatives, for example, from color$(x) = \texttt{red}$ we may get new particles:

$$\text{color}(x) = \texttt{red} \wedge \neg(\text{shape}(x) = \texttt{triangle}), \qquad \text{color}(x) = \texttt{red} \vee (\text{shape}(x) = \texttt{circle}).$$

Comment The empirical distribution of particles approaches the true posterior probability distribution as the number of particles approaches infinity. Is this a plausible thesis for human cognition?

References

Goodman, N. D., Tenenbaum, J. B., Feldman, J., and Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, 32, 108–154. doi:10.1080/03640210701802071.