I (Constrained NORTA) Suppose \( \alpha = \mathbb{E}[h(X) \mathbb{I}(X \in S)] \), where \( S := \{ z \in \mathbb{R}^d : Az \leq b \} \) (a polyhedron), \( X \) has the NORTA distribution, and \( h : \mathbb{R}^d \to \mathbb{R} \) is some real-valued function. Assume that the set \( S \) is a “rare set,” and thus the quantity \( \alpha \) is small. You seek the optimal importance sampling twist (for generating \( X \)), i.e., one that will minimize the resulting variance of the importance sampling estimator \( \hat{\alpha} \).

Note the following.

1. The straightforward way of attempting to solve this problem is to write an expression for the variance of the importance sampling estimator and then formulating the first order conditions for a minimum. Of course, the first-order conditions will not yield an analytic solution. So, one strategy might be to solve the first-order conditions iteratively, thus identifying the optimal twist progressively.

2. You might start with an even simpler version of the above problem: assume that \( X \) is multivariate normal with zero mean and known covariance matrix \( \Sigma \), and the set \( S \) is a cone with vertex at the point \( z_0 \).

II (Optimally Combined Directional Derivatives) Suppose you wish to estimate the gradient \( \nabla f(\theta_0) \) of a function \( f : \mathbb{R}^d \to \mathbb{R} \) at the point \( \theta = \theta_0 \). Assume \( f(\theta) = \mathbb{E}[Y(\theta)] \) and that \( Y(\cdot) \) is observable through a simulation. Consider the following alternative to forward and central difference approximation. Simulate \( m \) times at \( \theta = \theta_0 \) and at each of the points \( \theta_1, \theta_2, \ldots, \theta_k \). Now use the pairs of points \( (\theta_0, \theta_i), i = 1, 2, \ldots, k \) to construct directional estimators \( \hat{\nabla} f_{di} = \frac{f(\theta_i) - f(\theta_0)}{\| \theta_i - \theta_0 \|} u_i, i = 1, 2, \ldots, k \), where \( u_i \) is the unit vector along \( \theta_i - \theta_0 \). Now, construct the partial derivative along any axis by projecting each of \( \hat{\nabla} f_{di}, i = 1, 2, \ldots, k \) and then appropriately weighting them.

Numerous questions can be asked in the above context.

1. How should the directional estimators be weighted to minimize the mean squared error of the resulting estimator?

2. Where should the \( k \) points \( \theta_1, \theta_2, \ldots, \theta_k \) be placed to minimize the resulting mean squared error?

3. Can a CLT be proved for the resulting estimator?

3. Notice that in the above method, the total simulation effort is \( m \times (k + 1) \). A more straightforward approach may be to randomly generate \( m \times (k + 1) \) points in a ball of radius \( h \) around \( \theta \), run the simulation at each of these points once, construct a regression estimator using the observations, and then finally extract the derivative from the constructed model. Will this be a better estimator?
III (Portfolio Optimization) Assume that you have a portfolio of $k$ assets whose value evolves in time according to a known $k$-dimensional geometric Brownian motion. You have a unit resource that needs to be allocated across these $k$ assets. (For simplicity, assume that the initial unit cost of each of the assets is the same.) If you are interested in maximizing the expected revenue at the end of a specified time horizon $[0, T]$, subject to the variance of the revenue being below a specified threshold, how should the unit resource be allocated across the various assets?

1. Assume that the parameters of the stochastic differential equation governing the asset movement is such that there exists no closed-form solution for the distribution of the asset value at time $T$. So, the only way of estimating the moments of the revenue is simulation, and your solution will necessarily be algorithmic.

2. One method is to assume that at optimality, the constraint on the variance will be binding. Then, pose the optimization problem using a Lagrangian formulation and write the first-order conditions for optimality. This will give you an idea of how to construct a recursive solution for the optimal allocation.