Inference via Sampling (Contd)

Piyush Rai

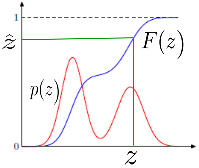
Topics in Probabilistic Modeling and Inference (CS698X)

March 2, 2019

Recap: Basic Sampling Methods

- Inverse CDF method. Assume $F(z)$ to be CDF of our distribution of interest $p(z)$

$$\hat{z} \sim \text{Uniform}(0,1)$$
$$z = F^{-1}(\hat{z})$$



- Reparametrization method (also used in VI - pathwise gradient methods), e.g.,

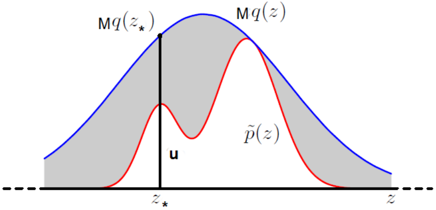
$$\hat{z} \sim \mathcal{N}(0,1) \Rightarrow z = \mu + \sigma \hat{z} \sim \mathcal{N}(\mu, \sigma^2)$$

- Note: The above are examples of the more general idea of transformation of distributions

$$p(\mathbf{z}) = q(\hat{\mathbf{z}}) \left| \frac{\partial \hat{\mathbf{z}}}{\partial \mathbf{z}} \right|$$

.. where $\left| \frac{\partial \hat{\mathbf{z}}}{\partial \mathbf{z}} \right|$ is the determinant of the Jacobian

Recap: Basic Sampling Methods



- Rejection Sampling: Sample from $p(z) = \frac{\tilde{p}(z)}{Z_p}$ by sampling from $q(z)$, s.t., $Mq(z) \geq \tilde{p}(z)$
  - $z_* \sim q(z)$ and $u \sim \text{Uniform}(0, Mq(z))$
  - If $u \leq \tilde{p}(z_*)$, accept z_* else reject
  - Repeat the above two steps until we have generated the desired number of samples

Computing Expectations via Monte Carlo Sampling

- Often we are interested in computing expectations of the form

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

where $f(\boldsymbol{z})$ is some function of a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$

- A simple approximation scheme: Monte Carlo integration
- Suppose we can generate L independent samples from $p(\boldsymbol{z})$: $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L} \sim p(\boldsymbol{z})$
- Monte-Carlo approximation replaces the expectation by an empirical average

$$\hat{f} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})$$

- Since the samples are independent of each other, can show the following (exercise)

$$\mathbb{E}[\hat{f}] = \mathbb{E}[f] \qquad \text{and} \qquad \text{var}[\hat{f}] = \frac{1}{L}\text{var}[f] = \frac{1}{L}\mathbb{E}[(f - \mathbb{E}[f])^2]$$

- Note that the variance in the estimate of expectation decreases as L increases

Computing Expectations via Importance Sampling

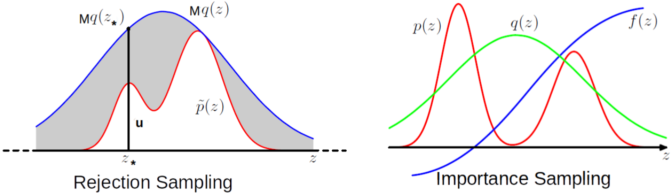
- Monte Carlo assumes we know how to generate samples from $p(\boldsymbol{z})$. What if we don't know?
- Transformation methods can be one way to handle this situation
- Importance Sampling is another way: Generate from a "proposal" $q(\boldsymbol{z})$, i.e., $\{\boldsymbol{z}^{(\ell)}\}_{\ell=1}^{L} \sim q(\boldsymbol{z})$
- Additionally, suppose we can evaluate $p(\boldsymbol{z})$ at any given $\boldsymbol{z}$
- Importance Sampling then approximates the original expectation as

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} = \int f(\boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}q(\boldsymbol{z})d\boldsymbol{z} \approx \frac{1}{L}\sum_{\ell=1}^{L} f(\boldsymbol{z}^{(\ell)})\frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$$

- This is basically "weighted" Monte Carlo integration
  - $w_\ell = \frac{p(\boldsymbol{z}^{(\ell)})}{q(\boldsymbol{z}^{(\ell)})}$ denotes the importance weight of each sample $\boldsymbol{z}^{(\ell)}$
- Works even when we can evaluate $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$ only up to a prop. constant (PRML 11.1.4)
- Note: Monte Carlo and Importance Sampling are NOT sampling methods!
  - .. that is, not used for generating samples but only for computing expectations using samples

Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions



- Difficult to find good prop. distr. especially when $\boldsymbol{z}$ is high-dim. (e.g., models with many params)
  - In high dimensions, most of the mass of $p(\boldsymbol{z})$ is concentrated in a tiny region of the $\boldsymbol{z}$ space
  - Difficult to *a priori* know what those regions are, thus difficult to come up with good proposal dist.
- A solution to these: MCMC methods

Markov Chain Monte Carlo (MCMC)

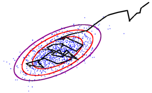
- Goal: Generate samples from some target distribution $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z}$, where $\boldsymbol{z}$ is high-dimensional
- Assume we can evaluate $p(\boldsymbol{z})$ at least up to a proportionality constant (i.e., can compute $\tilde{p}(\boldsymbol{z})$)
- Basic idea: MCMC uses a Markov Chain which, when converged, starts giving samples from $p(\boldsymbol{z})$

$$\underbrace{\boldsymbol{z}^{(1)} \rightarrow \boldsymbol{z}^{(2)} \rightarrow \boldsymbol{z}^{(3)} \rightarrow}_{\text{initial samples typically garbage}} \ldots \rightarrow \underbrace{\boldsymbol{z}^{(L-2)} \rightarrow \boldsymbol{z}^{(L-1)} \rightarrow \boldsymbol{z}^{(L)}}_{\text{after convergence, actual samples from } p(\boldsymbol{z})}$$

- Given a current sample $\boldsymbol{z}^{(\ell)}$ from the chain, MCMC generates the next sample $\boldsymbol{z}^{(\ell+1)}$ as
  - Use a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ to generate a candidate sample $\boldsymbol{z}^*$
  - Accept/reject $\boldsymbol{z}^*$ as the next sample based on an acceptance criterion (will see later)
  - If accepted, $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^*$. If rejected, $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$
- Note that in MCMC, the proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ depends on the previous sample (unlike methods such as rejection sampling)

MCMC: The Basic Scheme

- MCMC chain run infinitely long (i.e., post-convergence) will give ONE sample from the target $p(\boldsymbol{z})$



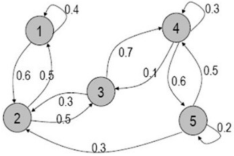
- But we usually require several samples to approximate $p(\boldsymbol{z})$. How do we get those?
  - Start at an initial $\boldsymbol{z}^{(0)}$. Using a prop. dist. $q(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$, run the chain long enough, say T_1 steps
  - Discard the first $(T_1 - 1)$ samples (called "burn-in" samples) and take the last sample $\boldsymbol{z}^{(T_1)}$
  - Continue from $\boldsymbol{z}^{(T_1)}$ up to T_2 steps, discard intermediate samples, take the last sample $\boldsymbol{z}^{(T_2)}$
    - This helps ensure that $\boldsymbol{z}^{(T_1)}$ and $\boldsymbol{z}^{(T_2)}$ are uncorrelated
  - Repeat the same for a total of S times
  - In the end, we have S i.i.d. samples from $p(\boldsymbol{z})$, i.e., $\boldsymbol{z}^{(T_1)}, \boldsymbol{z}^{(T_2)}, \ldots, \boldsymbol{z}^{(T_S)} \sim p(\boldsymbol{z})$
  - Note: Good choices for T_1 and $T_i - T_{i-1}$ are usually based on heuristics
  - Note: MCMC is an approximate method because we don't usually know what T_1 is "long enough"

MCMC: Some Basic Theory

- A first order Markov Chain assumes $p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(\ell)}) = p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$
- A 1st order Markov Chain $\boldsymbol{z}^{(0)},\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(L)}$ is a sequence of r.v.'s and is defined by
  - An initial state distribution $p(z^{(0)})$
  - A Transition Function (TF): $T_\ell(\boldsymbol{z}^{(\ell)} \rightarrow \boldsymbol{z}^{(\ell+1)}) = p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$.
- TF defines a distribution over the values of next state given the value of the current state
- Assuming a discrete state-space, the TF is defined by a $K \times K$ probability table

Transition probabilities can be defined using a *K*x*K* table if **z** is a discrete r.v. with *K* possible values

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0.4 & 0.6 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \\ 0.0 & 0.3 & 0.0 & 0.5 & 0.2 \end{bmatrix} \end{array}$$

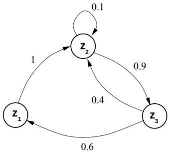


- Homogeneous Markov Chain: The TF is the same for all ℓ, i.e., $T_\ell = T$

MCMC: Some Basic Theory

- Consider the following simple TF with $K = 3$ (want to sample from a multinoulli)

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$



- Consider the initial state distribution $p(\boldsymbol{z}^{(0)}) = p(z_1^{(0)}, z_2^{(0)}, z_3^{(0)}) = [0.5, 0.2, 0.3]$
- Easy to see that $p(\boldsymbol{z}^{(0)}) \times T = [0.2, 0.6, 0.2] \Rightarrow$ distribution of $\boldsymbol{z}^{(1)}$
- Also easy to see that, after a few (say m) iterations, $p(\boldsymbol{z}^{(0)}) \times T^m = [0.2, 0.4, 0.4] = p(\boldsymbol{z})$ (say)
- For the above T, <u>any choice</u> of $p(\boldsymbol{z}^{(0)})$ leads to multinoulli $p(\boldsymbol{z})$i with $\pi = [0.2, 0.4, 0.4]$
  - Such a $p(\boldsymbol{z})$ is called the stationary/invariant distribution of this Markov Chain
- A Markov Chain has a stationary distribution if T has the following properties
  - Irreducibility: T's graph is connected (ensures reachability from anywhere to anywhere)
  - Aperiodicity: T's graph has no cycles (ensures that the chain isn't trapped in cycles)

MCMC: Some Basic Theory

- A sufficient (but not necessary) condition: A Markov Chain with transition function T has stationary distribution $p(\boldsymbol{z})$ if T satisfies Detailed Balance
- For any two states $\boldsymbol{z}$ and $\boldsymbol{z}'$, the Detailed Balanced condition is

$$p(\boldsymbol{z})T(\boldsymbol{z} \rightarrow \boldsymbol{z}') = p(\boldsymbol{z}')T(\boldsymbol{z}' \rightarrow \boldsymbol{z})$$

- Integrating out (or summing over) both sides w.r.t. $\boldsymbol{z}'$ gives

$$p(\boldsymbol{z}) = \int p(\boldsymbol{z}')T(\boldsymbol{z}' \rightarrow \boldsymbol{z})d\boldsymbol{z}'$$

- Therefore $p(\boldsymbol{z})$ is a stationary distribution of this chain
- Thus a Markov Chain with detailed balance will always converge to a stationary distribution

Some MCMC Algorithms

Metropolis-Hastings (MH) Sampling (Hastings, 1970)

- Suppose we wish to generate samples from a distribution $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$
- Assume a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$, e.g., $\mathcal{N}(\boldsymbol{z}|\boldsymbol{z}^{(\tau)}, \sigma^2\mathbf{I}_D)$
- In each step, draw $\boldsymbol{z}^* \sim q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$ and accept $\boldsymbol{z}^*$ with probability

$$A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})}\right)$$

- The acceptance probability makes intuitive sense
  - It favors accepting $\boldsymbol{z}^*$ if $\tilde{p}(\boldsymbol{z}^*)$ has a higher value than $\tilde{p}(\boldsymbol{z}^{(\tau)})$
  - Unfavors $\boldsymbol{z}^*$ if the proposal distribution q unduly favors its generation (i.e., if $q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$ is large)
  - Favors $\boldsymbol{z}^*$ if we can "reverse" to $\boldsymbol{z}^{(\tau)}$ from $\boldsymbol{z}^*$ (i.e., if $q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^*)$ is large). Needed for good "mixing"
- Transition function of this Markov Chain: $T(\boldsymbol{z}^{(\tau)} \rightarrow \boldsymbol{z}^*) = A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$
- Exercise: Show that $T(\boldsymbol{z} \rightarrow \boldsymbol{z}^{(\tau)})$ satisfies the detailed balance property

$$T(\boldsymbol{z} \rightarrow \boldsymbol{z}^{(\tau)})p(\boldsymbol{z}) = T(\boldsymbol{z}^{(\tau)} \rightarrow \boldsymbol{z})p(\boldsymbol{z}^{(\tau)})$$

The MH Sampling Algorithm

- Initialize $\boldsymbol{z}^{(0)}$ randomly
- For $\ell = 0, \ldots, L-1$
  - Sample $\boldsymbol{z}^* \sim q(\boldsymbol{z}^*|\boldsymbol{z}^{(\ell)})$ and $u \sim \text{Unif}(0,1)$
  - If $u < A(\boldsymbol{z}^*, \boldsymbol{z}^{(\ell)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)q(\boldsymbol{z}^{(\ell)}|\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\ell)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\ell)})}\right)$

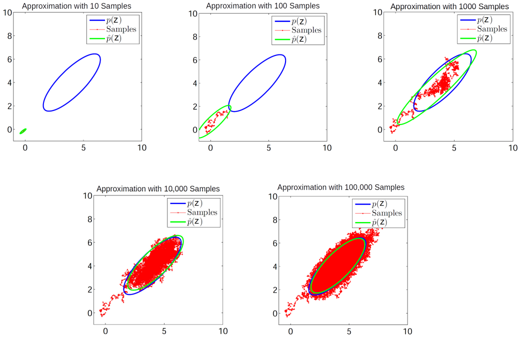
$$\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^* \qquad \text{(meaning: accepting with probability } A(\boldsymbol{z}^*, \boldsymbol{z}^{(\ell)})\text{)}$$

    else

$$\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$$

MH Sampling in Action: A Toy Example..

$$\text{Target } p(\boldsymbol{z}) = \mathcal{N}\left(\begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}1 & 0.8\\0.8 & 1\end{bmatrix}\right), \text{ Proposal } q(\boldsymbol{z}^{(t)}|\boldsymbol{z}^{(t-1)}) = \mathcal{N}\left(\boldsymbol{z}^{(t-1)}, \begin{bmatrix}0.01 & 0\\0 & 0.01\end{bmatrix}\right)$$

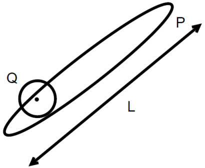


MH Sampling: Some Comments

- If proposal distrib. is symmetric, we get Metropolis Sampling algorithm (Metropolis, 1953) with

$$A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\tau)})}\right)$$

- Some limitations of MH sampling
  - MH can have a very slow convergence. Figure below: P is the target dist., Q is the proposal



Generic proposals use
$Q(x'; x) = \mathcal{N}(x, \sigma^2)$

σ large $\rightarrow$ many rejections

σ small $\rightarrow$ slow diffusion:
$\sim(L/\sigma)^2$ iterations required

  - Computing acceptance probability can be expensive: When $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$ represents a posterior distribution of some model, $\tilde{p}$ is the unnormalized posterior that depends on all the data (note: a lot of recent work on speeding up this step using subsets of data*)

* Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget (Korattikara et al, 2014)

Gibbs Sampling (Geman & Geman, 1984)

- Suppose we wish to sample from a joint distribution $p(\boldsymbol{z})$ where $\boldsymbol{z} = (z_1, z_2, \ldots, z_M)$
- However, suppose we can't sample from $p(\boldsymbol{z})$ but can sample from each conditional $p(z_i|\boldsymbol{z}_{-i})$
  - Can we done easily if we have a locally conjugate model
- For Gibbs sampling, the proposal is the conditional distribution $p(z_i|\boldsymbol{z}_{-i})$
- Gibbs sampling samples from these conditionals in a cyclic order
- Gibbs sampling is equivalent to Metropolis Hastings sampling with acceptance prob. $= 1$

$$A(\boldsymbol{z}^*, \boldsymbol{z}) = \frac{p(\boldsymbol{z}^*)q(\boldsymbol{z}|\boldsymbol{z}^*)}{p(\boldsymbol{z})q(\boldsymbol{z}^*|\boldsymbol{z})} = \frac{p(z_i^*|\boldsymbol{z}^*_{-i})p(\boldsymbol{z}^*_{-i})p(z_i|\boldsymbol{z}^*_{-i})}{p(z_i|\boldsymbol{z}_{-i})p(\boldsymbol{z}_{-i})p(z_i^*|\boldsymbol{z}_{-i})} = 1$$

where we use the fact that $\boldsymbol{z}^*_{-i} = \boldsymbol{z}_{-i}$

Gibbs Sampling: Sketch of the Algorithm

M: Total number of variables, T: number of Gibbs sampling steps

1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.

     $\vdots$
   - Sample $z_j^{(\tau+1)} \sim p(z_j|z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.

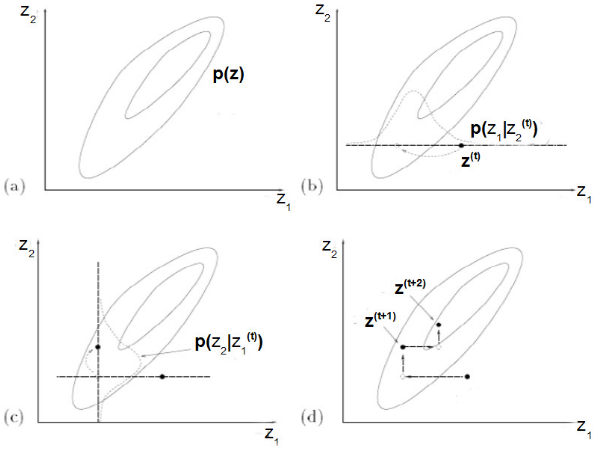
     $\vdots$
   - Sample $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.

Note: When sampling each variable from its conditional posterior, we use the most recent values of all other variables (this is akin to a co-ordinate ascent like procedure)

Note: Order of updating the variables *usually* doesn't matter (but see "Scan Order in Gibbs Sampling: Models in Which it Matters and Bounds on How Much" from NIPS 2016)

Gibbs Sampling: A Simple Example

Can sample from a 2-D Gaussian using 1-D Gaussians (recall that if the joint distribution is a 2-D Gaussian, conditionals will simply be 1-D Gaussians)



Gibbs Sampling: Some Comments

- One of the most popular MCMC algorithm
- Very easy to derive and implement for locally conjugate models
- Many variations exist, e.g.,
  - Blocked Gibbs: sample multiple variables jointly (sometimes possible)
  - Rao-Blackwellized Gibbs: Can collapse (i.e., integrate out) the unneeded variables while sampling. Also called "collapsed" Gibbs sampling
  - MH within Gibbs
- Instead of sampling from the conditionals, an alternative is to use the mode of the conditional.
  - Called the "Iterative Conditional Mode" (ICM) algorithm (doesn't give the posterior though)

Next Class

- Using posterior's gradient info in sampling algorithms
- Online MCMC algorithms
- Recent advances in MCMC
- Some other practical issues