2.4 Sensitivity of Optimisation Algorithms to Initialisation

This project requires an understanding of the Part IA Probability and Part IA Analysis courses.

1 Introduction

In modern statistics and machine learning, it is common to derive estimators and make decisions by minimising some *objective function* f. This task can generally not be solved in closed form. As such, a standard solution is to apply an iterative optimisation algorithm to attempt to find an approximate minimiser.

However, increasingly, statistical problems lead to complicated objective functions which admit multiple local minima. It is common practice to then run the optimisation algorithm numerous times from different initial conditions, in hope of finding the true optimum or a satisfactory one eventually. It is thus of interest to understand the sensitivity of our optimisation algorithms to their initialisation, and to understand which features of the objective function inform the outcomes of these algorithms.

2 Gradient Descent

The optimisation algorithm of study in this project is *gradient descent*. To run it, one must specify a differentiable *objective function* $f : \boldsymbol{D} \to \mathbf{R}$ with domain $\boldsymbol{D} \subseteq \mathbf{R}^d, d \geqslant 1$, an *initial point* $x_0 \in \boldsymbol{D}$, and a *step-size* $h > 0$. The iterates of the algorithm are then defined recursively as

$$x_t = x_{t-1} - h\nabla f(x_{t-1}) \quad \text{for } t \in \{1, 2, \ldots\}, \tag{1}$$

and the hope is that as $t \to \infty$, the iterates x_t converge (numerically) to a minimiser of f if h is sufficiently small.

In what follows, we will focus our attention on minimisation of the 'double-well' toy function f_θ, defined for $\theta \in (0, \pi)$ by

$$f_\theta : [-1, 1] \to \mathbf{R}, \quad x \mapsto \left(x^2 - \frac{3}{4}\right)^2 - x\cos\theta. \tag{2}$$

Question 1: Find the stationary points of this function in analytic form, and classify them as local minima, maxima, or saddlepoints. [*Hint: the stationary points can all be expressed as trigonometric functions; in particular, in your calculations you may use an expression for the cosine of the triple angle.*]

Given that we know the analytic form of the stationary points of f_θ, we can easily evaluate the performance of gradient descent (or indeed, any other optimisation algorithm).

Question 2: Take $\theta = \frac{\pi}{6}, h = 0.01$, and run gradient descent on $f = f_\theta$ for 1000 steps, from initial points $x_0 \in \{\frac{k}{50}\}_{k=-50}^{50}$. What do you observe about the outcomes?

3 The Monte Carlo Method

In computational settings, it is often the case that a quantity of interest ν is naturally expressed as an expectation, i.e. there is some random variable X and some function g such that

$$\nu = \mathbf{E}\left[g(X)\right]. \tag{3}$$

The *Monte Carlo method* (MC) involves drawing N independent samples $\{X^i\}_{i=1}^N$ which are distributed like X, and forming the estimator

$$\hat{\nu}_N \triangleq \frac{1}{N}\sum_{i=1}^{N} g(X^i). \tag{4}$$

Question 3: Show that $\hat{\nu}_N$ is *unbiased* for ν, i.e. $\mathbf{E}\left[\hat{\nu}_N\right] = \nu$. Assuming that $\mathbf{Var}(g(X)) < \infty$, obtain an expression for the variance of $\hat{\nu}_N$.

Returning to our toy function above, fix $\theta \in (0, \pi)$, $f = f_\theta$, and let $\{X_t^h\}_{t=0,1,\ldots}$ be the sequence of random variables obtained by i) sampling an initial point $X_0^h \sim \text{Uniform}([-1, 1])$, and ii) iterating $X_t^h = X_{t-1}^h - h\nabla f(X_{t-1}^h)$.

For some $T, h > 0$ such that $Th^{-1} \in \boldsymbol{N}$, we are interested in studying the behaviour of

$$\mu^h \triangleq \mathbf{E}\left[X_{Th^{-1}}^h\right] \qquad \text{and} \qquad \mu \triangleq \lim_{h \to 0^+} \mu^h, \tag{5}$$

i.e. the outcome of running gradient descent from a *randomised* initial point, using smaller and smaller step-sizes, and run for longer and longer. We take $T = 10$ fixed throughout, as in this example, this is approximately sufficient for convergence to take place.

A basic approach to estimating μ is to take h as small as possible, and to estimate μ^h as accurately as possible by the Monte Carlo estimate $\hat{\mu}_N^h$ by taking N as large as possible.

Question 4: Test this method out: fix $\theta = \frac{\pi}{4}$, and for $k \in \{0, 1, \cdots, 10\}$, take $h = 0.1 \cdot 2^{-k}$, and estimate μ^h using $N_k = 2^{20-k}$ samples, so that the same amount of computational time is used for each k. What do your estimates suggest about the behaviour of μ^h as h decreases? What do they suggest about the variance of $X_{Th^{-1}}^h$ as h decreases?

In this approach, because h is not exactly 0, we incur a finite *bias*, i.e. even in the limit of infinitely many samples, our estimator would converge to $\mu^h \neq \mu$. As such, the variance of our estimator would not fully reflect its accuracy. Instead, it is standard to use the following more general measure of accuracy. The *mean squared error* (MSE) of an estimator T of a quantity τ is defined by

$$\mathbf{MSE}(T; \tau) \triangleq \mathbf{E}[(T - \tau)^2]. \tag{6}$$

Question 5: Prove the 'bias-variance decomposition', i.e. show that

$$\mathbf{MSE}(T; \tau) = (\mathbf{E}[T] - \tau)^2 + \mathbf{Var}(T). \tag{7}$$

We present the following facts without proof: for h sufficiently small, there are constants $A_1, A_2, A_3 \in (0, \infty)$ such that

1. the bias of the approximation μ^h is bounded as $|\mu^h - \mu| \leqslant A_1 h$,
2. the variance of $X_{Th^{-1}}^h$ is bounded as $\mathbf{Var}(X_{Th^{-1}}^h) \leqslant A_2$, and
3. for $t \in \{0, 1, \ldots\}$, the cost of generating a sample of X_t^h satisfies $\mathbf{Cost}(X_t^h) = A_3 t$.

Question 6: Suppose we estimate μ by fixing $h > 0, N \in \mathbf{N}$, generating N i.i.d. samples $\{Y^i\}_{i=1}^N$ distributed as $X^h_{Th^{-1}}$, and forming the estimator

$$\hat{\mu}^h_N = \frac{1}{N}\sum_{i=1}^{N} Y^i. \tag{8}$$

Use the bias-variance decomposition to show that the MSE of $\hat{\mu}^h_N$ can be bounded above by $A_1^2h^2 + \frac{A_2}{N}$. Suppose now that the computational budget is C, i.e. the cost of generating all of the random variables used in the MC estimator is bounded above by C. Assume that we use our full budget, i.e. we choose (N, h) such that $N \cdot \frac{A_3 T}{h} = C$. Use this to express the upper bound on the MSE as a function of only h, and derive the h which minimises this upper bound. How does the optimal MSE scale with C?

4 Multi-Level Monte Carlo

For a given computational budget C, it is possible construct estimators with less variability than $\hat{\mu}^h_N$, and hence improve our accuracy. We exploit the intuition that if the initial point x_0 is fixed, we expect that the paths of X^h_t and $X^{h/2}_{2t}$ will stay close together, and thus that $\mu^h \approx \mu^{h/2}$.

In order to justify this later on, we introduce an extra fact without proof: for h sufficiently small, there is a constant $A_4 \in (0, \infty)$ such that

4. if two sequences of gradient descent iterates have the same initial point $X_0 \sim \text{Uniform}([-1, 1])$, then $\mathbf{Var}\left(X^h_{Th^{-1}} - X^{h/2}_{2Th^{-1}}\right) \leqslant A_4 h^2$.

We note quickly that the facts presented before Question 6 remain true in what follows.

For $X_0 \sim \text{Uniform}([-1, 1])$ and $l = 0, \ldots, L$, $L \in \mathbf{N}$, define $h_l = 0.1 \times 2^{-l}$, let $X^{h_l}_{Th_l^{-1}}$ be the $(Th_l^{-1})^{\text{th}}$ gradient descent iteration for f_θ, with $X^{h_l}_0 = X_0$, and with step-size h_l. Define the random variables

$$Y_0 = X^{h_0}_{Th_0^{-1}} \qquad \text{and} \qquad Y_l = X^{h_l}_{Th_l^{-1}} - X^{h_{l-1}}_{Th_{l-1}^{-1}}, \quad l = 1, \ldots, L. \tag{9}$$

We can then formally write that

$$\mu = \sum_{l \geqslant 0} \mathbf{E}\left[Y_l\right]. \tag{10}$$

Question 7: Justify that the above sum converges absolutely, and find an upper bound for the truncation error incurred by approximating $\mu \approx \sum_{l=0}^{L} \mathbf{E}\left[Y_l\right]$.

With this in mind, we can aim to approximate μ by taking a *truncation level* L, a sequence of *level sizes* $\{N_l\}_{l=0}^L$, and forming the *Multi-Level Monte Carlo estimator* (MLMC)

$$\hat{\mu}_{N_{1:L}} \triangleq \sum_{l=0}^{L} \left[\frac{1}{N_l}\sum_{i=1}^{N_l} Y_l^i\right]. \tag{11}$$

where for each i, $\{Y_l^i\}_{i=1}^{N_l}$ are independent, identically-distributed (iid) samples of Y_l, i.e., for each (i, l) we independently draw an initial point $X_0^{(i,l)} \sim \text{Uniform}([-1, 1])$ and define Y_l^i as in display (9) using $X_0^{(i,l)}$ rather than X_0. Hence, $\{Y_l^i\}_{i,l}$ are mutually independent.

Question 8: For $\theta \in \left\{\frac{k\cdot\pi}{2^7}\right\}_{k=1}^{2^6}$, compute $\hat{\mu}_{N_{1:L}}$, taking $L = 10$ and using level sizes i) $N_l \equiv 5$ and ii) $N_l = 2^{L-l}$. Which estimator exhibits greater variability?

Question 9: Derive an upper bound for the MSE of the MLMC estimator with truncation level L and level sizes $\{N_l\}_{l=0}^{L}$.

We now try to choose L and $\{N_l\}_{l=0}^{L}$ such that the MSE of the resulting MLMC estimator is minimised given a fixed computational budget.

Question 10: Suppose that the computational budget is C, i.e. the cost of generating all of the random variables used in the MLMC estimator is bounded above by C. By treating the N_l as continuous variables, $L = \infty$, derive an allocation of level sizes $\{\tilde{N}_l\}_{l\geqslant 0}$ which minimises the upper bound for the MSE of the resulting MLMC estimator derived in question 9. [*Hint: To minimise a function $F(x)$ subject to the constraint that $G(x) = c$, it suffices to identify stationary points of $H(x, \lambda) = F(x) + \lambda(G(x) - c)$. This is known as the method of Lagrange multipliers.*]

From now on, we move back to integer-valued levels by taking $N_l = \lfloor \tilde{N}_l \rfloor$.

Question 11: How do you find that L scales with C? Derive an expression for how the optimal MSE scales as C grows, and compare this to the MC estimator from Question 5.

5 Application to Double-Well Loss Function

We will now use the estimators derived above to study the behaviour of gradient descent on the double-well function f_θ defined earlier.

Question 12: Define $m_1(\theta)$ and $m_2(\theta)$ as the local *minima* of f_θ in $[-1, 1]$, defined such that $m_1(\theta) < m_2(\theta)$. Suppose that $h > 0$ and $T \in \mathbf{N}$ are sufficiently small and large, respectively, so that $\min\{|m_1(\theta) - X_T^h|, |m_2(\theta) - X_T^h|\} \approx 0$ for any initial point $x_0 \in [-1, 1]$. Define

$$p_1(\theta) = \mathbf{P}\left(\lim_{T\to\infty} X_T^h = m_1(\theta)\right) \qquad \text{and} \qquad p_2(\theta) = \mathbf{P}\left(\lim_{T\to\infty} X_T^h = m_2(\theta)\right). \tag{12}$$

Derive an expression for $p_1(\theta)$ and $p_2(\theta)$ in terms of $\mu, m_1(\theta)$ and $m_2(\theta)$.

Question 13: Use a Multi-Level Monte Carlo scheme with the optimal level sizes, as derived in Question 9, to form estimates of $p_1(\theta)$ and $p_2(\theta)$ for $\theta \in \left\{\frac{k\cdot\pi}{2^7}\right\}_{k=1}^{2^6}$. Plot your estimates to show how they vary with θ. For which values of θ does the outcome of running gradient descent on f_θ vary most?

