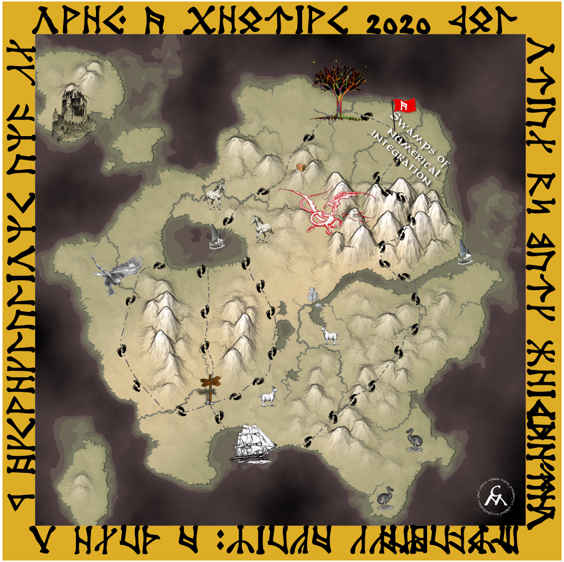

2020
Swamps of Numerical Integration

2020
Swamps of Numerical Integration
Monte Carlo Heights

Monte-Carlo Estimation

Cheng Soon Ong
Marc Peter Deisenroth

December 2020

Setting: Computing expectations

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

Setting: Computing expectations

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

Moments of random variables $$M_k(x) = \int x^k p(x)dx$$

Setting: Computing expectations

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

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

Setting: Computing expectations

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

Moments of random variables

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

Marginal likelihood

$$p(\boldsymbol{X}) = \int p(\boldsymbol{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Setting: Computing expectations

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

Moments of random variables

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

Marginal likelihood
"Average likelihood"

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

Setting: Computing expectations

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

Moments of random variables

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

Marginal likelihood
"Average likelihood"

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

Predictions in a Bayesian model

$$p(\boldsymbol{x}_*|\boldsymbol{X}) = \int p(\boldsymbol{x}_*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X})d\boldsymbol{\theta}$$

Setting: Computing expectations

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

Moments of random variables

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

Marginal likelihood
"Average likelihood"

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

Predictions in a Bayesian model
"Average predictive distribution"

$$p(\boldsymbol{x}_*|\boldsymbol{X}) = \int p(\boldsymbol{x}_*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X})d\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{X})}[p(\boldsymbol{x}_*|\boldsymbol{\theta})]$$

Key idea

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

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:

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

How it works

Key idea

Make use of random numbers to approximate an expectation.

- Compute expectations via statistical sampling:

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

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

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \int p(y_*|\boldsymbol{\theta}, \boldsymbol{x}_*)\, \underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{parameter posterior}}\, d\boldsymbol{\theta}$$

How it works

Key idea

Make use of random numbers to approximate an expectation.

- Compute expectations via statistical sampling:

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

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

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \int p(y_*|\boldsymbol{\theta}, \boldsymbol{x}_*) \underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{parameter posterior}} d\boldsymbol{\theta} \approx \frac{1}{S}\sum_{s=1}^{S} p(y_*|\boldsymbol{\theta}^{(s)}, \boldsymbol{x}_*), \quad \boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta}|\mathcal{D})$$

Properties of Monte Carlo estimation

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

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

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

- Error ϵ is normal (Gaussian) and its variance shrinks $\propto 1/S$, independent of the dimensionality

Monte Carlo estimation

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

- How do we get these samples?

Monte Carlo estimation

$$\mathbb{E}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{x}^{(s)}), \qquad \boldsymbol{x}^{(s)} \sim p(\boldsymbol{x})$$

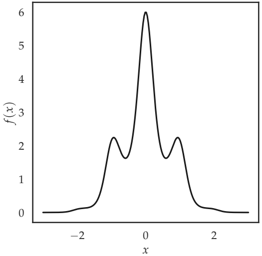
- How do we get these samples?
- Sampling from simple distributions
  - Use libraries if the distribution has a "name"

Monte Carlo estimation

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

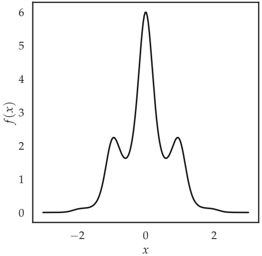
- How do we get these samples?
- Sampling from simple distributions
  - Use libraries if the distribution has a “name”
- Sampling from complicated distributions
  - Rejection sampling (does not scale to high dimensions)
  - Importance sampling (does not scale to high dimensions)
  - Markov chain Monte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

Example



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

Example

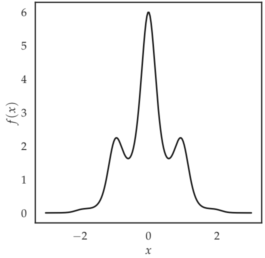


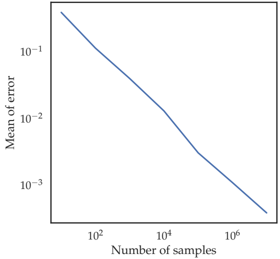
$$Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6\,\mathsf{exp}\left(-x^2 - \mathsf{sin}(3x)^2\right)\mathcal{U}[-3,3]dx$$

- Monte-Carlo estimator

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

Example



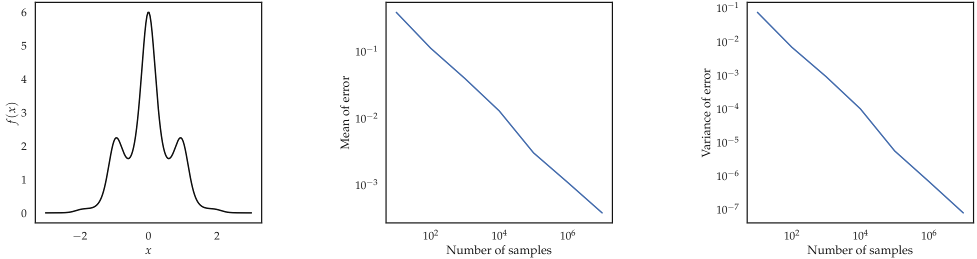


$$Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6\,\textsf{exp}\left(-x^2 - \textsf{sin}(3x)^2\right)\mathcal{U}[-3,3]dx$$

- Monte-Carlo estimator

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

Example



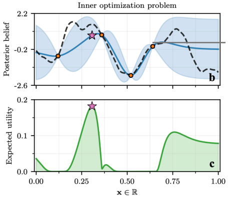
$$Z = \mathbb{E}_x[f(x)] = \int f(x)p(x)dx = \int_{-3}^{3} 6\,\mathbf{exp}\left(-x^2 - \mathbf{sin}(3x)^2\right)\mathcal{U}[-3,3]dx$$

- Monte-Carlo estimator

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

Some application areas

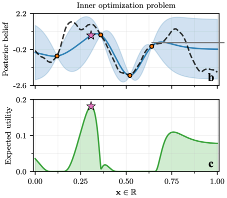
- Empirical risk minimization (Vapnik, 1991)
- Reinforcement learning (e.g., Sutton & Barto, 1998)
- Bayesian optimization
  (e.g., Snoek et al., 2012; Wilson et al., 2018)
- Variational deep learning
  (e.g., Rezende et al., 2014; Kingma & Welling, 2014)
- Probabilistic programming
  - ▶▶▶ Frank Wood's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



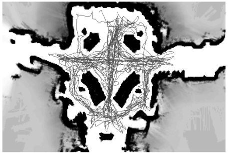
From Wilson et al. (2018)

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)



From Wilson et al. (2018)



From Dellaert et al. (1999)

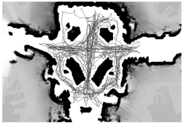
Considerations

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

- 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

Buckley, A., Butterworth, J., Gieseke, S., Grellscheid, D., Höche, S., Hoeth, H., Krauss, F., Lönnblad, L., Nurse, E., Richardson, P., et al. (2011). General-Purpose Event Generators for LHC Physics. *Physics Reports*, 504(5):145–233.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Proceedings of International Conference on Robotics and Automation*.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations*.

Murray, I. (2015). Monte Carlo Inference Methods. *NeurIPS Tutorial*.

O'Hagan, A. (1987). Monte Carlo is Fundamentally Unsound. *The Statistician*, 36(2/3):247–249.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Variational Inference in Deep Latent Gaussian Models. In *Proceedings of the International Conference on Machine Learning*.

References (cont.)

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

Vapnik, V. (1991). Principles of Risk Minimization for Learning Theory . In *Advances in Neural Information Processing Systems*.

Wilson, J. T., Hutter, F., and Deisenroth, M. P. (2018). Maximizing Acquisition Functions for Bayesian Optimization. In *Advances in Neural Information Processing Systems*.

Wood, F. (2015). Probabilistic Programming. *NeurIPS Tutorial*.