BIOMETRICS 59, 1–5
December 2006

DOI: 10.1111/j.1541-0420.2005.00454.x

BISTRO: Bayesian Importance Sampling for Phylogenetic Inference

B. Larget*
University of Wisconsin-Madison, WI 53706, USA
**email:* bret.larget@wisc.edu

and

C. Solís-Lemus*
University of Wisconsin-Madison, WI 53706, USA
**email:* solislemus@wisc.edu

SUMMARY: blabla

KEY WORDS: xxx; xxx; xxx; xxx.

1. Introduction

Phylogenetics studies the evolutionary relationships between species, typically visualized in a tree or phylogeny. DNA sequences are used as input data to estimate the phylogenetic tree that links species through common ancestry. This estimation can be performed through a variety of methods such as maximum parsimony (reference), maximum likelihood (reference) or bayesian inference (reference).

In the Bayesian framework, the goal is to obtain the posterior distribution of tree topologies, branch lengths and other model parameters. However, this posterior distribution does not have an explicit form. Furthermore, it is impossible to obtain samples directly from this distribution, so MCMC methods are widely used (references) to generate a Markov chain with the posterior distribution as stationary distribution.

The posterior sample generated by MCMC can then be used to do inference on parameters of interest, such as identifying trees or splits with higher posterior probabilities, or computing posterior means and credibility intervals of numerical model parameters.

The downside of MCMC methods is that its performance rapidly deteriorates as the parameter space increases due to slow or poor mixing. In phylogenetic analyses, the tree space increases dramatically with the number of species. Then, MCMC methods need a very big chain in order to navigate the huge tree space. In addition, the MCMC moves keep most of the parameters constant when proposing a new state, and thus the resulting chain is highly dependent. This situation could result in a decreased effective sample size as we need a chain with millions of generations to generate few independent observations.

With these limitations in mind, we propose an importance sampling method to generate independent samples from the posterior distribution in the phylogenetic setting. We introduce the program BISTRO (Bayesian Importance Sampling for TRees O...) for Bayesian phylogenetic inference through importance sampling. We compare the performance of MrBayes (reference) and BISTRO in a variety of simulated and real-life datasets, and conclude that this new sampling scheme is more efficient as proven by a bigger effective sample size (ESS). We show that the importance sampling methodology is adequate to estimate the posterior distribution of numerical parameters in a fixed topology, like branch lengths, base frequencies and rates, even in big datasets of up to 60 taxa. However, we found difficulties when incorporating topology uncertainy in datasets with as little as 25 taxa.

In section 2, we describe the methods applied to phylogenetics. In section 3, we apply BISTRO to different simulated and real-life datasets, and compared its performance to MrBayes in terms of ESS. In section 4, we discuss the difficulties to apply BISTRO to big datasets.

2. Importance Sampling for phylogenetics

Here we explain the methods of importance sampling applied to phylogenetics. For a review of importance sampling in general, see the appendix. Let $\theta = (T, t, \mathbf{Q}(\pi, r))$ be the parameters of interest in the phylogenetic setting, where T represents a tree topology, t represents the vector of branch lengths and $\mathbf{Q}(\pi, r)$ represents the rate matrix for the GTR model (reference) as a function of the base frequency vector $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$ and the transition rate parameters $s = (s_{AC}, s_{AG}, s_{AT}, s_{CG}, s_{CT}, s_{GT})$. Note that we have a different parametrization than MrBayes (add model parametrization here)

Let X denote the DNA sequences as input data. We want to generate independent samples from the posterior distribution $p(\theta|X)$. Thus, we need to find a density $g(\theta|X)$ such that we can generate samples from g instead of p. (explain importance sampling here).

The success of importance sampling relies on choosing a

density g that is close enough to the density of interest: centered in the same place (no bias) and with slightly heavier tails. We will focus in finding a g that ressembles the likelihood $L(X|\theta)$, instead of the posterior $p(\theta|X)$.

The importance sampling density $g(\theta|X)$ has three parts since θ has three parts: topology, branch lengths and rate matrix. We explain each part of g in the next subsections:

$$g(\theta|X) = g_1(\mathbf{Q})g_2(T|\mathbf{Q})g_3(t|T,\mathbf{Q}) \tag{1}$$

2.1 *Density for* $\mathbf{Q}$

The rate matrix $\mathbf{Q}$ is obtained from two components: a vector of base frequencies π and a vector of transition rates s which have commonly be assumed to follow a Dirichlet distribution: $\pi \sim Dirichlet(a_1, a_2, a_3, a_4)$ and $s \sim Dirichlet(b_1, b_2.b_3, b_4, b_5, b_6)$. We found, however, that it is not always the case that the Dirichlet distribution is a good fit for π and s, and we propose the use a new distribution: the symmetric generalized Dirichlet (how to put reference?).

2.1.1 *The Dirichlet distribution.*

The Dirichlet distribution is a very commonly used probability distribution on sets of positive random variables constrained to sum to one. The random variables $X_1, \ldots, X_k$ are said to have a Dirichlet distribution when they have the joint density

$$f(x_1, \ldots, x_k) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k x_i^{\alpha_i - 1}, \tag{2}$$

where $x_i > 0$ for all i and $\sum_{i=1}^k x_i = 1$, and the parameters $\alpha_i > 0$ for $i = 1, \ldots, k$.

Each random variable X_i has a marginal Beta$(\alpha_i, \eta - \alpha_i)$ distribution where $\eta = \sum_{i=1}^k \alpha_i$. It follows that X_i has mean $\mathsf{E}(X_i) = \alpha_i/\eta$ and variance $\mathsf{Var}(X_i) = \alpha_i(\eta - \alpha_i)/(\eta^2(\eta+1))$. To generate random variables $X_1, \ldots, X_k \sim$ Dirichlet$(\alpha_1, \ldots, \alpha_k)$, one simply generates independent random variables $Y_i \sim$ Gamma(α_1, λ) for $i = i, \ldots, k$ and any arbitrary $\lambda > 0$ (typically $\lambda = 1$) and letting $X_i = Y_i / \sum_{j=1}^k Y_j$.

A consequence is that when selecting the parameters $\{\alpha_i\}$ to match the marginal means, there remains only a single scale factor which determines all of the marginal variances. This suggests that by allowing the value of λ to vary with i that we may be able to create a distribution on positive random variables constrained to sum to one with the desired flexibility in the first and second moments.

2.1.2 *A Symmetric Generalized Dirichlet distribution.*

We define the symmetric generalized Dirichlet distribution (reference) on $X_1, \ldots, X_k$ to be the distribution of $(X_1, \ldots, X_k)$ where $X_i = Y_i / \sum_{j=1}^k Y_j$ for $i = 1, \ldots, k$ where the random variables $\{Y_i\}$ are mutually independent and $Y_i \sim$ Gamma(α_i, λ_i). As the distribution of the $\{X_i\}$ would be the same if all $\{Y_i\}$ were multiplied by a common constant, we add the constraint that $\sum_{i=1}^k \lambda_i = k$ so that the average values of the $\{\lambda_i\}$ parameters is one. (check if setting mean of $1/\lambda_i$ to be one is more convenient).

It is known (references) that the distribution of the sum $S = \sum_{i=1}^k Y_i$ may be written as an infinite mixture of Gamma densities. However, the joint density of $X_1, \ldots, X_k)$ has a closed form solution.

$$f(x_1, \ldots, x_k) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)\left(\prod_{i=1}^k \lambda_i^{\alpha_i}\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \times \frac{\prod_{i=1}^k x_i^{\alpha_i - 1}}{\left(\sum_{i=1}^k \lambda_i x_i\right)^{\sum_{i=1}^k \alpha_i}},$$

where $x_i > 0$ for all i and $\sum_{i=1}^k x_i = 1$

For the derivation, see reference.

I have not been able to derive closed form solutions for the marginal means and variances, but the means are close (if not exactly equal to) $(\alpha_i/\lambda_i) / \sum_{j=1}^k (\alpha_j/\lambda_j)$.

2.1.3 *Parameter Estimation.*

Suppose that a probability density g on the k-dimensional simplex has marginal mean $\{\mu_i\}$ and marginal variances $\{v_i\}$. We do the following.

$$\alpha_i = \frac{\mu_i^2(1-\mu_i)}{v_i}$$

$$\lambda_i = \frac{\mu_i(1-\mu_i)}{v_i} \Bigg/ \sum_{j=1}^k \frac{\mu_j(1-\mu_j)}{k v_j}$$

By construction, the mean of the $\{\lambda_i\}$ is one. I need to provide some more theoretical evidence that these parameter estimates work.

To choose the parameters of the symmetric generalized Dirichlets, we need to find unbiased estimates of π and s. This proved to be a difficult task. Estimating π by the observed base frequencies and s by the observed pairwise counts would yield biased estimates. Estimating Q for a fixed initial topology with branch lengths would also yield biased estimates. This is due to the fact that we need an estimate of Q *averaged* over different topologies and branch lengths. We provide a partial solution by using MCMC on the fixed Neighbor-Joining (NJ) tree (from Jukes-Cantor distances) to sample branch lengths and Q. We then use the sample of Q to provide estimates of center and variance for the symmetric generalized Dirichlet densities.

Parameters of the proposal density for $\mathbf{Q}$.

(1) Obtain the NJ tree from the JC distances
(2) Use MCMC on that fixed tree to obtain estimates of mean and variance for π: $(\hat{\pi}_1, \hat{\pi}_2, \hat{\pi}_3, \hat{\pi}_4)$ and s: $(\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \hat{s}_5, \hat{s}_6)$
(3) Compute the parameters of the proposal symmetric generalized Dirichlet densities as described in section 2.1.3

For the importance sampling scheme, we sample $\pi \sim$ *SymmetricGeneralizedDirichlet* and $s \sim$ *SymmetricGeneralizedDirichlet*, and then construct $\mathbf{Q}$ like this:

$$q_{ij} = \frac{s_{ij}}{2\pi_i} q_{ii} = -\sum_i q_{ij} \tag{3}$$

2.2 *Density for* T *given* $\mathbf{Q}$

The proposal density for tree topologies depend on the clade distribution (Larget, 2013). In order to estimate the clade distribution, we need a sample of tree topologies, which we obtain by a mixture of a bootstrap sample and an MCMC sample.

First, we bootstrap the sites and with the MCMC mean estimate of Q, we use the GTR distance matrix between

sequences to obtain a NJ tree. We repeat this procedure to obtain a bootstrap sample of NJ trees.

The bootstrap sample of trees has two problems: one, it is too spread out and therefore, it contains many clades that have too low posterior probability, and two, sometimes it lacks important clades with high posterior probability that never appear in the bootstrap sample.

To overcome these problems, we decide to weight trees according to its distance from a center tree. The procedure is like this: from the bootstrap sample, we compute a mean tree (reference? bret mean published somewhere?) and calculate the distance of each bootstrap tree to the mean tree. Each tree will then be assigned a weight of the form $\exp -\kappa d(T_{mean}, T)$, for a constant κ and for the distance $d(T_{mean}, T)$ defined as in (reference). In this way, trees that are close to the mean tree will have higher weight, and thus its clades will be sampled more often when using the clade distribution. This idea is based on the assumption that the mean tree is close to the MLE tree which has not been proven (but see for a slightly different mean tree reference Megan Owen).

In practice, we discovered that sometimes the mean tree is far from the MLE tree. To avoid a biased clade distribution, we decide to mix the bootstrap sample with an MCMC sample as described next. As suggested in reference of nonparsimonious paper, we used the parsimony score as a proxy of a integrated likelihood on tree topologies, and run a Markov chain to get a tree sample. We weight each tree in the sample with a weight of the form $\exp -3.3 parsimony$, where the constant 3.3 was suggested by visual inspection of reference of nonparsimoniuos paper.

In the end, we combine both the bootstrap sample and the MCMC sample, and use the combined sample to estimate the clade distribution.

Estimation of clade distribution.

(1) Obtain a sample of NJ bootstrap trees with GTR distances for the MCMC mean of $\mathbf{Q}$
(2) Compute the mean tree of the bootstrap sample as in (reference)
(3) Compute the distance of each bootstrap tree to the mean tree as in (reference), and weight each tree by $\exp -\lambda d(T_{mean}, T)$
(4) Obtain an MCMC sample using the parsimony score as a proxy of the integrated likelihood on tree topologies
(5) Weight each tree in the MCMC sample with $\exp -3.3 parsimony$
(6) Combine the two samples and use the resulting sample to estimate the clade distribution as in (Larget, 2013)

For the importance sampling scheme, we will sample one topology from the clade distribution as in (Larget, 2013) (add more details?).

2.3 *Density for t given $T, \mathbf{Q}$*

After drawing a random topology T, we initialize all the branch lengths with the NJ distances (and set 0.00001 as minimun length in case of negative distance), and then estimate the MLE distance for each branch at a time using the likelihood function

$$L(t) = \prod_k \sum_x \sum_y \pi_x P_{xy}(t) P(A_k|x) P(B_k|y) \qquad (4)$$

where the product is over sites, the two sums are over the state of the internal nodes x and y, $P_{xy}(t)$ is the transition probability from x to y in time length t given by the GTR model, and $P(A_k|x), P(B_k|y)$ are the probability of the subtrees given the state of the internal nodes.

When estimating the MLE for a given branch length, we keep all the other branch lengths in the subtrees constant at their current values. Since the NJ branch lengths are very different from the MLE branch lenghts, we need to do several MLE passes to get accurate branch length estimates.

After all branch lengths are set at their MLE, we traverse the tree in postorder from leaves to root. For a given cherry, we jointly sample the two branch lengths leading to leaves with a Gamma distribution centered at the joint MLE and with variance given by the observed 2×2 information matrix from the likelihood:

$$\begin{aligned} L(t_1, t_2) \quad &= \textstyle\prod_k \sum_y \pi_y \sum_{x_1} \sum_{x_2} P_{yx_1}(t_1) P_{yx_2}(t_2) * \\ &\quad P(A_k^{(1)}|x_1) P(A_k^{(2)}|x_2) P(B_k|y). \end{aligned}$$

The observed information matrix is negative the inverse of the Hessian matrix evaluated at the MLE.

This joint density allows us to account for the correlation of sister edges, which is an important component of the likelihood under certain situations (see section 4). Furthermore, the term $P(B_k|y)$ allows us to include all the data in the likelihood, not only the data in the subtrees of x_1 and x_2.

Joint Gamma sampling. The goal is to generate a joint Gamma random vector with mean $\mu = (\mu_1, \mu_2)$ and covariance matrix Σ

(1) Obtain the Cholesky decomposition $\Sigma = LL^T$, with L a lower triangular matrix
(2) Generate $T_1 \sim Gamma(\alpha_1, \lambda_1)$ with

$$\alpha_1 = \frac{\mu_1^2}{L_{11}^2}, \lambda_1 = \frac{\mu_1}{L_{11}^2} \qquad (5)$$

(3) Compute $z_1 = \frac{T_1 - \mu_1}{L_{11}}$
(4) Generate $T_2|T_1 \sim Gamma(\alpha_2, \lambda_2)$ with

$$\alpha_2 = \frac{(\mu_2 + L_{21} z_1)^2}{L_{22}^2}, \lambda_2 = \frac{(\mu_2 + L_{21} z_1)}{L_{22}^2} \qquad (6)$$

For the importance sampling scheme, we simply traverse the tree and sample the branch lengths of sister edges $(t_1, t_2) \sim JointGamma$.

2.3.1 *Truncated normal case.*

When the MLE of a branch length is close to zero, the likelihood function is not convex as a Gamma distribution. In these cases, we wish to find a density function proportional to the right tail of a normal distribution in order to match a likelihood function which is defined only on positive values and the log of the likelihood is very well approximated by a parabola. We will use a truncated normal at $x_0 = 0$ (see Figure 1). We wish to find values of μ and σ in order to match specified a specified mean and variance of the distribution F. We also require an expression for the inverse cdf of F for random number generation.

The density f has the following expression.

$$f(x) = \frac{1}{(1 - \Phi(z_0))\sigma\sqrt{2\pi}} \, \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \qquad x > x_0$$

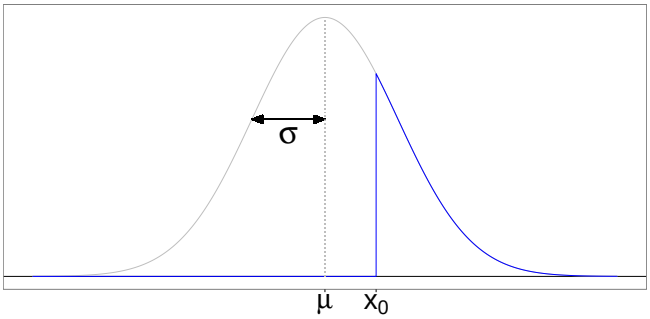

Figure 1. Truncated normal

where Φ is the cdf of the standard normal curve and $z_0 = (x_0 - \mu)/\sigma$.

The cdf of F has the following expression.

$$F(x) = \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi(z_0)}{1 - \Phi(z_0)}$$

Here is a derivation:

$$\begin{aligned} F(x) &= 1 - \int_x^\infty \frac{1}{(1-\Phi(z_0))\sigma\sqrt{2\pi}} \, \mathrm{e}^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \, \mathrm{d}t \\ &= 1 - \frac{1}{1-\Phi(z_0)} \int_{\frac{x-\mu}{\sigma}}^\infty \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{z^2}{2}} \, \mathrm{d}z \\ &= 1 - \frac{1 - \Phi(\frac{x-\mu}{\sigma})}{1-\Phi(z_0)} \\ &= \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi(z_0)}{1-\Phi(z_0)}, \qquad x > x_0 \end{aligned}$$

Using the inverse cdf method, we can generate a random variable with distribution F by the following equation where $U \sim \text{Uniform}(0,1)$.

$$X = \mu + \sigma\Phi^{-1}(1 - (1-U)(1-\Phi(z_0)))$$

The derivation is straightforward.

Finally, whenever $z_0 > 7$, we use an exponential distribution, instead of truncated normal because of numerical issues.

2.4 *Computation of the importance weights*

We computed the likelihood of the data given the drawn $\theta = (\mathbf{Q}, T, \lambda)$ and evaluated the importance sampling density $g(\theta|X)$ to obtain the weight:

$$w(\theta) = \frac{L(\theta|X)}{g(\theta|X)}. \tag{7}$$

Finally, we repeated the process to obtain an independent sample $\theta_1, \theta_2, ..., \theta_n \sim g(\theta|X)$ with the unnormalized weights $w(\theta_1), w(\theta_2), ..., w(\theta_n)$. Note that since the likelihood used is not a normalized density, we are in the case of self-normalizing importance sampling estimates, and thus, we need to normalize the weights:

$$\tilde{w}(\theta_j) = \frac{w(\theta_j)}{\sum_{i=1}^n w(\theta_i)}. \tag{8}$$

todo: - add here the whole algorithm of bistro - describe h functions of interest: indicator of tree

3. Examples

here we present examples

4. Discussion

problems

- correlation of branch lengths: if all three branches are equal size, then almost uncorrelated, but if one if long, the other two are correlated - dimensionality: at the end we did not see this problem much - clade distribution: if bistro mean and mb mean are far apart, our clade distribution will not be good (use mds plots to show) problems, limitations: two components: sample topolgy, and then sample BL. the second one is good, but the other one: there are cases when the shrinking of bootstrap towards the mean is good, in other not

Acknowledgements

The authors thank blabla.

Supplementary Materials

Web Appendix 1 referenced in Section xxx is available with this paper at the Biometrics website on Wiley Online Library.

References

Larget, B. (2013). The estimation of tree posterior probabilities using conditional clade probability distributions. *Systematic Biology* **62,** 501–511.

Owen, A. B. (2013). *Monte Carlo theory, methods and examples.*

Received October 2004. *Revised February* 2005.
Accepted March 2005.

Appendix

Importance Sampling Background

Let $X \sim f(x)$, and suppose that we want to calculate the expectation of a function $h(X)$ under $f(x)$:

$$E_f(h(X)) = \int h(x)f(x)dx := \mu \tag{A.1}$$

If the integral does not have a closed solution, we can estimate μ with the mean from a random sample $X_1, X_2, ..., X_n \sim f(x)$:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^n h(X_i) \tag{A.2}$$

Problem. It can be hard (or impossible) to get random samples from $f(x)$.

Solution. Sample from an easier density $g(x)$ such that inference of $h(X)$ under $f(x)$ can be approached as inference of $h(X)w(X)$ under $g(x)$ for a weight function $w(x) = f(x)/g(x)$ defined when both $f(x)$ and $g(x)$ are *normalized* densities (we will discuss the *unnormalized* case in next subsection). That is,

$$\begin{aligned} E_f(h(X)) &= \int h(x)f(x)dx \\ &= \int h(x)\tfrac{f(x)}{g(x)}g(x)dx \\ &= \int h(x)w(x)g(x)dx \\ &= E_g(h(X)w(X)) \end{aligned}$$

Importance Sampling Algorithm. The goal is to estimate $\mu = E_f(h(X))$, and $\sigma^2 = Var_f(h(X))$.

(1) Sample independently $Y_1, Y_2, ..., Y_m \sim g(x)$
(2) Define the weight function: $w(x) = f(x)/g(x)$
(3) Mean estimate: $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} h(y_i)w(y_i)$
(4) Variance estimate: $\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(h(y_i)w(y_i) - \hat{\mu})^2$

Standard error. Since $Y_1, Y_2, ..., Y_m \sim g(x)$ is an independent sample, we can compute the variance of the estimator as

$$Var_g(\hat{\mu}) = \frac{1}{m}Var_g(h(X)w(X)) = \frac{\sigma^2}{m} \tag{A.3}$$

Unnormalized case. The usual approach of importance sampling assumes that you have the normalized versions of both $f(x)$ and $g(x)$. In many real-life applications, this is not true.

Let $X \sim f(x) = c_1 f_0(x)$, and we again want to estimate $E_f(h(X))$. We sample independently $Y_1, Y_2, ..., Y_m \sim g(x) = c_2 g_0(x)$. Unlike in the previous setting, we compute the weights with the *unnormalized* densities: $w_0(y_i) = f_0(y_i)/g_0(y_i)$.

This unnormalized case is different from the normalized importance sampling case. Here we do inference of $h(X)$ directly, but with a weighted sample $Y_1, Y_2, ..., Y_m$ with weights $\tilde{w}_0(y_1), \tilde{w}_0(y_2), ..., \tilde{w}_0(y_m)$ with $\tilde{w}_0(y_j) = \frac{w_0(y_j)}{\sum_{i=1}^{m} w_0(y_i)}$. On the contrary, in the normalized importance sampling case, we do inference of $h(X)w(X)$ with a random sample from $g(x)$.

Unnormalized Importance Sampling Algorithm. The goal is to estimate $\mu = E_f(h(X))$, and $\sigma^2 = Var_f(h(X))$.

(1) Sample independently $Y_1, Y_2, ..., Y_m \sim g(x)$
(2) Define the weight function: $w_0(y_j) = f_0(y_j)/g_0(y_j)$, and the normalized weight as $\tilde{w}_0(y_j) = \frac{w_0(y_j)}{\sum_{i=1}^{m} w_0(y_i)}$.
(3) Mean estimate: $\hat{\mu} = \sum_{i=1}^{m} h(y_i)\tilde{w}_0(y_i)$
(4) Variance estimate: $\hat{\sigma}^2 = \sum_{i=1}^{m}(h(y_i) - \hat{\mu})^2\tilde{w}_0(y_i)$

The estimate $\hat{\mu}$ is sometimes called *self-normalized importance sampling estimate.*

Standard error. Since the sample $Y_1, Y_2, ..., Y_m$ with weights $\tilde{w}_0(y_1), \tilde{w}_0(y_2), ..., \tilde{w}_0(y_m)$ is no longer an independent sample (because weights add up to 1), the variance estimate is then (Owen, 2013)

$$\widehat{Var}(\hat{\mu}) = \sum_{i=1}^{m} \tilde{w}_0(y_i)^2(h(y_i) - \hat{\mu})^2. \tag{A.4}$$

Diagnostics. Importance sampling diagnostics are not clear-cut rules. One common approach is to compute the effective sample size:

$$n_e = \frac{\left(\sum_{i=1}^{n} w(y_i)\right)^2}{\sum_{i=1}^{n} w(y_i)^2}. \tag{A.5}$$

If the effective sample size is too small, then one or few weights could be too large compared to the others, and importance sampling is not as efficient as expected.