

CS281A/Stat241A recitation 11: Gibbs sampling and variants

November 12, 2007

Percy Liang

Gibbs sampling

- Goals:

- Sampling is generally used for Monte Carlo integration (to approximate expectations stochastically).

$$\mathbb{E}f(Z) \approx \frac{1}{T} \sum_{t=1}^T f(z^{(t)}),$$

where $z^{(1)}, \dots, z^{(n)}$ are i.i.d. draws. The RHS converges to the LHS by the Law of Large Numbers. For discrete variables, being able to sample i.i.d. efficiently is equivalent to being able to compute expectations efficiently.

- MCMC sampling is where $z^{(1)}, \dots, z^{(T)}$ are not i.i.d., but a Markov chain converging to the stationary distribution). People in practice use this as a (usually local) search procedure (think stochastic hill-climbing, simulated annealing).
- Example uses in machine learning:
 - * Situation: we can only evaluate $p(z) = \frac{1}{Z} \prod_c \psi_c(z)$ up to a constant
 - * Where are we computing expectations? compute expected sufficient statistics in E-step for hidden-variable models.
 - * Bayesian inference: sample both hidden variables (if any) and parameters.

- Properties

- General MCMC sampling: random walk on states of a graph
- Conditions: irreducible, aperiodic, reversibility

$$p(z) = \sum_{z'} p(z') T(z', z)$$

- Converges to the correct answer in the limit, but this could take exponential time
- Mixing time characterizes time required for a sample to be close to the stationary distribution (as measured by variational distance)
- Worse, its impossible to diagnose convergence
- Example: bottleneck graph, Ising model

- When is sampling preferable compared to other methods (exact junction tree, variational methods)?
 - Loopy graphical model (high tree-width); for Ising model, exact inference takes $O(n2^n)$, but one round of Gibbs sampling takes only $O(n^2)$
 - Can be useful even when E-step would be tractable; for a chain, forward-backward takes $O(nk^2)$ time, one iteration of Gibbs sampling takes $O(nk)$
 - Sampling is simple to code
 - Allows computing many expectations of all sorts of functions, not just ones that conform to the graphical model structure
- Recipe for creating a Gibbs sampler

$$\mathbf{z}_{-i} = (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n)$$

$$z_i^{(t+1)} \sim p(z_i \mid \mathbf{z}_{-i}^{(t)}) \quad z_j^{(t+1)} = z_j^{(t)} \text{ for } j \neq i$$

- For each node:
 - * Write down only factors that depend on the node you’re sampling (we only need conditional up to a proportionality constant)
 - * Normalize to get probability distribution
 - * Sample the value of node
- How to marginalize variables?
 - Gibbs sampling gives samples of (z_1, \dots, z_n)
 - How to compute $p(z_i)$?
 - Marginalization is easy: just throw away the variables being marginalized out

Methods for improve efficiency

- Rao-Blackwellization
 - Idea: even though we choose one value for the sample, we have to consider all possible values to compute the correct probabilities; currently, these are just thrown away; we can actually make use of that information via these “distributional samples”. However, for the next iteration, we still have to commit to a single value.
 - Suppose in Gibbs sampling, we sample from $z_i \sim p(z_i \mid \mathbf{z}_{-i})$, and we are interested in computing node marginals $p(z_i = k)$
 - Let S_i be the set of times $1 \leq t \leq T$ such that node i was sampled

- Instead of computing

$$p(z_i = k) \cong \frac{1}{T} \sum_{t=1}^T \mathbb{1}[z_i^{(t)} = k] \cong \frac{1}{|S_i|} \sum_{t \in S_i} \mathbb{1}[z_i^{(t)} = k]$$

a more accurate estimate (lower variance) would be

$$p(z_i = k) \cong \frac{1}{|S_i|} \sum_{t \in S_i} p(z_i^{(t)} = k \mid \mathbf{z}_{-i}^{(t)})$$

- Combining several kernel functions

- Cycles of samplers (kernels): loop through in fixed order
- Mixtures of samplers (kernels): keep on picking random sampler and use it

$$T(z, z') = \sum_k \alpha_k T_k(z, z')$$

- Caution: choosing which kernel *should not* depend on the current state

- Blocked sampling

- When potentials are strong, Gibbs sampling works poorly
- Extreme case: potential says two nodes must have same value
- Solution: sample multiple nodes at the same time
- One option: can sample trees; two interleaved trees is better than checkerboard pattern?

- Metropolis-Hastings

- Define a proposal $q(z' \mid z)$
- MH algorithm:
 - * $z' \sim q(z' \mid z^{(t)})$
 - * Accept ($z^{(t+1)} \leftarrow z'$) with probability

$$\min \left\{ 1, \frac{p(z')}{p(z^{(t)})} \frac{q(z^{(t)} \mid z')}{q(z' \mid z^{(t)})} \right\}$$

- Example: chain with a few long range dependencies

- Collapsed sampling: remove variables to improve efficiency

- Suppose $z = (z_A, z_B)$; instead of sampling from $p(z)$, sample from $p(z_A) = \sum_{z_B} p(z)$

- Example:

$$p(\theta, z_1, \dots, z_n) = \text{Dir}(\theta; \alpha) \prod_{i=1}^n \text{Mult}(z_i; \theta)$$

- Non-collapsed sampling: cycle through sampling θ, z_1, \dots, z_n
- Collapsed sampling:

$$p(z_i = k \mid \mathbf{z}_{-i}) = \frac{\sum_{j \neq i} \mathbb{1}[z_j = k]}{n - 1}$$

- In practice, we have likelihood terms $p(x_i \mid z_i)$:

$$p(z_i = k \mid \mathbf{z}_{-i}, \mathbf{x}) \propto \frac{\sum_{j \neq i} \mathbb{1}[z_j = k]}{n - 1} p(x_i \mid z_i = k)$$

- Tomorrow: collapsed sampling for LDA is based on this idea

- Auxiliary variable sampling: add variables to improve efficiency
 - Augment state space from z to (z, a)
 - Define $p(z, a)$ such that $\int_a p(z, a)$ is the desired $p(z)$
 - Now, any valid sampler with stationary distribution $p(z, a)$ will be good for sampling $p(z)$ (remember, just drop a)
 - This is useful because developing good MCMC samplers for $p(z \mid a)$ is easier than for $p(z)$
 - Example: slice sampling

$$p(z, a) = \mathbb{1}[0 \leq a \leq p(z)]$$

In practice, we can't evaluate $p(z)$

$$p(z) = \frac{1}{Z} \prod_c \psi_c(z)$$

$$p(z, a) = \frac{1}{Z} \prod_c \mathbb{1}[0 \leq a_c \leq \psi_c(z)]$$

- * Sampling a (trivial): just draw a uniform variable between 0 and $\psi_c(z)$
- * Sampling z (easier): only have to consider values of z with $\psi_c(z) \geq a_c$ for all cliques c (important: consider them with equal probability; the fact that they have different $p(z)$ is already taken into account by conditioning on a)
- Example of slice sampling: Swendsen-Wang for the Ising model
 - * For each edge connecting two nodes with the same label, independently choose place a bond between them or not
 - * Form connected components: put bonded nodes in same connected component
 - * For each connected component, choose label ± 1 with equal probability