

Theoretic Fundamentals of Machine and Deep Learning

Markov Chain Monte Carlo

Aleksandr Petiushko

Lomonosov MSU, Faculty of Mechanics and Mathematics
MIPT, RAIRI
Nuro, Autonomy Interaction Research

Winter-Spring, 2023



Content

- ➊ Markov Chain
- ➋ Stationary Distribution
- ➌ Markov Chain Monte Carlo
- ➍ Gibbs sampler
- ➎ Metropolis-Hastings sampler
- ➏ Langevin dynamics, SGLD and its Metropolis Adjustment

Motivation

- Variational Inference recap: it is an optimization process of an approximation of one distribution by another parameterized one
- VI is sample-efficient during inference because we sample from the converged distribution (that we learned)
- VI can provide the low-variance (see above item) but probably highly biased solution (because it is hard to provide a good family of distributions to cover the original one)
- Q: Can we reduce the bias (even by sacrificing the low-variance behavior)?
- A: It is possible by so called MCMC-methods that are essentially the sequential sampler having the theoretical guarantees on convergence to the original distribution but having high variance because of their sampling nature

Definitions

- Markov process: *stochastic* process $X_t, t \in \mathbb{R}_+$
- Markov chain (MC) — discrete variant of it: $X_n, n \in \mathbb{N} \cup \{0\}$

Definition

A **discrete-time Markov chain** (DTMC) is a sequence of random variables X_0, X_1, X_2, \dots with the **Markov property**: probability of moving to the next state depends only on the present state (not on the previous states):

$$P(X_{n+1} = x | X_0 = x_0, \dots, X_n = x_n) = P(X_{n+1} = x | X_n = x_n)$$

(if $P(X_0 = x_0, \dots, X_n = x_n) > 0$)

Remark: Hereafter we'll consider only the case where the state space $S = \{X_n\}_{n=0}^\infty$ is finite: $|S| = D < \infty$

Definitions

Definition

Time-homogeneous Markov chains:

$$P(X_{n+1} = a | X_n = b) = P(X_n = a | X_{n-1} = b) \quad \forall n > 1.$$

Remark: Hereafter we'll consider only the case of time-homogeneous MC.

Definition

Transition Probability Matrix: $D \times D$ matrix P : $P_{ij} = P(X_{n+1} = j | X_n = i)$.

Note: $\sum_{j=1}^D P_{ij} = 1$ for all $i = 1, \dots, D$.

Definitions

Definition

Stationary distribution π : a row vector

$\pi = (\pi_1, \dots, \pi_D)$, $\sum_{i=1}^D \pi_i = 1$, $\pi_i \geq 0, i = 1, \dots, D$ such as it is unchanged under transition matrix P : $\pi P = \pi$.

Theorem

Under some conditions (*irreducibility* and *aperiodicity*), there is a **unique stationary distribution** π (also called **equilibrium** distribution), and $\lim_{k \rightarrow \infty} P^k = \mathbf{1}\pi$ ($\mathbf{1}$ is the column vector with all entries equal to 1). This is also called **ergodic** MC.

Note: It means that the probability π_j of being in state j at the limit doesn't depend on the initial state i

Definitions

Definition

MC is called **reversible** if it satisfies the **detailed balance equations**:

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \forall i, j = 1, \dots, D$$

Theorem

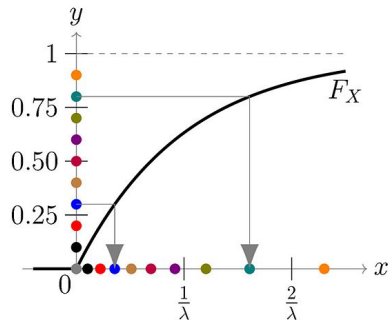
Distribution π in the reversible MC definition is the stationary distribution.

Proof: $\sum_i \pi_i P_{ij} = \sum_i \pi_j P_{ji} = \pi_j \sum_i P_{ji} = \pi_j \Rightarrow \pi P = \pi$.

Note: Transition matrices that are symmetric ($P_{ij} = P_{ji}$) always have detailed balance, and a uniform distribution over the states is an equilibrium distribution π .

Inverse transform sampling¹

- How to make a random sampler for any arbitrary distribution P with cumulative distribution function (cdf) F ?
- Suppose that we have a uniform sampler U from interval $[0, 1]$ — common case for every programming language
- Obvious approach: having sampled random value $u \sim U[0, 1]$, return the largest number x from the distribution P such that $P(-\infty < X < x) \leq u$
- The main idea is to use the inverse of cdf F^{-1} : first sample u , then $x = F^{-1}(u)$

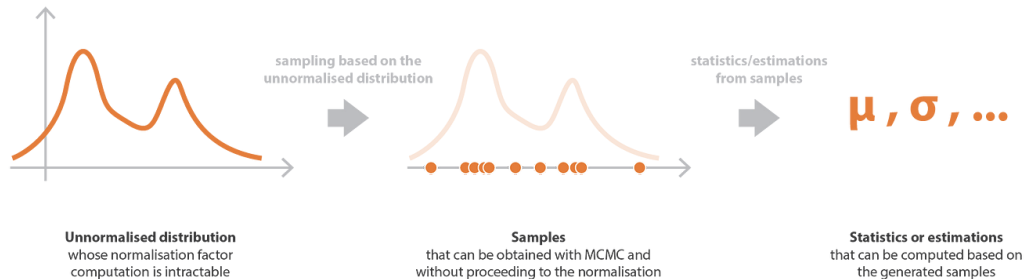


¹Wiki

Markov Chain Monte Carlo sampling

- Let's assume that we have **multi**-dimensional random value (r.v.) $X \in \mathbb{R}^n$ from multivariate probability distribution P
- Also in practice we either do not have simple formula for F^{-1} or we even need to work on top of empirical distribution P
- In this case the inverse transform method is usually inapplicable
- Solution: to use the **Markov Chain Monte Carlo** (MCMC) methods
- Idea: by constructing an MC that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain; the more steps, the closer the distribution of the sample matches the actual desired distribution.
 - ▶ Final equilibrium distribution is independent on the initial one

MCMC process illustration²



²Joseph Rocca's blog

Gibbs sampling³

- $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ from multivariate probability distribution $P = p(x_1, \dots, p_n)$

Gibbs Sampler

- Initialization: $x^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$
- Sampling: in order starting from the first component (or randomly), i.e.
 $x_j^{(i+1)} \sim p(x_j^{(i+1)} | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i)}, \dots, x_n^{(i)})$ for all $j = 1, \dots, n$
- Repeat until stopping condition (e.g., number of iterations $i = k$)

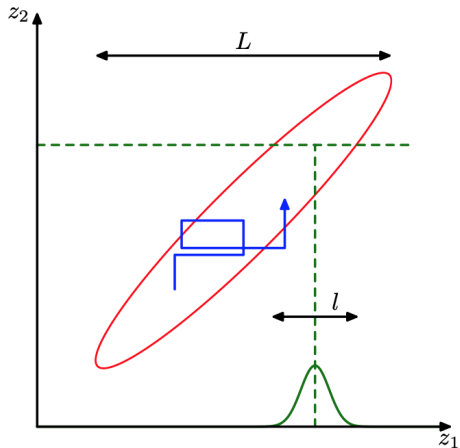
Note. The conditional distribution of one variable given all others is proportional to the **joint distribution**: $p(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) = \frac{p(x_1, \dots, x_n)}{p(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)} \propto p(x_1, \dots, x_n)$

³Geman, Stuart, and Donald Geman. “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images” 1984

Gibbs: Math behind

- Suppose we are picking index j randomly on every step and the new value $x_j \sim p(x_1, \dots, x_{j-1}, \cdot, x_{j+1}, \dots, x_n)$
- Such a process defines a reversible MC with the equilibrium distribution $p(x)$, because
 - ▶ $p(x^{(i+1)} = b | x^{(i)} = a) = \frac{1}{n} \frac{p(b)}{\sum_{c: a_k = c_k \forall k \neq j} p(c)}$ if $a_k = b_k \forall k \neq j$, and $p(x^{(i+1)} = b | x^{(i)} = a) = 0$ otherwise
 - ▶ $p(a)p(x^{(i+1)} = b | x^{(i)} = a) = \frac{1}{n} \frac{p(a)p(b)}{\sum_{c: a_k = c_k \forall k \neq j} p(c)} = \frac{1}{n} \frac{p(b)p(a)}{\sum_{c: b_k = c_k \forall k \neq j} p(c)} = p(b)p(x^{(i+1)} = a | x^{(i)} = b)$
- Balanced equations hold

Gibbs walk illustration⁴



⁴Bishop, C.M. Pattern Recognition and Machine Learning, Springer, 2006.

Metropolis⁵-Hastings⁶ sampling

- More sophisticated approach where we are working with $f(x) \propto p(x)$ up to some normalizing constant
- Suppose we have some generating distribution: $y \sim g(y|x)$ (in Metropolis case we have symmetrical generating distribution $g(y|x) = g(x|y)$)
- Let us find the ratio $r(y|x) = \frac{f(y)}{f(x)} \frac{g(x|y)}{g(y|x)} = \frac{p(y)}{p(x)} \frac{g(x|y)}{g(y|x)}$, the ratio
$$r(x|y) = \frac{1}{r(y|x)} = \frac{p(x)}{p(y)} \frac{g(y|x)}{g(x|y)}$$
- Let us accept the new $x^{(i+1)} = y$ with probability $A(y|x)$ from acceptance distribution $A(y|x) = \min(1, r(y|x))$;
 - ▶ Otherwise keep the old value $x^{(i+1)} = x^{(i)} = x$
 - ▶ It is usually done through sampling uniform r.v. $u \sim U[0, 1]$ and comparing with $A(y|x)$

⁵Metropolis, Nicholas, et al. "Equation of state calculations by fast computing machines" 1953

⁶Hastings, W. Keith. "Monte Carlo sampling methods using Markov chains and their applications" 1970

Metropolis-Hastings: Math behind⁷

- Transition probability $P(X^{(i+1)} = y | X^{(i)} = x) = p(y|x) = g(y|x)A(y|x)$
- If $r(y|x) \leq 1$, then $A(y|x) = r(y|x)$ and $A(x|y) = \min(1, r(x|y)) = \min(1, \frac{1}{r(y|x)}) = 1$
 - ▶ Then detailed balance conditions become $p(x)p(y|x) = p(x)g(y|x)A(y|x) = p(x)g(y|x)r(y|x) = p(x)g(y|x)\frac{p(y)}{p(x)}\frac{g(x|y)}{g(y|x)} = p(y)g(x|y) = p(y)g(x|y)A(x|y) = p(y)p(x|y)$
- If $r(y|x) > 1$, then $A(y|x) = 1$ and $A(x|y) = r(x|y)$
 - ▶ Then detailed balance conditions become $p(x)p(y|x) = p(x)g(y|x)A(y|x) = p(x)g(y|x) = p(y)g(x|y)\frac{p(x)}{p(y)}\frac{g(y|x)}{g(x|y)} = p(y)g(x|y)r(x|y) = p(y)g(x|y)A(x|y) = p(y)p(x|y)$
- So the MC is reversible and $f(x) \propto p(x)$ becomes the stationary distribution

⁷Geyer, Charles J. "Introduction to markov chain monte carlo." Handbook of markov chain monte carlo, **AP** 2011

Gibbs is MH

Gibbs sampler is a particular case of MH sampler:

- Let us generate y from x by Gibbs sampler, which means that for some $j, 1 \leq j \leq n$
 $y_{\neg j} = x_{\neg j} = z$, $p(y) = p(y_j|y_{\neg j})p(y_{\neg j}) = p(y_j|z)p(z)$,
 $p(x) = p(x_j|x_{\neg j})p(x_{\neg j}) = p(x_j|z)p(z)$ and the proposal generation probabilities are
 $g(y|x) = p(y_j|x_{\neg j}) = p(y_j|z)$, $g(x|y) = p(x_j|y_{\neg j}) = p(x_j|z)$
 - Then the ratio $r(y|x) = \frac{p(y)}{p(x)} \frac{g(x|y)}{g(y|x)} = \frac{p(y_j|z)p(z) \cdot p(x_j|z)}{p(x_j|z)p(z) \cdot p(y_j|z)} = 1$
 - And acceptance distribution $A(y|x) = \min(1, r(y|x)) = \min(1, 1) = 1$
- \Rightarrow Gibbs Sampler is the MH Sampler with acceptance probability 1.

Comparison of Gibbs and MH samplers

Gibbs sampler has the advantage over MH sampler because:

- No rejection step: every transition is accepted
- No need to tune the proposal distribution $g(y|x)$

Gibbs sampler has the disadvantage over MH sampler because:

- We need somehow to derive the true conditional probability distributions $p(x_j^{(i+1)} | x_1^{(i)}, \dots, x_{j-1}^{(i)}, x_{j+1}^{(i)}, \dots, x_n^{(i)})$ for all $j = 1, \dots, n$
- Can be very slow because of coordinate step and correlated parameters

Drawback of MH-based samplers

All the above traditional MCMC methods are not scalable in Deep Learning, in each iteration: because:

- In each iteration the whole data need to be used to generate a proposal,
- In each iteration the whole data need to be used to calculate the acceptance probability

⇒ We need to come up with some mitigation of the this linear dependency on the number of training samples (analogy: GD vs SGD)

MCMC by Langevin⁹ dynamics

- Overdamped Langevin Ito diffusion Stochastic Differential Equation:
 $dX = \nabla \log \pi(X) + \sqrt{2}dW$, where W is the standard Brownian motion
- In the limit as $t \rightarrow \infty$ we'd like the probability distribution of $X(t)$ to approach a stationary distribution (and we'd like to have it invariant under the diffusion, i.e., equal to π)
- Using Euler–Maruyama discretization method with a fixed time step $\eta > 0$, we obtain $x^{(i+1)} = x^{(i)} + \eta \nabla \log \pi(x^{(i)}) + \sqrt{2\eta}\epsilon^{(i)}$, where $\epsilon^{(i)} \sim N(0, I)$
- The multiplying factor $\sqrt{2\eta}$ is needed in order to guarantee the process to be a correct sampler (by the Fokker-Planck⁸ Equation)
- Detailed balance conditions usually do not hold \Rightarrow can be biased

⁸Wiki

⁹Neal, Radford M. “MCMC using Hamiltonian dynamics.” 2011


Metropolis-adjusted Langevin algorithm (MALA)¹⁰

- Let's combine MH and Langevin dynamics
- $x^{(i+1)} = x^{(i)} + \eta \nabla \log \pi(x^{(i)}) + \sqrt{2\eta} \epsilon^{(i)}$, where $\epsilon^{(i)} \sim N(0, I)$
- We are using the following generating distribution
 $g(y|x^{(i)}) = N(y; x^{(i)} + \eta \nabla \log \pi(x^{(i)}), 2\eta * I)$
- Acceptance distribution is $A(y|x^{(i)}) = \min(1, \frac{\pi(y)}{\pi(x^{(i)})} \frac{g(x^{(i)}|y)}{g(y|x^{(i)})})$
- If uniform random $u \sim U[0, 1]$ is less than $A(y|x^{(i)})$ then we accept the proposal $x^{(i+1)} = y$, otherwise keep the previous value $x^{(i+1)} = x^{(i)}$
- Compared to simple MH, MALA has the advantage that it usually proposes moves into regions of higher probability \Rightarrow more likely to be accepted \Rightarrow much faster convergence
- Compared to simple Langevin, MALA has the advantage of satisfying the detailed balance equations \Rightarrow not biased

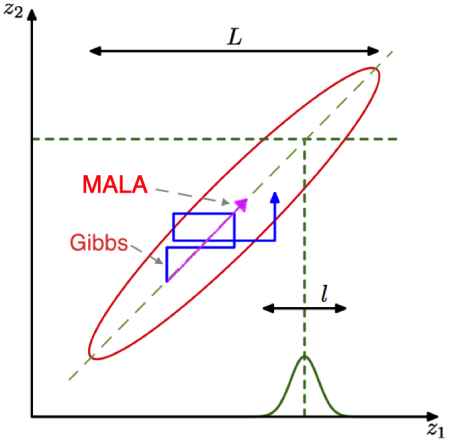
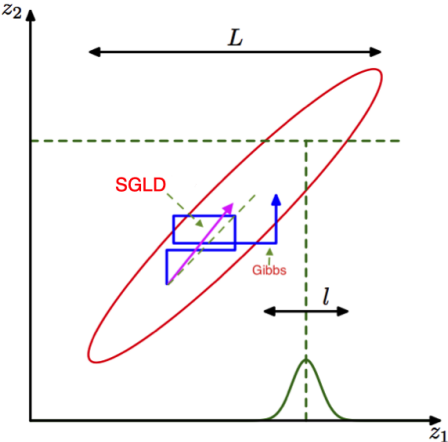
¹⁰Besag, Julian. "Comments on "Representations of knowledge in complex systems" by U. Grenander and AP MI Miller." 1994

Stochastic Gradient Langevin Dynamics (SGLD)¹¹

- Previous approaches require the whole dataset $X = \{x_1, \dots, x_N\}$ to estimate distribution / gradient
- Here we concentrate on the posterior distribution $p(\theta|X)$
- Langevin dynamics: $\theta^{(i+1)} = \theta^{(i)} + \eta \nabla \log p(\theta^{(i)}|X) + \sqrt{2\eta}\epsilon^{(i)}$, where $\epsilon^{(i)} \sim N(0, I)$
- Having $p(\theta|X) \propto p(\theta, X) = p(\theta) \prod_{k=1}^N p(x_k|\theta)$, we can rewrite the dynamics:
$$\theta^{(i+1)} = \theta^{(i)} + \eta(\nabla \log p(\theta^{(i)}) + \sum_{k=1}^N \nabla \log p(x_k|\theta^{(i)})) + \sqrt{2\eta}\epsilon^{(i)}$$
- Let us use only a mini-batch of size n ever step: $B = \{x_{k_1}, \dots, x_{k_n}\}$
- Then the update is $\theta^{(i+1)} = \theta^{(i)} + \eta(\nabla \log p(\theta^{(i)}) + \frac{N}{n} \sum_{j=1}^n \nabla \log p(x_{k_j}|\theta^{(i)})) + \sqrt{2\eta}\epsilon^{(i)}$
- As for Langevin dynamics, here we also could use the Metropolis adjustment (MALA)

¹¹Welling, Max, and Yee W. Teh. “Bayesian learning via stochastic gradient Langevin dynamics.” 2011 

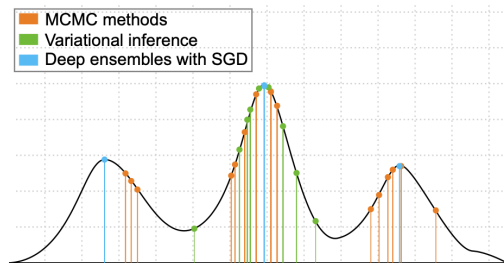
SGLD and MALA



MCMC, VI by backprop and Deep Ensembles comparison

Comparison¹² of different techniques for sampling the posterior:

- MCMC algorithms sample the true posterior but successive samples might be correlated,
- Variational Inference uses a parametric distribution that can suffer from mode collapse,
- Deep ensembles focus on the modes of the distribution.



¹²Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., and Bennamoun, M. Hands-on Bayesian neural networks — A tutorial for deep learning users. 2020.

Takeaway notes

- Markov Chains: a very powerful tool
- Markov Chain Monte Carlo: how to design an iterative process of sampling having in the limit as the stationary distribution the needed one
- MCMC is low-biased (due to the theoretical guarantees), but high-variance solution because of sampling nature
- MCMC is really slow
- How to design transition steps: that is the question (Langevin?)

Thank you!