Monte-Carlo Estimation

Cheng Soon Ong
Marc Peter Deisenroth

December 2020
Setting: Computing expectations

\[ \int f(x)p(x)dx = \mathbb{E}_{x \sim p}[f(x)] \]
Setting: Computing expectations

\[ \int f(x)p(x)dx = E_{x \sim p}[f(x)] \]

**Moments** of random variables

\[ M_k(x) = \int x^k p(x)dx \]
Setting: Computing expectations

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

**Moments** of random variables

\[
M_k(x) = \int x^k p(x) dx = \mathbb{E}_{x \sim p(x)}[x^k]
\]
Setting: Computing expectations

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

Moments of random variables

\[ M_k(x) = \int x^k p(x)dx = \mathbb{E}_{x \sim p}[x^k] \]

Marginal likelihood

\[ p(X) = \int p(X|\theta)p(\theta)d\theta \]
Setting: Computing expectations

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

Moments of random variables

\[ M_k(x) = \int x^k p(x)dx = \mathbb{E}_{x \sim p}[x^k] \]

Marginal likelihood

"Average likelihood"

\[ p(X) = \int p(X|\theta)p(\theta)d\theta = \mathbb{E}_{\theta \sim p(\theta)}[p(X|\theta)] \]
Setting: Computing expectations

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

**Moments** of random variables

\[M_k(x) = \int x^k p(x)\,dx = \mathbb{E}_{x \sim p}[x^k]\]

**Marginal likelihood**

“Average likelihood”

\[p(X) = \int p(X|\theta)p(\theta)\,d\theta = \mathbb{E}_{\theta \sim p(\theta)}[p(X|\theta)]\]

**Predictions in a Bayesian model**

\[p(x_*|X) = \int p(x_*|\theta)p(\theta|X)\,d\theta\]
Setting: Computing expectations

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

Moments of random variables
\[ M_k(x) = \int x^k p(x)dx = \mathbb{E}_{x \sim p}[x^k] \]

Marginal likelihood
“Average likelihood”
\[ p(X) = \int p(X|\theta)p(\theta)d\theta = \mathbb{E}_{\theta \sim p(\theta)}[p(X|\theta)] \]

Predictions in a Bayesian model
“Average predictive distribution”
\[ p(x_*|X) = \int p(x_*|\theta)p(\theta|X)d\theta \]
\[ = \mathbb{E}_{\theta \sim p(\theta|X)}[p(x_*|\theta)] \]
Key idea

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

Key idea

Make use of random numbers to approximate the expectation.
How it works

Key idea
Make use of random numbers to approximate an expectation.

Compute expectations via statistical sampling:

\[
\mathbb{E}[f(x)] = \int f(x)p(x)\,dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x)
\]
How it works

Key idea

Make use of random numbers to approximate an expectation.

- Compute expectations via statistical sampling:

\[
E[f(x)] = \int f(x)p(x)dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x)
\]

- Example: Making predictions in a supervised setting (e.g., Bayesian logistic regression with training set \( D = \{X, y\} \) at test input \( x_* \))

\[
p(y_*|x_*, D) = \int p(y_*|\theta, x_*) p(\theta|D) d\theta
\]
How it works

Key idea
Make use of random numbers to approximate an expectation.

Compute expectations via statistical sampling:

\[
E[f(x)] = \int f(x)p(x)dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x)
\]

Example: Making predictions in a supervised setting (e.g., Bayesian logistic regression with training set \(D = \{X, y\}\) at test input \(x_*\))

\[
p(y_*|x_*, D) = \int p(y_*|\theta, x_*) p(\theta|D) d\theta \approx \frac{1}{S} \sum_{s=1}^{S} p(y_*|\theta^{(s)}, x_*) , \quad \theta^{(s)} \sim p(\theta|D)
\]
Properties of Monte Carlo estimation

\[ E[f(x)] = \int f(x)p(x)dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x) \]

- Estimator is unbiased and asymptotically consistent, i.e.,

\[ \lim_{S \to \infty} \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}) = E[f(x)] + \epsilon \]

- Error \( \epsilon \) is normal (Gaussian) and its variance shrinks \( \propto 1/S \), independent of the dimensionality
Monte Carlo estimation

\[ \mathbb{E}[f(x)] = \int f(x)p(x) \, dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x) \]

- How do we get these samples?
Monte Carlo estimation

\[ E[f(x)] = \int f(x)p(x)dx \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x) \]

- How do we get these samples?
- Sampling from simple distributions
  - Use libraries if the distribution has a “name”
- Sampling from complicated distributions
  - Rejection sampling (does not scale to high dimensions)
  - Importance sampling (does not scale to high dimensions)
  - Markov chain Monte Carlo (MCMC)

Iain Murray’s NeurIPS-2015 tutorial
Monte Carlo estimation

\[
E[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{S} \sum_{s=1}^{S} f(\mathbf{x}^{(s)}), \quad \mathbf{x}^{(s)} \sim p(\mathbf{x})
\]

- How do we get these samples?
  - Sampling from simple distributions
    - Use libraries if the distribution has a “name”
  - Sampling from complicated distributions
    - Rejection sampling (does not scale to high dimensions)
    - Importance sampling (does not scale to high dimensions)
    - Markov chain Monte Carlo (MCMC) >> Iain Murray’s NeurIPS-2015 tutorial
Example

\[ Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)\,dx = \int_{-3}^{3} 6 \exp \left( -x^2 - \sin^2(3x) \right) \mathcal{U}[-3,3] \,dx \]
Example

\[ Z = E_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6 \exp \left( -x^2 - \sin(3x)^2 \right) U[-3,3] dx \]

▶ Monte-Carlo estimator

\[ E_{x \sim U}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim U[-3,3] \]
Example

\[ Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6 \exp \left(-x^2 - \sin(3x)^2\right) U[-3, 3]dx \]

Monte-Carlo estimator

\[ \mathbb{E}_{x \sim U}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim U[-3, 3] \]
Example

\[
Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6 \exp(-x^2 - \sin(3x)^2) U[-3,3]dx
\]

Monte-Carlo estimator

\[
\mathbb{E}_{x \sim U}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim U[-3,3]
\]
Some application areas

- Empirical risk minimization (Vapnik, 1991)
- Reinforcement learning (e.g., Sutton & Barto, 1998)
- Bayesian optimization (e.g., Snoek et al., 2012; Wilson et al., 2018)
- Variational deep learning (e.g., Rezende et al., 2014; Kingma & Welling, 2014)
- Probabilistic programming
  - Frank Wood’s NeurIPS-2015 tutorial

From Wilson et al. (2018)
Some application areas

- Empirical risk minimization (Vapnik, 1991)
- Reinforcement learning (e.g., Sutton & Barto, 1998)
- Bayesian optimization (e.g., Snoek et al., 2012; Wilson et al., 2018)
- Variational deep learning (e.g., Rezende et al., 2014; Kingma & Welling, 2014)
- Probabilistic programming
  - Frank Wood’s NeurIPS-2015 tutorial
- High-energy physics (e.g., Buckley et al., 2011)
- Robotics (e.g., Dellaert et al., 1999)
Considerations

\[ E[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim p(x) \]

- Require many samples to get a good estimate of the value of the integral
- Design efficient samplers (computationally efficient, low variance)
- Function needs to be cheap to evaluate
- Good for learning, if we are just interested in an unbiased estimator
- Estimator does not take the locations of the samples into account
  - Could be problematic in small-sample regimes (O’Hagan, 1987)
Summary: Monte Carlo estimation

- Random numbers to compute expectations
- Estimator has nice properties (e.g., unbiased, asymptotically consistent)
- Scales to high dimensions
- General approach and straightforward
- Widely applicable
- Generating samples is the key challenge (not covered here)
References


