Monte Carlo Gradient Estimation in Machine Learning

Introduction

- Computing the gradient of an expectation of a function is as the core of many tasks needed for the modern world to function
    - Management of supply chains
    - pricing and hedging financial instruments
    - control of traffic lights
    - ML and AI
- So figuring out how to do this is important, this paper is about methods to acheive this and trade-offs between different methods

In this paper the central question is computing $\mathcal{F}$:

$$\mathcal{F}(\theta) = \int p(\mathbf{x};\theta) f(\mathbf{x};\phi) \mathrm{d}\mathbf{x} = \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x};\phi)] \tag{1}$$

A "mean-value analysis" where we take the average of a function f with structural parameters ϕ over an input distribution $p(\mathbf{x};\theta)$ with distributional parameters θ. f is refered to as the cost and $p(\mathbf{x};\theta)$ is refered to as the measure. They restrict their review to settings where the measure is continuous on its domain and differentiable with respect to is distributional parameters. (i.e it is nicely behaved)

- This allows them to treat a wide range of problems from: queuing theory, variational inferene, portfolio management, reinforcement learning...

We want to learn the distributional parameters θ, which makes the derivative of $\mathcal{F}$ with respect to θ important to us.

Challenges:

- We can often not evaluate this expectation in closed form, $\mathbf{x}$ is high dimensional and so quadrature doesn't work well
    - Quadrature here means approximation of the integral with a function
- We may be requesting a gradient with respect to a high dimensional parameter vector
- The cost function may not be differentiable, or may be a black box (we can only observe the output and we know nothing else about the function)
- Ideally we want to have a quick (parallelisable) and accurate way of estimating this gradient, that minimises the number of evaluations of the cost

They propose to overcome this using monte carlo estimators of the integrals and gradients

Monte Carlo and Stochastic Optimisation

We can evaluate an integral like (1) by using Monte Carlo methods: We draw independent samples $\hat{\mathbf{x}}^{(1)}, \ldots, \hat{\mathbf{x}}^{(N)}$ from $p(\mathbf{x};\theta)$ and then computing:

$$\bar{\mathcal{F}}_N = \frac{1}{N}\sum_{n=1}^{N} f(\hat{\mathbf{x}}^{(n)}) \tag{2}$$

$\bar{\mathcal{F}}_N$ is a random variable as it depends on the samples from $p(\mathbf{x};\theta)$, we can repeat the process and collect multiple samples of $\bar{\mathcal{F}}_N$.

As long as we can write the integral in the form of (1) and sample from $p(\mathbf{x};\theta)$ we can utilise Monte Carlo estimation.

There are four properties that we look for in a Monte Carlo estimator:

1. Consistency:
   As the number of samples N is increased, the estimate $\bar{\mathcal{F}}_N$ should converge to $\mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x};\phi)]$. Usually this is a consequence of the law of large numbers
2. Unbiasedness:
   If we repeat this estimation process multiple times, the estimate should be centred on the actual value of the integral. I.e it should satisfy

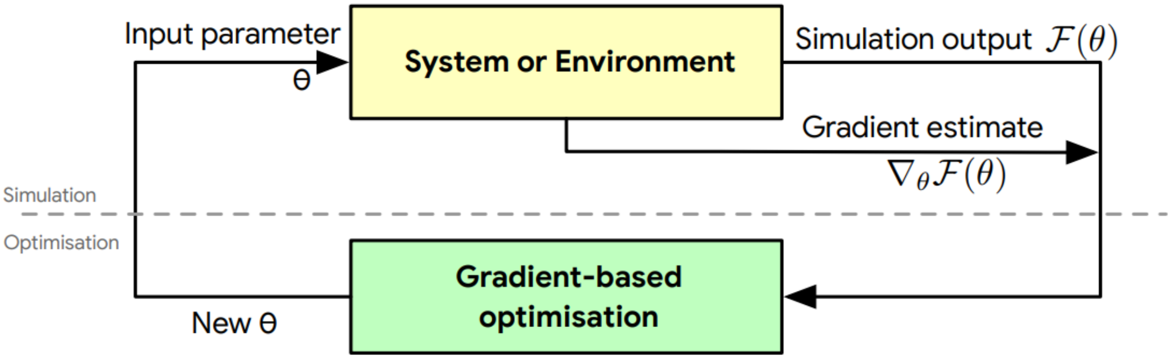
   $$\mathbb{E}_{p(\mathbf{x};\theta)}[\bar{\mathcal{F}}_N] = \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})]$$

   We want unbiased estimators as we can guarantee they converge. Sometimes we might used biased estimators on rare occasions, however.
3. Minimum Variance
   (2) is a random variable, so it stands to reason that we would prefer the estimator with the minimal variance. There are two concrete reasons we would want this:
   1. The resulting gradient estimates are more accurate
   2. Low variance gradient estimators make learning more efficient, allowing smaller learning step sizes and thus (potentially) allowing a smaller overall number of steps to reach convergence (fasting training).
4. Computational efficiency:
   For obvious reasons we will prefer the estimator that takes less computational resource to evaluate. Usually we are looking to estimators that are linear in the number of parameters, and that can parallelised.

The gradient of $\mathcal{F}$ with respect to θ gives us two things:

1. It can help explain the way the cost changes with a given parameter
2. It can be be used to *optimise* the distributional parameters θ

The typical stochastic optimisation loop is as follows


Input parameter
θ
System or Environment
Simulation output $\mathcal{F}(\theta)$
Gradient estimate
$\nabla_\theta \mathcal{F}(\theta)$
Simulation
Optimisation
Gradient-based optimisation
New θ

Note: stochastic gradient descent can be considered doubly stochastic optimisation because we are using stochastic approximation in both the simulation output ($\mathcal{F}(\theta)$) and in the optimistion step.

There are many applications of this (which the paper goes into in more depth):

- Variational Inference
- Reinforcement Learning
- Sensitivity Analysis
- Discrete Event Systems and Queuing Theory
- Experimental Design

Intuitive analysis of Gradient Estimators

Here are two ways to calculate the desired gradients:

- **Derivatives of measure**: Differentiation of the measure $p(\mathbf{x};\theta)$. Estimators of this type include score function estimator and measure-valued gradient.
- **Derivatives of Paths**: Differentiation of the cost $f(\mathbf{x})$ which "encodes the pathway from parameters θ, through the random variable $\mathbf{x}$, to the cost value." In this class ae pathwise gradient, harmonic gradient estimators and finite differences, and Malliavin-weighted estimators.

In this section of the paper they focus on *score function*, *pathwise*, and *measure-valued* gradient estimators. They have all four of the desired properties but differ on their variance properties and computational costs.

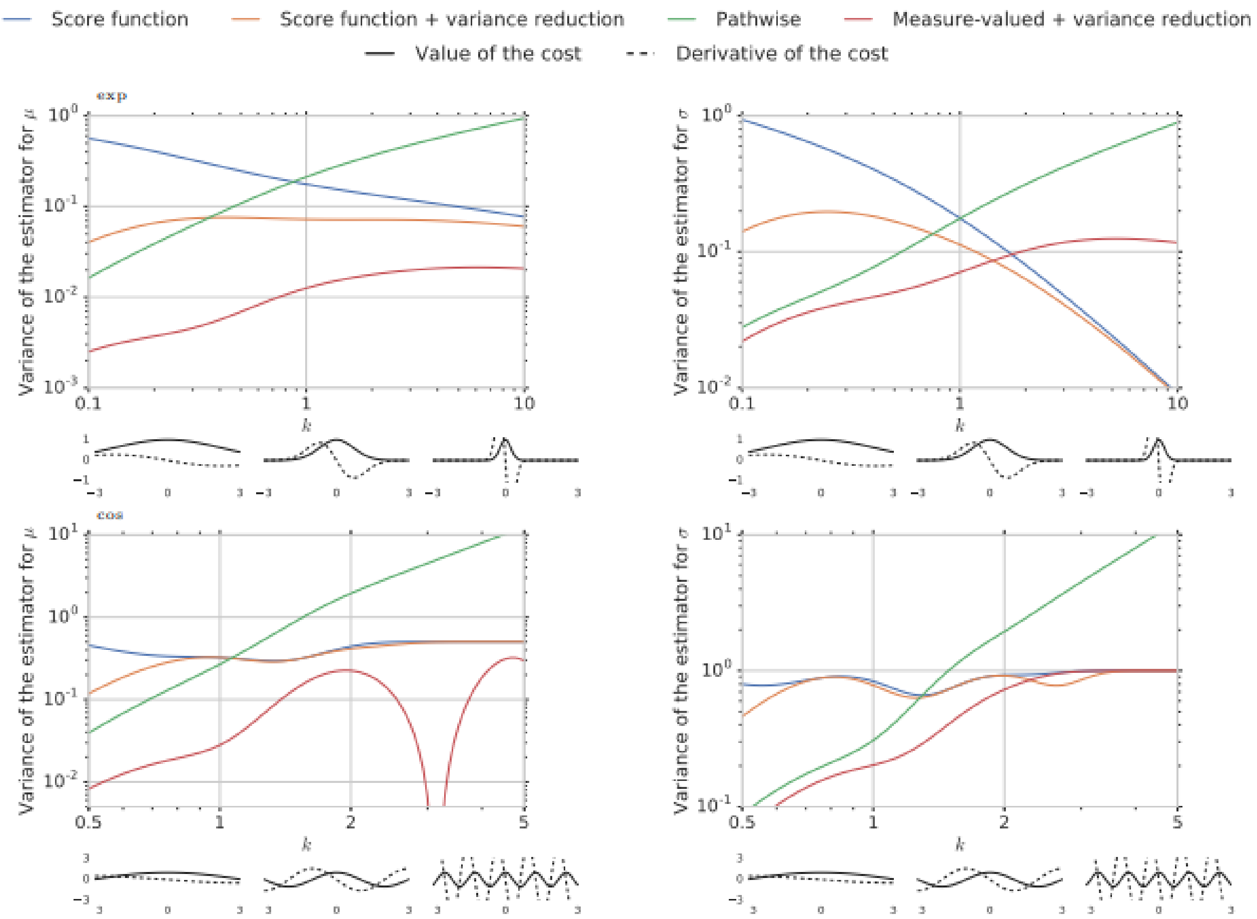


Figure 3: Variance of the stochastic estimates of $\nabla_\theta \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[f(x;k)]$ for $\mu = \sigma = 1$ as a function of k. Top: $f(x;k) = \exp(-kx^2)$, bottom: $f(x;k) = \cos kx$. Left: $\theta = \mu$; right: $\theta = \sigma$. The graphs in the bottom row show the function (solid) and its gradient (dashed): for $k \in \{0.1, 1, 10\}$ for the exponential function, and $k \in \{0.5, 1.58, 5.\}$ for the cosine function.

Takeaway from this section:

- Usually we cannot ascribe a universal ranking to gradient esimators, they will have different orderings of performance depending on the parameters of the cost.
- Measure-valued derivative estimator requires $2D$ samples for D parameters.
- If the cost function isn't differentiable, pathwise gradient won't be usable.

In short when choosing an unbiased gradient estimator we should pay attention to:

1. computational cost
2. whether the cost function is differentiable or not
3. how the gradient estimator changes as the cost function changes
4. are there effective variance reduction techniques?

Score Function Gradient Estimators

This derivative of measure-type estimator goes by:

- score function estimator
- likelihood ratio method
- REINFORCE estimator

The score function is the derivative of the log of a probability distribution, with respect to the parameters of that distribution:

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

By rearranging this we can use this identity to rewrite integrals of gradients as an expectation under the measure p.

The score function is the central quanity in maximimum likehood estimation. A property that we use later is that it's expectation is zero:

$$\mathbb{E}_{p(\mathbf{x};\theta)}[\nabla_\theta \log p(\mathbf{x};\theta)] = \int p(\mathbf{x};\theta)\frac{\nabla_\theta p(\mathbf{x};\theta)}{p(\mathbf{x};\theta)}d\mathbf{x} = \nabla_\theta \int p(\mathbf{x};\theta) d\mathbf{x} = \nabla_\theta 1 = 0$$

The variance of the score, known as Fisher information, is an important quanitity for establishing the Cramer-Rao lower bound.

Deriving the Estimator

$$\begin{aligned}\eta &= \nabla_\theta \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})] = \nabla_\theta \int p(\mathbf{x};\theta) f(\mathbf{x}) d\mathbf{x} = \int f(\mathbf{x}) \nabla_\theta p(\mathbf{x};\theta) d\mathbf{x} \\ &= \int p(\mathbf{x};\theta) f(\mathbf{x}) \nabla_\theta \log p(\mathbf{x};\theta) d\mathbf{x} \\ &= \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x}) \nabla_\theta \log p(\mathbf{x};\theta)] \\ \bar{\eta}_N &= \frac{1}{N}\sum_{n=1}^{N} f(\hat{\mathbf{x}}^{(n)}) \nabla_\theta \log p(\hat{\mathbf{x}}^{(n)};\theta); \qquad \hat{\mathbf{x}}^{(n)} \sim p(\mathbf{x};\theta)\end{aligned}$$

By using the identity above we turn an integral involving the derivative of the measure p into an expectation under that measure which we can use Monte Carlo estimation for.

We can add any constant β we like to $f(\mathbf{x})$ and still obtain an unbiased estimator, owing to the fact the score function has zero expectation. This will be shown to be useful later for a simple, effective form of variance reduction.

Estimator properties

The score-function estimator relates the overall gradient to the gradient of the log measure reweighted by the cost function. "This intuitiveness is why the score function estimator was one of the first and most widely-used estimators for sensitivity analysis."

Unbiasedness

Whenever the interchange between differentiation and integration is valid, this will yield an unbiased estimator. The interchangeability depends on conditions where it is possible to interchange limits and integrals, and is in most cases relies on the dominated convergence theorem or the Leibniz integral rule.

The following conditions must be satisfied:

- The measure $p(\mathbf{x};\theta)$ is continuously differentiable with respect to its parameters.
- The product $f(\mathbf{x})p(\mathbf{x};\theta)$ is both integrable and differentiable for all parameters.
- There exists an integrable function $g(\mathbf{x})$ such that $\sup_\theta ||f(\mathbf{x})\nabla_\theta p(\mathbf{x};\theta)||_1 \leq g(\mathbf{x}) \; \forall x$

These assumptions usually hold in machine learning applications.

Absolute Continuity

We can rewrite the estimator another way, showing the importance of absolute continuity:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})] &= \int \nabla_\theta p(\mathbf{x};\theta) f(\mathbf{x}) d\mathbf{x} \\ &= \int \lim_{h\to 0} \frac{p(\mathbf{x};\theta+h)-p(\mathbf{x};\theta)}{h} f(\mathbf{x}) d\mathbf{x} \\ &= \lim_{h\to 0} \frac{1}{h} \int p(\mathbf{x};\theta) \frac{p(\mathbf{x};\theta+h)-p(\mathbf{x};\theta)}{p(\mathbf{x};\theta)} f(\mathbf{x}) d\mathbf{x} \\ &= \lim_{h\to 0} \frac{1}{h} \int p(\mathbf{x};\theta) \left(\frac{p(\mathbf{x};\theta+h)}{p(\mathbf{x};\theta)} - 1 \right) f(\mathbf{x}) d\mathbf{x} \\ &= \lim_{h\to 0} \frac{1}{h} (\mathbb{E}_{p(\mathbf{x};\theta)}[\omega(\theta,h) f(\mathbf{x})] - \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})])\end{aligned}$$

The ratio $\omega(\theta,h)$ is similar to the one that appears in importance sampling, and makes a similar assumption of absolute continuity: $p(\mathbf{x};\theta+h) > 0$ whenever $p(\mathbf{x};\theta) > 0$, a condition that is broken when, for instance, θ controls the support of the distribution, such as the uniform distribution $\mathcal{U}[0,\theta]$.

Estimator Variance

From the work in the intuitive analysis section we can see that the variance of the score-function estimator can vary widely with the cost function.

Writing $\mu(\theta) := \mathbb{E}_{p(\mathbf{x};\theta)}[\bar{\eta}_N]$ we can write the variance of the estimator for $N=1$ as follows:

$$\mathbb{V}_{p(\mathbf{x};\theta)}[\bar{\eta}_{N=1}] = \mathbb{E}_{p(\mathbf{x};\theta)} \left[(f(\mathbf{x}) \nabla_\theta \log p(\mathbf{x};\theta))^2 \right] - \mu(\theta)^2$$

which makes the connection between the variance and the dimensionality of the parameters clear, since the derivative of the score has the same dimensionality as the parameters.

We can also write the variance as follows:

$$\mathbb{V}_{p(\mathbf{x};\theta)}[\bar{\eta}_{N=1}] = \lim_{h \to 0} \mathbb{E}_{p(\mathbf{x};\theta)} \left[(\omega(\theta, h) - 1)^2 f(\mathbf{x})^2 \right] - \mu(\theta)^2$$

which, for a fixed h, shows the dependency between variance and the importance weight ω.

There are three sources of variability in the estimator:

- Variance from the importance ratio:

For fixed h:

$$\mathbb{E}_{p(\mathbf{x};\theta)} \left[(\omega(\theta, h) - 1)^2 f(\mathbf{x})^2 \right]$$

If we have a "near-failure" of absolute continuity, where $p(\mathbf{x};\theta) \ll p(\mathbf{x};\theta + h)$ then the integral in the above expectation will be still be finite, but will be very large.

- Variance from the dimensionality of the parameters

(The derivation of this result is in the paper) "As the dimensionality increases, we find that the importance weights converge to zero, while at the same time their expectation is one for all dimensions. This difference between the instantaneous and average behaviour of $\omega(\theta, h)$ means that in high dimensions the importance ratio can become highly skewed, taking large values with small probabilities and leading to high variance as a consequence."

- Variance from the cost function

The cost function contributes to the variance, if the cost function is a sum of D terms, each which bounded variance, then the variance of the estimator will be of order $\mathcal{O}(D^2)$. One way of reducing the variance is to determine which parts of the cost function do not influence the parameters, as these will increase the variance and can be gotten rid of.

Computational Considerations

The score function can also be expressed:

$$\eta = \nabla_\theta \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})] = \mathrm{Cov}[f(\mathbf{x}), \nabla_\theta \log p(\mathbf{x};\theta)]$$

$$\mathrm{Cov}[f(\mathbf{x}), \nabla_\theta \log p(\mathbf{x};\theta)]^2 \leq \mathbb{V}_{p(\mathbf{x};\theta)}[f(\mathbf{x})] \mathbb{V}_{p(\mathbf{x};\theta)}[\nabla_\theta \log p(\mathbf{x};\theta)]$$

The score function gradient can be interpreted as a measure of covariance between the cost function (the second term the covariance introduces is zero because the expectation of the score is zero) and the second line is an application of the Cauchy-Schwartz inequality.

If the cost function is highly variable, this can lead to a highly variable gradient. To overcome this we usually try to constrain the cost function by normalisation or bounding its value via clipping.

Because only the final value of the cost is needed to calculate this estimator, we can use a wide range of potenial types of functions as a cost: differentiable functions, discrete functions, dynamical systems, black box simulators. Overall the computational cost of the score-function estimator is low: $\mathcal{O}(N(D+L))$ for D-dimentional θ, where L is the cost of evaluating the cost function, N is the number of samples used in the estimator.

Considerations to take are:

- Any type of cost function can be used, as long as we can evaluate them easily
- The measure must be differentiable with respect to its parameters
- We need to be able to sample from the measure
- This can be applied to both discrete and continuous distributions
- We can implement this estimator with a single sample if needed
- When using this estimator try to use variance reduction too.