

# Andy Jones

Technical blog

 ajones788@gmail.com

## Log-derivative trick

The “log-derivative trick” is really just a simple application of the chain rule. However, it allows us to rewrite expectations in a way that is amenable to Monte Carlo approximation.

## Log-derivative trick

Suppose we have a function  $p(x; \theta)$  (in this context we’ll mostly think of  $p$  as a probability density), and we’d like to take the gradient of its logarithm with respect to  $\theta$ ,

$$\nabla_{\theta} \log p(x; \theta).$$

By a simple application of the chain rule, we have

$$\nabla_{\theta} \log p(x; \theta) = \frac{\nabla_{\theta} p(x; \theta)}{p(x; \theta)}$$

which, rearranging, implies that

$$\nabla_{\theta} p(x; \theta) = p(x; \theta) \nabla_{\theta} \log p(x; \theta).$$

## Score function estimator

In many statistical applications, we want to estimate the gradient of an expectation of a function  $f$ :

$$\nabla_{\theta} \mathbb{E}_{p(x; \theta)} [f(x)].$$

To learn more about a few applications where this gradient estimation problem shows up, as well as more modern methods for solving it, I’d recommend [this review](#) by Shakir Mohamed et al.

Unfortunately, we cannot directly approximate this expression with naive Monte Carlo methods. This is because the expression isn’t in general an expectation.

Expanding the expectation we have:

$$\begin{aligned}\nabla_{\theta} \mathbb{E}_{p(x;\theta)}[f(x)] &= \nabla_{\theta} \int p(x; \theta) f(x) dx \\ &= \int \underbrace{\nabla_{\theta} p(x; \theta)}_{\text{density?}} f(x) dx \quad (\text{Leibniz rule})\end{aligned}$$

However,  $\nabla_{\theta} p(x; \theta)$  will not in general be a valid probability density, so we cannot approximate this with

$$\nabla_{\theta} \mathbb{E}_{p(x;\theta)}[f(x)] \approx \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} p(x_i; \theta) f(x_i).$$

Thankfully, the log-derivative trick allows us to rewrite it as a true expectation:

$$\begin{aligned}\nabla_{\theta} \mathbb{E}_{p(x;\theta)}[f(x)] &= \nabla_{\theta} \int p(x; \theta) f(x) dx \\ &= \int \nabla_{\theta} p(x; \theta) f(x) dx \quad (\text{Leibniz rule}) \\ &= \int p(x; \theta) \frac{\nabla_{\theta} p(x; \theta)}{p(x; \theta)} f(x) dx \quad \left( \text{Multiply by } 1 = \frac{p(x; \theta)}{p(x; \theta)} \right) \\ &= \int p(x; \theta) \nabla_{\theta} \log p(x; \theta) f(x) dx \quad (\text{Log-derivative trick}) \\ &= \mathbb{E}_{p(x;\theta)}[\nabla_{\theta} \log p(x; \theta) f(x)].\end{aligned}$$

We can then approximate this expectation with  $n$  Monte Carlo samples from  $p(x; \theta)$ ,  $x_1, \dots, x_n$ :

$$\mathbb{E}_{p(x;\theta)}[\nabla_{\theta} \log p(x; \theta) f(x)] \approx \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} \log p(x_i; \theta) f(x_i).$$

## Applications

Although relatively straightforward, the score function estimator shows up all over the place. In reinforcement learning, it's known as the REINFORCE method, in which the gradient of the policy is being taken. In variational inference, it shows up when trying to optimize the evidence lower bound (ELBO). And in computational finance, this estimator is important for performing “sensitivity analysis”, or understanding how financial outcomes change with underlying model assumptions.

Another interesting line of work has been exploring ways to reduce the variance of the score function estimator, which can have extremely high variance, especially in discrete settings. Much work has been done to design effective control variates. Also, in discrete latent variable models, another popular approach is to introduce a continuous relaxation of the problem, which reduces gradient variance.

## Conclusion

The log-derivative trick is a straightforward manipulation of the derivative of a logarithm, but it provides an important route to estimating otherwise unmanageable integrals.

## References

- Shakir Mohamed's [blog post](#) on the subject.
- Mohamed, Shakir, et al. "Monte carlo gradient estimation in machine learning." arXiv preprint arXiv:1906.10652 (2019).
- Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Machine learning 8.3-4 (1992): 229-256.
- Glasserman, Paul. Monte Carlo methods in financial engineering. Vol. 53. Springer Science & Business Media, 2013.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144, 2016.
- Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712, 2016.
- Tucker, George, et al. "Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models." Advances in Neural Information Processing Systems. 2017.