Sampling methods

2023-01-19

Table of content

- Simple Monte Carlo
- Importance sampling
- Rejection sampling (Optional)
- Variance reduction (Optional)
- Markov chain
- Metropolis-Hastings
- Gibbs sampling
- Maximum-a-posteriori (MAP) estimation
- Laplace's approximation
- Variational inference

Main objective of sampling

We will be using Monte Carlo methods to solve one or both of the following problems:

- **Problem 1**: To generate samples $\{x^{(i)}\}_{i=1}^{N}$ from a given probability distribution $p(x)$
- **Problem 2**: To estimate expectations of functions under the distribution $p(x)$

$$\mathop{\mathbb{E}}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx$$

Simple Monte Carlo

We start with the estimation problem using simple Monte Carlo:

- **Simple Monte Carlo**: Given $\{x^{(i)}\}_{i=1}^{N} \sim p(x)$ we can estimate the estimation $\mathbb{E}_{x\sim p(x)}[f(x)]$ as follow:

$$\mathop{\mathbb{E}}_{x\sim p(x)}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

- Note that the estimator above is consistency according to the Law of Large Numbers (LLN).

Importance sampling

Importance sampling is a method for estimating the expectation of a function $f(x)$

- The density from which we wish to draw samples, $p(x)$, can be evaluated up to normalizing constant, $\tilde{p}(x)$

$$p(x) = \frac{\tilde{p}(x)}{Z_p}$$

- There is a simpler density, $q(x)$, from which it is easy to sample from and easy to evaluate up to a normalizing constant (i.e $\tilde{q}(x)$)

$$q(x) = \frac{\tilde{q}(x)}{Z_q}$$

Importance sampling

- Note that in importance sampling, we wish to sample from $p(x)$, but end up sample from simpler density $q(x)$. Hence, we need to correct this difference by introducing the weights:

$$\tilde{w}_i = \frac{\tilde{p}(x)}{\tilde{q}(x)}$$

- By simple Monte Carlo, we have:

$$\frac{1}{N}\sum_{i=1}^{N}\tilde{w}_i \approx \mathop{\mathbb{E}}_{x\sim q(x)}\left[\frac{\tilde{p}(x)}{\tilde{q}(x)}\right] = \int \frac{\tilde{q}(x)}{\tilde{p}(x)} q(x)dx = \frac{Z_p}{Z_q}$$

Importance sampling

- Apply simple Monte Carlo for our estimator:

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})\frac{p(x^{(i)})}{q(x^{(i)})} = (*)$$

- We want to further simplify the expression above so that our estimator will only rely on $\tilde{p}$ and $\tilde{q}$

$$(*) = \frac{Z_q}{Z_p}\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})\frac{\tilde{p}(x^{(i)})}{\tilde{q}(x^{(i)})} = \frac{Z_q}{Z_p}\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \cdot \tilde{w}_i$$

$$\approx \frac{\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \cdot \tilde{w}_i}{\frac{1}{N}\sum_{i=1}^{N} \tilde{w}_i} = \sum_{i=1}^{N} f(x^{(i)}) \cdot w_i$$

where $w_i = \frac{\tilde{w}_i}{\sum_{i=1}^{N} \tilde{w}_i}$ is our importance weighted estimator

Rejection sampling (Optional)

- Goal: We want to calculate expectations under $p(x) = \tilde{p}(x)/Z_p$ which is a very complicated one-dimensional density
- Assume that we have a simpler proportional density $q(x)$ which we can evaluate (within a multiplicative factor of Z_q)
- Furthermore, assume we know a constant c such that:

$$c\tilde{q}(x) > \tilde{p}(x)$$

Rejection sampling (Optional)

The procedure is as follows:

1. Generate two random numbers
    1. The first, x, is generated from the proposal density $q(x)$
    2. The second, u is generated uniformly from the interval $[0, c\tilde{q}(x)]$
2. Accept or reject the sample x by comparing the value of u with the value of $\tilde{p}(x)$
    1. If $u > \tilde{p}(x)$, then x is rejected
    2. Otherwise x is accepted; x is added to our set of samples $\{x^{(i)}\}$ and the value of u is discarded

Variance reduction (Optional)

- We want to find a Monte Carlo estimator with smaller variance than the standard estimator $\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$
- Methods:
  1. Antithetic variables
  2. Control variates

Antithetic variables (Optional)

In cases where $f(x^{(i)})$ has the same distribution as $f(-x^{(i)})$, we can use:

$$\hat{I}_{\text{anti}} = \frac{1}{2N} \sum_{i=1}^{N} [f(x^{(i)}) + f(-x^{(i)})]$$

Lemma

$$\text{Var}(\hat{I}_{\text{anti}}) \leq \text{Var}(\hat{I})$$

Notes: we have less variance, but twice the computation

Control variates (Optional)

Let Z be a random variable such that $\mathbb{E}(Z) = 0$. For any β, we have:

$$\hat{I}_{\text{CV}} = \frac{1}{N} \sum_{i=1}^{N} [f(x^{(i)}) - \beta Z_n]$$

is an unbiased estimator of $\mathbb{E}[f(x)]$.

The smallest variance is obtain by taking:

$$\beta_{\text{opt}} = \frac{\text{Cov}(f(x), Z)}{\text{Var}(Z)}$$

Markov chain

So far, we have discussed methods to generate i.i.d samples. Now we will consider the case where we generate dependent samples.

- Suppose we have a sequence of data $x_{1:T} = \{x_1, ..., x_T\}$. We say that our data follow a **first-order Markov chain** if:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$$

- Using this assumption, we can factor the joint distribution as:

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t-1})$$

- The second order Markov chain is:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1}, x_{t-2})$$

- The m-th order Markov chain is:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1:t-m})$$

Markov chain

- A useful distinction to make at this point is between stationary and non-stationary distributions that generate our data
  - **Stationary Markov chain**: the distribution generating the data does not change through time

$$p(x_{t+1} = y | x_t = x) = p(x_{t+2} = y | x_{t+1} = x)$$

  - **Non-stationary Markov chain**: the distribution generating the data is a function of time: The transition probabilities $p(x_{t+1} = y | x_t = x)$ depend on the time t

Transition matrix

- When x_t is discrete (e.g. $x_t \in \{1, ..., K\}$ which is called state space), the conditional distribution $p(x_t|x_{t-1})$ can be written as a $K \times K$ matrix.
- We call this the transition matrix A: $A_{ij} = p(x_t = j|x_{t-1} = i)$, the probability of going from state i to state j.
- Notice

$$\begin{aligned} p(x_t = j) &= \sum_i p(x_t = j|x_{t-1} = i)p(x_{t-1} = i) \\ &= \sum_i A_{ij} p(x_{t-1} = i) \end{aligned}$$

- Each row of the matrix sums to one, $\sum_i A_{ij} = 1$, this is called a stochastic matrix

Chapman-Kolmogorov equation

- The n-step transition $A(n)$ is defined as:

$$A_{ij}(n) = p(x_{t+n} = j | x_t = i)$$

- Note that $A(1) = A$
- **Chapman-Kolmogorov equation** states that

$$A(m+n) = A(m)A(n)$$

or equivalently

$$A_{ij}(m+n) = \sum_{k=1}^{K} A_{ik}(m)A_{kj}(n)$$

- We can easily observe that $A(n) = A^n$

Stationary distribution

- We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain
- Let A be the transition matrix, e.g $p(x_{t+1} = j | x_t = j) = A_{ij}$ and $\pi_t(j) = p(x_t = j)$. The initial distribution is given by π_0 and:

$$\pi_1(j) = \sum_i \pi_0(i) A_{ij}$$

- Assume that π_t is a row vector with entries $\pi_t(j)$. This vector is the distribution of x_t, e.g. $p(x_t = j) = \pi_t(j)$.

$$\pi_1 = \pi_0 A \text{ or generally } \pi_t = \pi_0 A^t$$

- Do this infinitely many steps, the distribution of x_t may converge

$$\pi = \pi A$$

then we have reached the stationary distribution (aka the invariant distribution) of the Markov chain

Stationary distribution

- We can find the stationary distribution of a Markov chain by solving the eigenvector equation

$$A^\top v = v \text{ and set } \pi = v^\top$$

v is the eigenvector of $A^\top$ with eigenvalue 1

Detailed balance equation

- A MC is called **irreducible** if we can get from any state to any other state.
- A MC is called **regular** if the transition matrix satisfies $A_{ij}^n > 0$ for some n and all i, j
- A MC is **time reversible** if there exists a distribution π such that

$$\pi(i)A_{ij} = \pi(j)A_{ji}$$

This is called the detailed balance equation.

Theorem

If a Markov chain with transition matrix A is regular and satisfies detailed balance w.r.t distribution π, then π is a stationary distribution

Metropolis-Hastings Algorithm

Importance and rejection sampling work only if the proposal density $q(x)$ is similar to $p(x)$. In high dimensions, it is hard to find one such q.

- The Metropolis–Hastings algorithm instead makes use of a proposal density q which depends on the current state $x^{(t)}$
- The density $q(x|x^{(t)})$ might be a simple distribution such as a Gaussian centered on the current $x^{(t)}$, but can be any density from which we can draw samples (usually symmetric distribution)
- In constrast to importance and rejection sampling, it is not necessary for $q(x'|x^{(t)})$ to look similar to $p(x)$

Note: We usually choose Gaussian as the proposal distribution in sampling problems. This is because given the variance or both the mean and variance, the Gaussian distribution has the **maximum entropy**.

Metropolis-Hastings Algorithm

As before, assume that we can calculate $\tilde{p}(x)$ for any x. The procedure is as follow:

- A tentative new state x' is generated from the proposal density $q(x'|x^{(t)})$. To decide whether to accept the new state, we compute

$$a = \frac{\tilde{p}(x')q(x^{(t)}|x')}{\tilde{p}(x^{(t)})q(x'|x^{(t)})}$$

  - If $a \geq 1$ then the new state is accepted
  - Otherwise, the new state is accepted with probability a
  - If accepted, set $x^{(t+1)} = x'$. Otherwise, set $x^{(t+1)} = x^{(t)}$
- This is a Markov chain with stationary distribution $\pi(x)$ is chosen to be the target distribution $p(x)$

Gibbs sampling

- Suppose the parameter vector x has been divided into n components:
$$x = (x_1, ..., x_n)^\top$$
- At each iteration, the **Gibbs sampler** cycles through the component of x, drawing each subset conditioning on the value of all other components
- This means we perform n steps at each sampling iteration t to obtain $x^{(t+1)}$
- No reject, only accept

Gibbs sampling

At iteration t:

- Choose an ordering of n sub-vectors of x
- For $j = 1$ to n:
  - Sample x_j^t from the distribution given all the other components:

    $$x_j^t \sim p(x_j | x_{-j}^{t-1})$$

    where x_{-j}^{t-1} represents all of the components of x except for x_j at their current values:

    $$x_{-j}^{t-1} = (x_1^t, x_2^t, ..., x_{j-1}^t, x_{j+1}^{t-1}, ..., x_n^{t-1})$$

Posterior inference for latent variable models

A latent variable model has a factorization $p(x, z) = p(z)p(x|z)$ where

- x are the observations or data
- z are the unobserved (latent) variables
- $p(z)$ is usually called the **prior**
- $p(x|z)$ is usually called the **likelihood**
- The conditional distribution of the unobserved variables given the observed variables (aka the **posterior**) is

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x, z)dz}$$

Posterior inference for latent variable models

- The integral $p(x) = \int p(x, z)dz$ is intractable whenever z is high dimensional. This makes evaluating or sampling from the normalized posterior $p(z|x)$ for a given x and z also intractable.
- Here is a list of operations that are expensive:
  - Computing a posterior probability: $p(z|x) = \frac{p(z)p(x|z)}{p(x)}$
  - Computing the evidence/marginal likelihood $p(x) = \int p(z, x)dz$
  - Sampling $z \sim p(z|x)$

Maximum-a-posteriori (MAP) estimation

- Goal: We want to find the most likely unknown quantity under the posterior (i.e the mode)
- We convert the Bayesian estimation problem into a maximization problem:

$$\begin{aligned}
\hat{z}_{\text{MAP}} &= \arg\max_z p(z|x) \\
&= \arg\max_z p(z)p(x|z) \\
&= \arg\max_z \log\, p(z) + \log\, p(x|z)
\end{aligned}$$

Laplace's approximation

- The **Laplace's approximation** method is one of the first approximation inference methods that has been proposed, even before MCMC (and variational inference)
- We want to approximate the log-posterior $\log p(\boldsymbol{z}|\boldsymbol{x})$ using a Taylor's expansion around the mode $\hat{\boldsymbol{z}}_{\text{MAP}}$

$$\log p(\boldsymbol{z}|\boldsymbol{x}) \approx \log p(\hat{\boldsymbol{z}}_{\text{MAP}}|\boldsymbol{x}) - \frac{1}{2}(\boldsymbol{z} - \hat{\boldsymbol{z}}_{\text{MAP}})^\top \hat{\boldsymbol{M}}(\boldsymbol{z} - \hat{\boldsymbol{z}}_{\text{MAP}}) + \text{const}$$

where $\hat{\boldsymbol{M}}$ is the negative Hessian of $\log p(\boldsymbol{z}|\boldsymbol{x})$ evaluated at $\hat{\boldsymbol{z}}_{\text{MAP}}$

$$\hat{\boldsymbol{M}} = -\frac{\partial^2}{\partial \boldsymbol{z} \partial \boldsymbol{z}^\top} \log p(\boldsymbol{z}|\boldsymbol{x}) \Big|_{\boldsymbol{z}=\hat{\boldsymbol{z}}_{\text{MAP}}}$$

- Apply the quadratic expansion similar to above, the multivariate Gaussian approximate posterior is

$$p(\boldsymbol{z}|\boldsymbol{x}) \approx \mathcal{N}\left(\boldsymbol{z}|\hat{\boldsymbol{z}}_{\text{MAP}}, \hat{\boldsymbol{M}}^{-1}\right)$$

Variational inference

Variational inference is an approximate inference method where we seek a tractable (e.g., factorized) approximation to the target intractable distribution

Variational inference works as follows:

- Choose a tractable distribution $q(z) \in \mathcal{Q}$ from a feasible set $\mathcal{Q}$. This distribution will be used to approximate $p(z|x)$.
    - For example, $q(z) = \mathcal{N}(z|\mu, \Sigma)$. The idea is that we'll try choose a $\mathcal{Q}$ that makes $q(z)$ a good approximation of the true posterior $p(z|x)$.
    - Encode some notion of "difference" between $p(z|x)$ and $q(z)$ that can be efficiently estimated. Usually we will use the KL divergence
    - Minimize this difference. Usually we will use an iterative optimization method

Kullback-Leibler (KL) divergence

We will measure the difference between two distribution p and q using the **Kullback-Leibler** divergence

$$\mathrm{KL}(q(z)||p(z|x)) = \int q(z) \log \frac{q(z)}{p(z|x)} dz$$

$$= \mathop{\mathbb{E}}_{z \sim q} \left[\log \frac{q(z)}{p(z|x)} \right]$$

Properties of KL divergence:

- $\mathrm{KL}(q||p) \geq 0$
- $\mathrm{KL}(q||p) = 0 \Leftrightarrow q = p$
- $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$
- KL divergence is not a metric (or distance), since it is not symmetric

Information (I-)Projection

I-projection: $q* = \text{argmin}_{q \in \mathcal{Q}} \text{KL}(q||p) = \mathbb{E}_{x \sim q(x)} \log \frac{q(x)}{p(x)}$

- $p \approx q \Rightarrow \text{KL}(q||p)$ small
- I-projection underestimates support, and does not yield the correct moments

Moment (M-)Projection

M-projection: $q* = \text{argmin}_{q \in \mathcal{Q}} \text{KL}(p||q) = \mathbb{E}_{x \sim p(x)} \log \frac{p(x)}{q(x)}$

- $p \approx q \Rightarrow \text{KL}(q||p)$ small
- M-projection yields a distribution $q(x)$ with the correct mean and covariance
- However, M-projection require expectation w.r.t p, hence intractable
- Most variational inference algorithms make use of the I-projection

ELBO: Evidence lower bound

- Evaluating $\mathrm{KL}(q(z)||p(z|x))$ is intractable because of the integral over z and the term $p(z|x)$, which is intractable to normalize.
- We can still "optimize" this KL without knowing the normalization constant $p(x)$. We solve a surrogate optimization problem: maximize the **evidence lower bound (ELBO)**.

ELBO: Evidence lower bound

Now assume that our approximate inference distribution $q(x)$ is parameterized by ϕ. Maximizing the ELBO is equivalent to minimizing $\mathrm{KL}(q_\phi(z)||p(z|x))$

$$\begin{aligned}
\mathrm{KL}(q_\phi(z)||p(z|x)) &= \mathbb{E}\log\frac{q_\phi(z)}{p(z|x)} \\
&= \mathbb{E}_{z\sim q_\phi}\left[\log\left(q_\phi(z)\cdot\frac{p(x)}{p(z,x)}\right)\right] \\
&= \mathbb{E}_{z\sim q_\phi}\left[\log\frac{q_\phi(z)}{p(z,x)}\right] + \mathbb{E}_{z\sim q_\phi}\log\,p(x) \\
&:= \mathcal{L}(\phi) + \log\,p(x)
\end{aligned}$$

where $\mathcal{L}(\phi)$ is the **ELBO**:

$$\mathcal{L}(\phi) = \mathbb{E}_{z\sim q_\phi}\left[\log\,p(z,x) - \log\,q_\phi(z)\right]$$

Since $p(x)$ is const, maximizing ELBO $\Leftrightarrow$ minimizing $\mathrm{KL}(q_\phi(z)||p(z|x))$

Stochastic variational inference

- Recall the ELBO loss from the previous slide:

$$\mathcal{L}(\phi) = \mathbb{E}_{z \sim q_\phi} \left[\log p(z, x) - \log q_\phi(z) \right]$$

- We want to optimize this function with gradient methods, particularly stochastic gradient descent. Hence, we will need to we will need to compute an unbiased estimate of $\nabla_\phi \mathcal{L}(\phi)$

The reparameterization trick

- We need to sample $z \sim q_\phi(z)$ to estimate the gradient of ELBO with simple Monte Carlo. But the expectation

$$\mathcal{L}(\phi) = \mathbb{E}_{z \sim q_\phi} \left[\log\, p(z, x) - \log\, q_\phi(z)\right]$$

  is over $q_\phi(z)$ which depends on ϕ
- We break this sampling process into two parts:
  - Sample a random variable ϵ that has fixed (or no) parameters, such as a uniform distribution or standard normal.
  - Determinsitically compute z's as a function of ϕ and ϵ, such that:
    - $\epsilon \sim p(\epsilon)$
    - $z = T(\epsilon, \phi) \Rightarrow z \sim q_\phi(z)$

The reparameterization trick

- This makes the density independent of the parameter ϕ, which will let us use simple Monte Carlo: $z = T(\phi, \epsilon)$

$$\begin{aligned}
\nabla_\phi \mathcal{L}(\phi) &= \nabla_\phi \, \mathbb{E}_{z \sim q_\phi(z)} \left[\log p(x, z) - \log q_\phi(z)\right] \\
&= \nabla_\phi \, \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon))\right] \\
&= \mathbb{E}_{\epsilon \sim p(\epsilon)} \, \nabla_\phi \left[\log p(x, T(\phi, \epsilon)) - \log q_\phi(T(\phi, \epsilon))\right]
\end{aligned}$$

- For example, $z = \mu + \sigma\epsilon = T(\phi, \epsilon)$ (here $\phi = (\mu, \sigma)$).
  - $\epsilon \sim \mathcal{N}(0, 1)$
  - $z = \mu + \sigma\epsilon \Rightarrow z \sim \mathcal{N}(\mu, \theta)$

Stochastic variational inference

- Instead of computing the full gradient (which is in general not possible), we compute a simple Monte Carlo estimate of it
- For example, instead of

$$\mathbb{E}_{\epsilon\sim p(\epsilon)}\,\nabla_\phi\left[\log\,p(x,T(\phi,\epsilon))-\log\,q_\phi(T(\phi,\epsilon))\right]$$

- We work with a mini-batch of size m

$$\hat{\mathbb{E}}_{\epsilon\sim p(\epsilon)}\,\nabla_\phi\left[\log\,p(x,T(\phi,\epsilon))-\log\,q_\phi(T(\phi,\epsilon))\right]$$

$$\approx\frac{1}{m}\sum_{i=1}^{m}\nabla_\phi\left[\log\,p(x,T(\phi,\epsilon_i))-\log\,q_\phi(T(\phi,\epsilon_i))\right]$$

MCMC: Pros & Cons

- Pros of MCMC:
  - Asymptotically exact
  - Lots of theoretical guarantees
  - Somewhat well-understood
- Cons of MCMC:
  - Hard to assess convergence
  - Hard to tune hyperparameters
  - Hard to tell if you are making progress
  - Can't use minibatches (easily)

SVI: Pros & Cons

- Pros of SVI:
    - Simple
    - Can tell if making progress
    - Can naturally use minibatches for fitting to large datasets
- Cons of SVI:
    - Limited flexibility of variational approximation
    - Very little guarantees
    - Can get stuck at a bad approximate distribution

Remarks

Other relevant topics that we have not covered:

- Inverse CDF sampling, slice sampling, Box-Muller
- Approximate Bayesian computation & Bayesian synthetic likelihood
- Mean field approximation & CAVI
- Hamiltonian Monte Carlo, Langevin Monte Carlo, Adaptive MCMC
- Parallel & Simulated tempering, Coupling, Nested sampling
- Normalizing flows

References & Further resources

Main references

- **CSC412 lecture slides**: https://erdogdu.github.io/csc412/
- Probabilistic Machine Learning: Advanced Topics (2023) by Kevin Murphy

References for sampling methods in **time series**

- Nonlinear Time Series Analysis (2018) by Chen & Tsay
- Time Series Analysis by State Space Method (2012) by Durbin & Koopman

References for **MCMC**

- Monte Carlo Statistical Methods (2004) by Casella & Robert

References for **variational inference**

- Blei et al. (2018), Variational Inference: A Review for Statistician