Monte Carlo Variance Reduction
One Control Variate at a Time

Jacob Scott

In collaboration with

KC Kong, Konstantin Matchev, Stephen Mrenna, Prasanth Shyamsundar

Pheno, May 2023

Motivation

The Why

- Comparing theory and experiment requires integration.
- Most modern simulation tools use MC/`vegas`.
- Good to have better accuracy with less resources.

Motivation

The Why

- Comparing theory and experiment requires integration.
- Most modern simulation tools use MC/`vegas`.
- Good to have better accuracy with less resources.

The What

A look at **Importance sampling** applied by `vegas` and **Control Variates** applied on top by `control-vegas`.

Monte Carlo Integration

Expectation Value

$$E_p[f] = \int_{\text{range of } p} \mathrm{d}x\, f(x) p(x)$$
$$\approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

where $x_i \sim p(x)$

Integral

$$I = \int_a^b \mathrm{d}x\, f(x)$$
$$= (b-a) \int_a^b \mathrm{d}x\, \frac{f(x)}{b-a}$$
$$= (b-a) E_{U[a,b]}[f]$$
$$\approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$$

where $x_i \sim U[a, b]$.

Monte Carlo Integration

Expectation Value

$$E_p[f] = \int_{\text{range of } p} \mathrm{d}x\, f(x) p(x)$$
$$\approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

where $x_i \sim p(x)$

Integral

$$I = \int_a^b \mathrm{d}x\, f(x)$$
$$= (b-a) \int_a^b \mathrm{d}x\, \frac{f(x)}{b-a}$$
$$= (b-a) E_{U[a,b]}[f]$$
$$\approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$$

where $x_i \sim U[a, b]$.

Monte Carlo Integration

- Very simple
- Scales well with dimensionality
- Will converge
- But it converges slowly
- Even worse if function is highly peaked

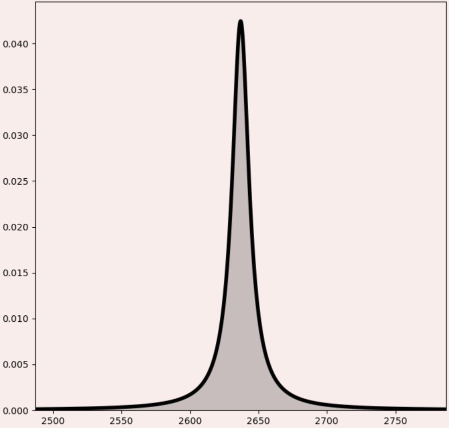


Figure 1: E.g. A Breit-Wigner distribution

Importance Sampling

$$I = \int_a^b \mathrm{d}x\, f(x) = \int_a^b \mathrm{d}x\, \frac{f(x)}{p(x)} p(x) = E_p\left[\frac{f}{p}\right] \approx \frac{1}{N}\sum_{i=0}^{N}\frac{f(x_i)}{p(x_i)}$$

$$\text{where} \quad x_i \sim p(x)$$

$$\text{and} \quad \int_a^b \mathrm{d}x\, p(x) = 1$$

Importance Sampling

$$I = \int_a^b \mathrm{d}x\, f(x) = \int_a^b \mathrm{d}x\, \frac{f(x)}{p(x)} p(x) = E_p\left[\frac{f}{p}\right] \approx \frac{1}{N}\sum_{i=0}^{N} \frac{f(x_i)}{p(x_i)}$$

$$\text{where} \quad x_i \sim p(x)$$

$$\text{and} \quad \int_a^b \mathrm{d}x\, p(x) = 1$$

- If $p(x) \propto f(x)$, then we find exact value of integral.
- So we want a p that mimics f, e.g. p peaks where f does.

Control Variates

$$I = \int_a^b \mathrm{d}x\, f(x) = \int_a^b \mathrm{d}x\, f^*(x) \approx \frac{b-a}{N} \sum_{i=1}^{N} f^*(x_i)$$

$$\text{where} \quad f^*(x) = f(x) + c\Big(g(x) - E_{U[a,b]}[g]\Big)$$

$$\text{and} \quad x_i \sim U[a, b]$$

Control Variates

$$I = \int_a^b \mathrm{d}x\, f(x) = \int_a^b \mathrm{d}x\, f^*(x) \approx \frac{b-a}{N} \sum_{i=1}^{N} f^*(x_i)$$

$$\text{where} \quad f^*(x) = f(x) + c\Big(g(x) - E_{U[a,b]}[g]\Big)$$

$$\text{and} \quad x_i \sim U[a,b]$$

- What is c?
  - Want to minimize variance: $\frac{\partial \mathrm{Var}(f^*)}{\partial c} = 0$
  - Gives us $c^* = -\frac{\mathrm{Cov}(f,g)}{\mathrm{Var}(g)}$
- New variance: $\mathrm{Var}(f^*) = \Big[1 - \rho^2(f,g)\Big]\mathrm{Var}(f)$
  (and $|\rho(f,g)| \leq 1$)

ρ is the Pearson correlation coefficient.

Control Variates

$$I = \int_a^b \mathrm{d}x\, f(x) = \int_a^b \mathrm{d}x\, f^*(x) \approx \frac{b-a}{N} \sum_{i=1}^{N} f^*(x_i)$$

$$\text{where} \quad f^*(x) = f(x) + c\Big(g(x) - E_{U[a,b]}[g]\Big)$$

$$\text{and} \quad x_i \sim U[a,b]$$

- We want $g(x)$ to be correlated to $f(x)$,
- and to have a known expectation value.

Combining CV & IS

$$I = \int_a^b \mathrm{d}x\, \frac{f(x)}{p(x)} p(x) \qquad \text{(Start with IS)}$$

$$= \int_a^b \mathrm{d}x \left[\frac{f(x)}{p(x)} p(x) + c \left(\frac{g(x)}{p(x)} p(x) + E_p \left[\frac{g}{p} \right] \right) \right] \qquad \text{(Add CV)}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} + c^* \left(\frac{g(x_i)}{p(x_i)} + E_p \left[\frac{g}{p} \right] \right) \qquad \text{(where } x_i \sim p(x)\text{)}$$

- So now we need an appropriate $p(x)$ for both $f(x)$ and $g(x)$ and an appropriate $g(x)$ such that we know $E[g/p]$.
- Is this helpful?

vegas

Uses an adaptive form of importance sampling

1. Specify number of iterations and number of evaluations per iteration.
2. Create map between uniformly-spaced y_i's and $x_i's$ via Jacobian.
3. Maps $[0, 1]$ to $[a, b]$ varying widths between points.
4. Estimate integral and update map for the number of iterations.

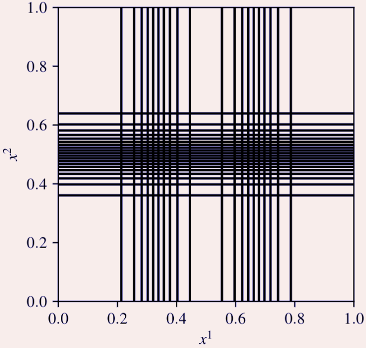


Figure 2: Map for 2 dimensions from a 4D double Gaussian. From 2111.07806 [1]

control-vegas

Remember, we want:

1. A $g(x)$ that is correlated to $f(x)^*$,
2. and whose expectation value is known.

---

*New variance: $\mathrm{Var}(f^*) = \left[1 - \rho^2(f, g)\right] \mathrm{Var}(f)$

control-vegas

Remember, we want:

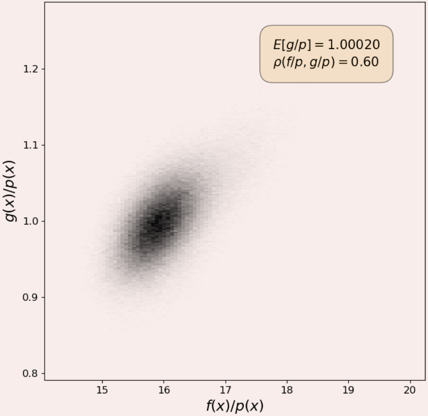
1. A $g(x)$ that is correlated to $f(x)^*$,
2. and whose expectation value is known.

Idea: we use the maps that vegas generates as $g(x)$.
Why?

1. The maps are are correlated to each other,
2. and $E_p[g/p] = \int_a^b \mathrm{d}x\, g(x) = 1$ since $g(x)$ is a PDF.

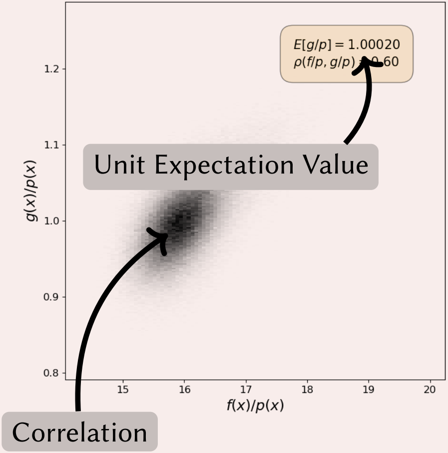
*New variance: $\mathrm{Var}(f^*) = \left[1 - \rho^2(f,g)\right]\mathrm{Var}(f)$

Is g/p valid? $\left[\text{With } f(x) = \sum_i^{96} x_i(1 - x_i)\right]$



20 iterations, 10^4 evaluations per using 5th iteration as CV

Is g/p valid? $\left[\text{With } f(x) = \sum_i^{96} x_i(1 - x_i)\right]$



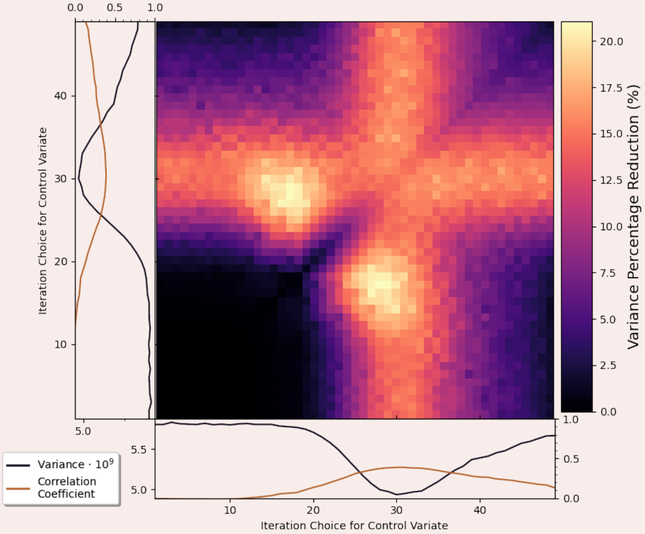
- g/p and f/p are correlated.
  - Quantiatively shown with $\rho = 0.60$.
- Expectation value is 1.
- So this choice for g/p is valid.

20 iterations, 10^4 evaluations per using 5th iteration as CV

Updating Your CV

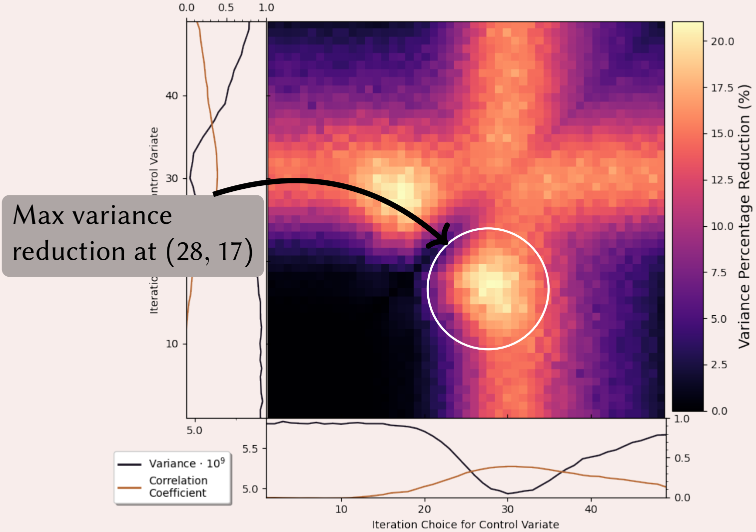
- `vegas` produces a map for each of the N iterations it completes.
- There are then $N - 1$ choices for our CV.
- Which iteration minimizes the variance?
- Can we choose multiple iterations and have mulitple CVs?

Updating Your CV - 16D Gaussian



50 iterations, 2.5×10^4 evaluations per, averaged over 10 runs

Updating Your CV - 16D Gaussian



50 iterations, 2.5×10^4 evaluations per, averaged over 10 runs

How It Runs

```python
from control_vegas import CVIntegrator
from control_vegas.functions import NGauss, NPolynomial

# Create function
ng = NGauss(16)
np = NPolynomial(96)

# Create integrator class
cvig = CVIntegrator(ng, evals=5000, tot_iters=50, cv_iters=[25, 27])
cvip = CVIntegrator(np, evals=5000, tot_iters=50, cv_iters='all')

# Run the integration
cvig.integrate()
cvip.integrate()
```

Does It Work?

Function	Dim	1 CV VPR	2 CV VPR	All CV VPR
Gaussian	2	18.6%	31.0%	46.0%
	4	17.4%	27.0%	39.0%
	8	16.9%	24.2%	34.5%
	16	12.7%	17.1%	24.5%
Polynomial	18	29.2%	44.0%	32.3%
	54	43.8%	48.0%	69.3%
	96	52.1%	61.4%	82.1%

1. One and two control variate cases are using optimal choices,
2. and adds no extra time to run.
3. 'All CVs' takes $\sim$ 20x longer.

50 iterations, 5000 evaluations per, averaged over 10 runs

What We Got & Future Work

What we got:

- A for-free means for MC variance reduction built on vegas

What we want:

1. Smarter ways of choosing CV(s)
2. Faster execution
3. Usage of other variance reduction methods
   - e.g, antithetic variates and ML models like normalizing flow
4. A better name (there's some contenders)

References

[1] G. Peter Lepage. "Adaptive multidimensional integration: vegas enhanced". In: *Journal of Computational Physics* 439 (Aug. 2021), p. 110386. DOI: 10.1016/j.jcp.2021.110386. arXiv: 2111.07806 [comp-ph]. URL: https://doi.org/10.1016%2Fj.jcp.2021.110386.

[2] Christina Gao, Joshua Isaacson, and Claudius Krause. "i-flow: High-dimensional integration and sampling with normalizing flows". In: *Machine Learning: Science and Technology* 1.4 (Nov. 2020), p. 045023. DOI: 10.1088/2632-2153/abab62. URL: https://doi.org/10.1088%2F2632-2153%2Fabab62.

[3] William H. Press et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* 3rd ed. Cambridge University Press, 2007. ISBN: 0521880688.

[4] Jacob Scott. *Control Vegas.* Version 1.1.0. Apr. 2023. URL: https://github.com/crumpstrr33/control-vegas.

n Control Variates

The variance is:

$$\begin{aligned}
\mathrm{Var}\left(f^*\right) &= \mathrm{Var}\left(f(x) + \sum_{i=1}^{n} c_i(g_i(x) - E[g_i])\right) \\
&= \mathrm{Var}\left(f\right) + 2\mathrm{Cov}\left(f, \sum_{i=1}^{n} c_i g_i\right) + \mathrm{Var}\left(\sum_{i=1}^{n} c_i g_i\right) \\
&= \mathrm{Var}\left(f\right) + 2\sum_{i=1}^{n} c_i \mathrm{Cov}\left(f, g_i\right) + \sum_{ij}^{n} c_i c_j \mathrm{Cov}\left(g_i, g_j\right)
\end{aligned}$$

where $\mathrm{Cov}\left(g_i, g_i\right) = \mathrm{Var}\left(g_i\right)$.

n Control Variates

Taking derivatives with respect to the coefficients gives

$$\frac{\partial \mathrm{Var}\left(f^*\right)}{\partial c_j} = 2\mathrm{Cov}\left(f, g_j\right) + 2\sum_{i=1}^{n} c_i \mathrm{Cov}\left(g_i, g_j\right)$$

and setting that equal to zero:

$$\sum_{i=1}^{n} c_i \mathrm{Cov}\left(g_i, g_j\right) = -\mathrm{Cov}\left(f, g_j\right).$$

If we let $A_j = -\mathrm{Cov}\left(f, g_j\right)$ and $B_{ij} = \mathrm{Cov}\left(g_i, g_j\right)$ then

$$\sum_{i=1}^{n} B_{ij} c_i = A_j \qquad \text{or} \qquad \mathbf{c} = B^{-1}\mathbf{A}$$

Scalar Top Loop

With $s_{12} = -s_{23} = 130^2$, $s_1 = s_2 = s_3 = 0$, $s_4 = 125^2$ and $m_t = 173.9$,

$$
\begin{aligned}
f_5 = S_{\mathrm{Box}}(&s_{12}, s_{23}, s_1, s_2, s_3, s_4, m_t^2, m_t^2, m_t^2, m_t^2) \\
&+ S_{\mathrm{box}}(s_{23}, s_{12}, s_2, s_3, s_4, s_1, m_t^2, m_t^2, m_t^2, m_t^2) \\
&+ S_{\mathrm{box}}(s_{12}, s_{23}, s_3, s_4, s_1, s_2, m_t^2, m_t^2, m_t^2, m_t^2) \\
&+ S_{\mathrm{box}}(s_{23}, s_{12}, s_4, s_1, s_2, s_3, m_t^2, m_t^2, m_t^2, m_t^2)
\end{aligned}
$$

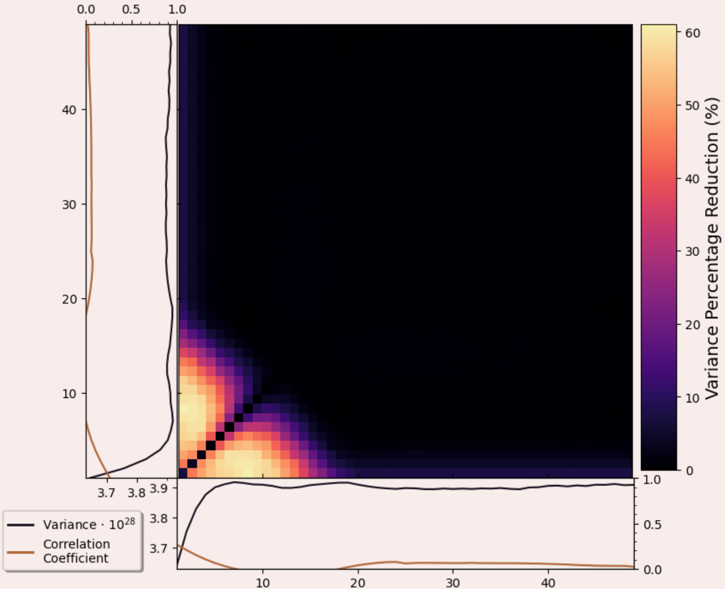
where

$$
S_{\mathrm{box}}(s_{12}, s_{23}, s_1, s_2, s_3, s_4, m_1^2, m_2^2, m_3^2, m_4^2) = \int_0^1 \frac{\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3}{\widetilde{\mathcal{F}}_{\mathrm{box}}^2}
$$

and

$$
\begin{aligned}
\widetilde{\mathcal{F}}_{\mathrm{box}}^2 = &\left(1 + t_1 + t_2 + t_3\right)\left(t_1 m_1^2 + t_2 m_2^2 + t_3 m_3^2 + m_4^2\right) \\
&- \left(s_{12} t_2 + s_{23} t_1 t_3 + s_1 t_1 + s_2 t_1 t_2 + s_3 t_2 t_3\right)
\end{aligned}
$$

Scalar Top Loop



More Results

		Vegas	1 CV			2 CVs			All CVs		
Function	Dim	Time (s)	VPR	Time (s)		VPR	Time (s)		VPR	Time (s)	
Gaussian	2	0.12	18.61%	0.12	(1.0)	30.98%	0.18	(1.4)	45.97%	7.44	(60.1)
	4	0.23	17.40%	0.23	(1.0)	27.04%	0.32	(1.4)	38.97%	8.91	(38.2)
	8	0.33	16.89%	0.33	(1.0)	24.19%	0.45	(1.4)	34.54%	11.85	(36.2)
	16	0.62	12.74%	0.62	(1.0)	17.09%	0.93	(1.5)	24.54%	19.65	(31.7)
Camel	2	0.19	0.17%	0.19	(1.0)	0.77%	0.29	(1.5)	1.06%	8.63	(44.4)
	4	0.25	0.10%	0.25	(1.0)	0.37%	0.35	(1.4)	0.46%	11.39	(46.1)
	8	0.45	0.22%	0.45	(1.0)	0.24%	0.64	(1.4)	0.36%	15.03	(33.5)
	16	0.68	9.77%	0.68	(1.0)	16.12%	1.04	(1.5)	1.00%	21.85	(32.1)
Entangled Circles	2	0.20	0.28%	0.20	(1.0)	0.86%	0.27	(1.3)	1.58%	6.84	(34.5)
Annulus with Cuts	2	0.12	0.01%	0.12	(1.0)	28.64%	0.17	(1.4)	90.87%	6.70	(55.6)
Scalar-top-loop	3	0.33	6.82%	0.33	(1.0)	50.56%	0.43	(1.3)	57.79%	9.20	(27.9)
Polynomial	18	0.66	29.16%	0.66	(1.0)	44.02%	1.04	(1.6)	32.28%	24.70	(37.4)
	54	3.23	43.75%	3.23	(1.0)	48.02%	5.00	(1.5)	69.32%	71.22	(22.0)
	96	5.14	52.07%	5.14	(1.0)	61.44%	6.12	(1.2)	82.09%	138.99	(27.1)

Table 1: Results for 50 iterations and 5000 events per iteration averaged over 10 runs. The lighter (darker) colored cells are for runs that are more than 1.0% (5.0%) off from the true value. The values in the parentheses in the time column are how much longer that instance took to run compared to the corresponding Vegas instance.