On Bayesian survival analysis and the stochastic mesh method*

Erwann Rogard

2022-04-05

Abstract

Both Sequential clinical trials and (American) option pricing are optimal stopping problems. There exists well documented simulation methods for the latter: are they applicable to the former? Such was the topic of my MPhil examination in December 2005. This is a follow up paper originally dated 2006-11-08. Since it was never made public, the new date is appropriate. The specific method under consideration is the stochastic mesh. It addresses the computational challenge inherent to the intuitive random tree method, by using a combination of independently simulation paths, and likehood ratios linking each node to those in the next period. section 1 summarizes the existing literature, and goes on to discuss the Bayesian setting. section 2 addresses parametric boundaries.

1 Problem formulation

Consider a sequential clinical trial extending over a maximum of i_* periods to assess the effect θ of a treatment T, in comparison to the placebo P. A prior $p(\xi|\mathbf{y}_0)$ and a likelihood $p(\mathbf{y}_i|\xi)$ are assumed, where $\xi = (\phi, \theta)$, ϕ representing a nuisance parameter and $\mathbf{y}_i = (y_0, ..., y_i)$ is data accumulated up to the i^{th} trial, $0 \leq i \leq i_*$. If the i^{th} trial is carried out, the action space is $\mathcal{A} = \{T, P, C\}$, where C means continue to $i+1$, for $0 \leq i < i_*$. For $i = i_*$ the relevant space is $\mathcal{A}^s = \{T, P\}$.

Let $u_i(\theta, a)$ denote utility for period i and $\mathbf{a}_i = \{a_j\}_{j=0}^i$ the sequence of actions up to period i. Until we stop, utility incorporates a cost of sampling, c_i, so that $u_i(\theta, C) = -c_i$, $u_i(\theta, T) = t(i, \theta) - c_i$ and $u_i(\theta, P) = p(i, \theta) - c_i$ for some functions p and t.

*Revisions:`https://github.com/erwannr/statistics/commits/main/stochmesh`

To make comparisons across periods we set $i = 0$ as the reference point, and decide on a discounting factor β, $0 < \beta \leq 1$, such that incurring $u_i(\theta, a)$ at period i is worth $\beta^j u_i(\theta, a)$ at time 0. For a given sequence of actions $\mathbf{a}_i$, the total utility is $u(\theta, \mathbf{a}_i) = \sum_{j=0}^{i} \beta^j u_j(\theta, a_j)$.

Based on the revised distribution at the i^{th} trial, $p(\theta|\mathbf{y}_i)$, and assuming $a \neq C$, we have to solve $\max_{a \in \mathcal{A}^s} \mathrm{E}[u(i, \theta, a)|\mathbf{y}_i]$, where $u(i, \theta, a) = u(\theta, \{\mathbf{a}_i : a_j = C, j < i; a_i = a \in \mathcal{A}^{\mathrm{s}}\})$. Let $u(\mathbf{y}_i)$ and $a^{\mathrm{s}}(\mathbf{y}_i)$ denote the corresponding optimal utility and solution.

The full problem can be characterized by the Bellman recusion (BR):

$$u_*(\mathbf{y}_n) = u(\mathbf{y}_n) \tag{1}$$
$$\upsilon(\mathbf{y}_i) = \mathrm{E}[u_*(\mathbf{y}_{i+1})|\mathbf{y}_i] \tag{2}$$
$$\upsilon_*(\mathbf{y}_i) = \max(u(\mathbf{y}_i), \upsilon(\mathbf{y}_i)) \tag{3}$$

for $i = i_*, ..., 0$ where $\upsilon(\mathbf{y}_i)$ and $u_*(\mathbf{y}_{i+1})$ are the expected utility, form moving to $i + 1$ (and pursuing an optimal strategy thereafter), and from making the overal best decision. Clearly, $u_*(\mathbf{y}_i) = \upsilon(\mathbf{y}_i) \Rightarrow a(\mathbf{y}_i) = C$, otherwise $a(\mathbf{y}_i) = a^{\mathrm{s}}(\mathbf{y}_i)$.

In principle, we can solve the above problem as follows:

Algorithm 1 (BR).

1. Fix m and starting from $\mathbf{y}_0$, sample recursively $y_{i+1}^{j_1,...,j_i,j_{i+1}} \overset{\text{iid}}{\sim} p(y_{i+1}|\mathbf{y}_i^{j_1,...,j_i})$, $j_{i+1} = 1, ..., m$, $i = 0, ..., i_* - 1$, resulting in a tree structure rooted at $\mathbf{y}_0$ and m^{i_*} terminal nodes

2. At all the nodes at $i = i_*$ compute (1) and for every other node, in the order $i = i_* - 1, ..., 0$ compute (2–3) with (2) approximated as

$$\hat{\upsilon}(\mathbf{y}_i^{j_1,...,j_i}) = \frac{1}{m} \sum_{j=1}^{m} \hat{u}_*(\mathbf{y}_{i+1}^{j_1,...,j_i,j}) \tag{4}$$

This has the following characteristic:

1. The simulated tree requires a computational budget exponential in i_*

2. The relevant distribution, at each node, is the predictive density $p(y_{i+1}|\mathbf{y}_i)$

3. A trivial optimization problem, (3) is solved at each node of the tree

4. The method is independent of the problem's structure (model and loss)

5. The bias of (4) is positive for finite m, and zero in the limit as $m \to \infty$

Due to the first characteristic the algorithm is impractical beyond 2 or 3 periods for a personal computer. However, the memory requirement can be made linear in i_* by depth first processing.

1.1 Independent paths alternative

To remedy the first limitation of *Algorithm* 1 we replace the simulated tree by independent paths and correct by importance sampling:

$$\mathrm{E}[u_*(\mathbf{y}_{i+1})|\mathbf{y}_i] = \mathrm{E}_*[u_*(\mathbf{y}^*_{i+1})\frac{p(\mathbf{y}^*_{i+1}|\mathbf{y}^*_i = \mathbf{y}_i)}{p(\mathbf{y}^*_{i+1})}] \tag{5}$$

where $\mathbf{y}^*_i$ indicates a variable independent of $\mathbf{y}_i$ but with the same distribution. Here we should understand $\mathbf{y}_i$ as the sufficient statistics of the data for θ, otherwise we have $p(\mathbf{y}^*_{i+1}|\mathbf{y}^*_i = \mathbf{y}_i) = 0$ unless the sample $\mathbf{y}^*_i$ agrees exactly with $\mathbf{y}_i$. Convergence is obtained if $p(\mathbf{y}_{i+1}|\mathbf{y}_i, ..., \mathbf{y}_1) = p(\mathbf{y}_{i+1}|\mathbf{y}_i)$[3, Section 8.5.1].

The new procedure is

Algorithm 2 (BR).

1. Simulate m_* iid paths with transition $p(\mathbf{y}_{i+1}|\mathbf{y}_i), i = 0, ..., i_* - 1$

2. At all the nodes at $i = i_*$ compute (1) and for every other node, in the order $i = i_* - 1, ..., 0$ compute (2–3) with (2) approximated as

$$\hat{v}(\mathbf{y}^j_i) = \frac{1}{m_*}\sum_{m=1}^{m_*} w(\mathbf{y}^m_{i+1}|\mathbf{y}^j_i)\hat{u}_*(\mathbf{y}^j_{i+1}) \tag{6}$$

$$w(\mathbf{y}^m_{i+1}|\mathbf{y}^j_i) = \frac{p(\mathbf{y}^m_{i+1}|\mathbf{y}^j_i)}{p(\mathbf{y}^m_{i+1})} \tag{7}$$

At $i = 0$ there is only one node. Both formula are correct but (7) simplifies to $w(\mathbf{y}^m_1|\mathbf{y}_0) = 1$. For each $i = 1, ..., i_* - 1$ we have to compute $m \times m$ weights, that is as many conditional densities $p(\mathbf{y}^m_{i+1}|\mathbf{y}^j_i)$ and m_*

marginals $p(\mathbf{y}_{i+1}^m)$. The computational demand is therefore of the order $m^2(i_* - 1)$.

This algorithm assumes that we can sample from and evaluate $p(\mathbf{y}_{i+1}|\mathbf{y}_i)$, and evaluate $p(\mathbf{y}_{i+1})$. Except for the simplest models, integrating out the parameter $\xi = (\phi, \theta)$, cannot be done analytically, which leads to the next Section.

1.2 Approximating unknown densities

The product of the transition densities in step 1 of *Algorithm* 2 equals the marginal density of data: $\Pi_{i=1}^{i_*} p(\mathbf{y}_{i+1}|\mathbf{y}_i) = p(\mathbf{y}_{i_*})$. In the iid case, $p(\mathbf{y}_{i_*}) = \mathrm{E}[\Pi_{i=1}^{i_*} p(y_i|\xi)]$. Replacing step 1 by

$$\xi^j \sim p(\xi^j|y_0) \tag{8}$$

$$(y_1^j, ..., y_{i_*}^j) \sim \Pi_i p(y_i^j|\xi) \tag{9}$$

much simplifies the algorithm in the case where the transition densities $p(\mathbf{y}_{i+1}|\mathbf{y}_i)$ are not analytically known. However, the latter are still present in the importance ratio. We can use simulation to approximate each integral of the ratio of densities:

$$w(\mathbf{y}_{i+1}^m|\mathbf{y}_i^j) = \frac{\mathrm{E}[p(\mathbf{y}_{i+1}^m|\xi)|\mathbf{y}_i^j]}{\mathrm{E}[p(\mathbf{y}_{i+1}^m|\xi)]} \tag{10}$$

It seems too costly to sample new draws to evaluate each expectation but we can instead reuse the existing draws $\{\theta^l\}_{l=1}^m$ from $p(\theta)$ and correct with importance sampling:

$$\mathrm{E}[p(\mathbf{y}_{i+1}^m|\xi)|\mathbf{y}_i^j] = \mathrm{E}[p(\mathbf{y}_{i+1}^m|\xi)\frac{p(\xi|\mathbf{y}_i^j)}{p(\xi)}] \tag{11}$$

$$= \mathrm{E}[p(\mathbf{y}_{i+1}^m|\xi)p(\mathbf{y}_i^j|\xi)]/p(\mathbf{y}_i^j) \tag{12}$$

Therefore, the appropriate weight is

$$w(\mathbf{y}_{i+1}^m|\mathbf{y}_i^j) \approx \frac{1}{m_*}\sum_{l=1}^{m_*}\frac{p(\mathbf{y}_i^j|\xi^l)}{p(\mathbf{y}_i^j)}p(\mathbf{y}_{i+1}^m|\xi^l)/\frac{1}{m_*}\sum_{l=1}^{m_*}p(\mathbf{y}_{i+1}^m|\xi^l) \tag{13}$$

$$\approx \frac{1}{m_*}\sum_{l=1}^{m_*}\frac{p(\mathbf{y}_i^j|\xi^l)p(\mathbf{y}_{i+1}^m|\xi^l)}{\frac{1}{m_*}\sum_{l=1}^{m_*}p(\mathbf{y}_i^j|\xi^l)}/\frac{1}{m_*}\sum_{l=1}^{m_*}p(\mathbf{y}_{i+1}^m|\xi^l) \tag{14}$$

The first line is not practically useful as $p(\mathbf{y}_{i+1}^j)$ is not analytically known. The second line, only involves likelihood calculations which, in general, can

be computed exactly. The implication is that this modified algorithm is in principle applicable to arbitrarily complex Bayesian models. In practice, there are two limitations:

1. Although the number of likelihood computations remains of order $i_* \times m_*^2$ the order of the number of algebraic operations is increased by m_* to $i_* m_*^3$.

2. To obtain a satisfactory approximation, we need $p(\xi) \gg p(\xi|\mathbf{y}_{i+1})$ but also $p(\xi)$ close to $p(\xi|\mathbf{y}_{i+1})$. The second requirement is unlikely for large i.

1.2.1 Numerical aspect

We now discuss the first limitation: whereas (7) involved only a ratio of two terms, (14) involves two sequences of likelihood computations, $p(\mathbf{y}_i^j|\xi^1), ..., p(\mathbf{y}_i^j|\xi^{m_*})$ and $p(\mathbf{y}_{i+1}^m|\xi^1), ..., p(\mathbf{y}_{i+1}^m|\xi^{m_*})$. For a given (j, m), the last sequence becomes the first sequence at $i+1$. Therefore, the total number of likelihood computations for (6) is $i_* \times m_*^2$, whose order is unchanged compared with that using (7). There are $i_* \times m_*$ marginals, each of which requires m_* "+" operations to be approximated. There are another m_* "$\times$" and "+" operations for each weight. In all, the number of "+" and "$-$" operations is of order $i_* \times m^3$. We will discuss model specific approximations to speed up this part of the algorithm.

1.2.2 Distributional aspect

We now discuss the second problem. As more data is sampled $p(\xi|\mathbf{y}_{i+1})$ will be more concentrated around the data point $\mathbf{y}_{i+1}$ and farther from $p(\xi)$. This makes it more likely that a region with high probability under $p(\xi|\mathbf{y}_{i+1})$ has a very small probability under $p(\xi)$ resulting in a poor estimate (14) and therefore a poor estimate for (6).

To address this issue, we now assume that the $\{\xi^m\}_{m=1}^{m_*}$'s are not sampled from $p(\xi^m)$ but from an arbirary $q(.) \gg p(.)$ whose normalizing constant need not be assumed known. The appropriate weight, therefore, is

$$w(\mathbf{y}_{i+1}^m|\mathbf{y}_i^j) \approx \frac{\frac{1}{m}\sum_{l=1}^{m_*} p(\mathbf{y}_{i+1}^m|\xi^l)p(\mathbf{y}_i^j|\xi^l)p_{\text{un}}(\xi^l)/q_{\text{un}}(\xi^l)}{\frac{1}{m}\sum_{l=1}^{m_*} p(\mathbf{y}_i^j|\xi^l)p_{\text{un}}(\xi^l)/q_{\text{un}}(\xi^l)} \Big/ \frac{\frac{1}{m}\sum_{l=1}^{m_*} p(\mathbf{y}_i^m|\xi^l)p_{\text{un}}(\xi^l)/q_{\text{un}}(\xi^l)}{\frac{1}{m}\sum_{l=1}^{m_*} p_{\text{un}}(\xi^l)/q_{\text{un}}(\xi^l)} \tag{15}$$

where $p_{\text{un}}(.)$ and $q_{\text{un}}(.)$, are the un–normalized densitities. Compared with (14), each summand is now multiplied by $p_{\text{un}}(\xi^l)/q\text{un}(\xi^l)$ and the order of computation is unchanged.

For a given (i, j), we have to choose $q(.)$ "bridging" $p(\xi)$ and $p(\xi|\mathbf{y}_i^j)$, for example $q(.) \propto p(.)^{1-\beta} p^{\beta}(.|\mathbf{y}_i^j)$ for some $0 < \beta < 1$. If we were to generate a new sample $\{\xi\}_{m=1}^{m_*}$ taylored to each (i, j), this would defeat the purpose of importance sampling, whose benefit stems from reusing samples. Moreover, if we fix a path j, it is seems quite plausible that the sample $q(.)$ constructed from bridging the two "end points" distributions $p(.)$ and $p(|\mathbf{y}_{i_*}^j)$ will also serve as a good bridge between $p(.)$ and $p(.|\mathbf{y}_i^j)$ for $1 < i < i_*$.

1.3 Stopping rule formulation

We define

$$\tau_* \triangleq \min\{i : u(\mathbf{y}_i) > u(\mathbf{y}_i)\} \tag{16}$$

and $u(\theta, \mathbf{y}_i) \triangleq u(i, \theta, a(\mathbf{y}_i))$. In view of (1–3), $\upsilon(\mathbf{y}_i) = \mathrm{E}[u(\theta, y_{\tau_*})|\mathbf{y}_i]$, and in particular,

$$\upsilon \triangleq \upsilon(\mathbf{y}_0) = \mathrm{E}[u(\theta, \mathbf{y}_{\tau_*})] \tag{17}$$

where E[] is understood as condition on y_0 to alleviate subsequent notation. With the above stopping rule, a given path $\mathbf{y}$ is either stopped before i, at i or beyond i. This defines a partition $\mathcal{Y}_i = \mathbf{C}_{i-1}^{\mathrm{c}} \cup \mathbf{S}_i \cup \mathbf{C}_i$. The relationships between the sets are $\mathbf{C}_i = \mathbf{C}_{i-1} \cup C_i$, for $i = 1, ..., i_* - 1$, $\mathbf{C}_0 = C_0$ and $\mathbf{S}_i = \mathbf{C}_{i-1} \cap C_{i-1}^{\mathrm{c}}$ where

$$C_i = \{\mathbf{y}_i : \upsilon(\mathbf{y}_i) > u(\mathbf{y}_i), \quad i = 0, ..., i_* - 1\} \tag{18}$$

$$C_n = \{\emptyset\} \tag{19}$$

We can estimate the stopping rule as follows. Suppose we have generated $j = 1, ..., m$ paths and computed $(u(\mathbf{y}_i^j), \hat{u}(\mathbf{y}_i^j))$ by BR–IS, for each node (i, j). For any given $\mathbf{y}_i^*$, independent of the previous draws, we can estimate $\upsilon(\mathbf{y}_i^*)$ by plugging $\mathbf{y}_i^*$ in place of $\mathbf{y}_i^j$ into (6) and the latter into (16) results in an estimate $\hat{\tau}$, which by (17) gives

$$\upsilon(\hat{\tau}) \triangleq \mathrm{E}[u(\theta, \mathbf{y}_{\hat{\tau}})] \tag{20}$$

$$< \upsilon \tag{21}$$

As we recall, the BR (and therefore BR–IS) estimates υ with a positive bias, whereas (20) yields a negative bias. By combining the two estimators, we

can construct a confidence interval which contains the true value υ with a given confidence level.

The second way to estimate the stopping rule is to model it. Let $C = (C_1, ..., C_{i_*-1})$ and suppose we postulate $\mathbf{C} : \mathbf{\Gamma} \to \mathcal{Y}^{i_*-1}$. As we saw, the continuation regions determine the stopping rule, which in turn determine a continuation value:

$$\tau(\gamma) = \min\{i : \mathbf{y}_i \notin C_i(\gamma)\} \tag{22}$$

$$\upsilon(\gamma) = \mathrm{E}[u(\theta, \mathbf{y}_{\tau(\gamma)})] \tag{23}$$

Our objective is the maximization of (23). The model is at best as good as the true optimal rule:

$$\sup_{\gamma \in \Gamma} \upsilon(\gamma) \leq \upsilon \tag{24}$$

Recalling the characteristics of BR at the end of the previous Section, let us now contrast them with those of SR:

1. Estimating the expection in (23) requires parallel paths
2. The relevant distribution is $p(\theta, \mathbf{y}_\tau)$
3. We have to optimize over all entries of γ simultaneously
4. The structure of the optimization problem depends on the distribution and utility under consideration

The first characterisitc alleviates the need for the tree structure of BR, while the second one bypasses the difficulties of BR–IS in the case where transition and marginal densities of data are not known. This comes at the cost of having to find an appropriate model and an appropriate optimization procedure. A minor restriction, is that $p(\theta)$ be proper. Practically, this often means restricting the search to conjugate priors, or otherwise incorporating some data into the prior[2, Section 4.3].

In general, the different portions of the continuation region are parameterized separately i.e. $C_i(\gamma) = C_i(\tilde{\gamma}_i)$ where $\tilde{\gamma}_i$ does not overlap with any of the $\tilde{\gamma}_j$'s. We call $\boldsymbol{\gamma}_i$ the vector γ truncated after the ith continuation region, such that $\mathbf{C}_i = \mathbf{C}(\boldsymbol{\gamma}_i)$.

2 Solving the parametric boundaries problem

The quantity to maximize,(23), is an expectation which in general has to be approximated by a simulation average. One approach therefore, it to

sample a fixed number of draws, and vary γ in the search for the optimum. Because the resulting function approximation is not smooth, it has to be approximated in successive steps, usually one dimension at at time. This method may not converge to the desired solution[3]. Moreover, we cannot easily control the degree of accuracy if the number of paths is kept fixed.

It may be better to explicitly recognize the stochastic nature of the problem and solve it in an iterative fashion that allows for termination based on some convergence criterion. We start from an initial guess $\gamma^{(0)}$, and update it so as to maximize a local approximation to (23):

$$\gamma^{(b+1)} = \Pi_\Gamma(\gamma^{(b)} - \alpha^{(b)} H^{(b)^{-1}} \hat{\nabla}^{(b)} \upsilon) \tag{25}$$

where $\alpha^{(b)}$, $H^{(b)}$ and $\hat{\nabla}^{(b)}\upsilon$ are the step size, scaling matrix and an approximation to the gradient, respectively.

Recall that the vector parameter γ determines the continuation region, or equivalently the stopping regions, $\mathbf{C}_i(\boldsymbol{\gamma}_i)$ and $\mathbf{S}_i(\boldsymbol{\gamma}_i)$, respectively. We will assume that within $\mathbf{S}_i(\boldsymbol{\gamma}_i)$, γ also determine the treatment and placebo regions, $T_i(\gamma)$ and $P_i(\gamma)$. Consider the following decomposition:

$$\upsilon(\gamma) = \sum_{i=1}^{i_*} \sum_{a \in \{T,P\}} \upsilon_{i,a}(\boldsymbol{\gamma}_i) \tag{26}$$

$$\upsilon_{i,a}(\boldsymbol{\gamma}_i) = \mathrm{E}[\Upsilon_{i,a}(\boldsymbol{\gamma}_i)] \tag{27}$$

$$\Upsilon_{i,a}(\boldsymbol{\gamma}_i) = u(i,\theta,a) 1_{\{\mathbf{y}_{i-1} \in \mathbf{C}_{i-1}(\boldsymbol{\gamma}_i), \mathbf{y}_i \in a_i(\boldsymbol{\gamma}_i)\}} \tag{28}$$

A pathwise simulation is not possible because $\nabla \upsilon_{i,a}(\boldsymbol{\gamma}_i) \neq \mathrm{E}[\nabla \Upsilon(\boldsymbol{\gamma}_i)] = 0$.

We may re–express (27) as nested integrals:

$$\upsilon_{i,a}(\boldsymbol{\gamma}_i) = \int_\Theta \int_{C_1(\tilde{\gamma}_1)} ... \int_{C_{i-1}(\tilde{\gamma}_{i-1})} \int_{a_i(\tilde{\gamma}_i)} u(i,\theta,a) p(\theta, \mathbf{y}_i) dy_i ... dy_1 d\theta \tag{29}$$

Taking $\partial/\partial\gamma_j$ of this expression may result in a new expression of nested integral form, with some of the y_k's in $p(\theta, \mathbf{y}_i)$ fixed. The integrand is an un–normalized density, $L^{\mathrm{un}}_{i,a,j}(.)$ under certain conditions, in particular that $u(i,\theta,a)$ is everywhere positive (or negative) on Θ:

$$\partial_j \upsilon_{i,a}(\boldsymbol{\gamma}_i) \triangleq \partial \upsilon_{i,a}(\boldsymbol{\gamma}_i)/\partial\gamma_j = \mathrm{E}_q[L^{\mathrm{un}}_{i,a,j}(\theta, \mathbf{y}_i)/q^{\mathrm{un}}(\theta)]/\mathrm{E}[1/q^{\mathrm{un}}(\theta)] \tag{30}$$

where $q(.)$ is an arbitrary density (normalized or not) such that $q \gg p$. In principle the expectations on the right hand side can be approximated by a simulation average. In the particular case $q^{\mathrm{un}}(.) = L^{\mathrm{un}}_{i,a,j}(.)$, an MCMC

procedure can be used. This approach would have to be repeated for all admissible combinations $\{i, a, j\}$. Although a given sample may be re–used across $\{i, a, j\}$'s, that would be costly in terms of importance weights calculation and finding a $q(.)$ which is satisfactory for all combinations is a challenge. Provided we can make this claim rigorous, the difficulty that remains is to find a sampler $q(.)$ which works well *across* paths.

A straightforward alternative is the finite difference method, under which (25) is termed the Kiefer–Wolfowitz algorithm. In contrast with the preceding method, we only need to simulate from $p(\theta, \mathbf{y}_i)$.

Fix b, so that $\tau = \tau(\gamma^{(b)})$ and let $\tau_k = \tau(\gamma + e_k h_k)$. A general expression for the finite difference gradient is

$$\nabla_{\mathrm{FD}} \upsilon = \mathrm{E}_q[(w \times \nabla_{\mathrm{FD}}) u(\theta, \mathrm{y}_\tau)] \tag{31}$$

where $q(.)$ is an abitrary distribution such that $q \gg p$, and the k^{th} element of the integrand is

$$(w \times \nabla_{\mathrm{FD}}) u(\theta, \mathbf{y}_\tau) = \frac{(w \times u)(\theta, \mathbf{y}_k) - (w \times u)(\theta, \mathbf{y})}{h_k} \tag{32}$$

with $w(\theta, \mathbf{y}_\tau) = p(\theta, \mathbf{y}_\tau)/q(\theta, \mathbf{y}_\tau)$, and $\times$ is the product operator for functions i.e. $(f \times g)(.) = f(.)g(.)$. The resulting expression for (31) will equal $\nabla \upsilon$ plus a residual term that converges to zero as $||h|| \to 0$ where $h = (h_1, ..., h_{\dim(\gamma)})$.

The procedure to estimate (31) is to sample m iid paths from $q(\theta, \mathbf{y}_{\max(\tau, \tau_1, ..., \tau_{\dim(\gamma)})})$ and evaluate the sample average $\widehat{\nabla}_{\mathrm{FD}} \upsilon = \frac{1}{m} \sum_{j=1}^{m} (w \times \nabla_{\mathrm{FD}} u)(\theta, \mathbf{y}_\tau)$. Each term in the difference in (32) is evaluated with the same path which reduces variance provided the two terms are positively correlated. At each iteration, b, a fresh sample of size m is generated and $\hat{\nabla}_{\mathrm{FD}}^{(b)} \upsilon$ is computed. In setting m we should keep in mind that averaging of the $\gamma^{(b)}$'s is already present, as implicit in (25).

We can weaken the above formulation to $w(\theta, \mathbf{y}_\tau) \propto p(\theta, \mathbf{y}_\tau)/q(\theta, \mathbf{y}_\tau)$ with an abitrary normalizing constant, which implies that we should rescale the weights to sum to one. This may be a matter of design[1] or necessity, in case either of $p(.)$ or $p(.)$ are known only up to a constant of proportionality. Unbiasedness of $\hat{\nabla}_{\mathrm{FD}} \upsilon$, relative to $\nabla_{\mathrm{FD}} \upsilon$ is only preserved in the limit as $n \to \infty$.

Bibliography

[1] Tim Hesterberg. "Weighted Average Importance Sampling and Defensive Mixture Distributions". In: *Technometrics* 37 (1995), pp. 185–194.

[2] Andrew Gelman et al. *Bayesian Data Analysis*. 2nd ed. Chapman & Hall/CRC, 2004. ISBN: 0-412-03991-5.

[3] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. New York, NY, USA: Springer, 2004. ISBN: 0-387-00451-3.