BISTRO: Bayesian Importance Sampling...

BL,CSL

I. Importance Sampling Background

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

$$E_f(h(X)) = \int h(x)f(x)dx := \mu$$

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

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

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,

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

Algorithm 1: Importance Sampling

Goal: 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

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

Unnormalized case

The usual approach of importance sampling assumes that you have the normalized version of both $f(x)$ and $g(x)$. In many real-life applictions, 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 $g(x)$.

Algorithm 2: Unnormalized Importance Sampling

Goal: 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 (**?**)

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

Diagnostics

Importance sampling diagnostics are not clear-cut rules. One common approach to detect if importance sampling performed well is to compute the effective sample size:

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

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

II. Importance Sampling for Phylogenetics

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 several species. 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, we want to obtain the posterior distribution of trees, 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, which in turns could result in a decreased effective sample size (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. This new sampling scheme is more efficient as proven by a bigger effective sample size.

III. Methods

Let $\theta = (T, \lambda, \mathbf{Q}(\pi, r))$ be the parameters of interest in the phylogenetic setting, where T represents a tree topology, λ 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 $r = (r_{AC}, r_{AG}, r_{AT}, r_{CG}, r_{CT}, r_{GT})$. (add model parametrization here) Let X denote the DNA sequences as input data.

We want to generate independent samples from the posterior distribution $p(\theta|X)$ (the $f(x)$ density in section **??**). Thus, we need to find a density $g(\theta|X)$ such that we can generate samples from g instead of p.

The success of importance sampling relies on choosing a g that is close enough to the density of interest, and that it has heavier tails. Since the posterior distribution depends on the choice of the prior, we will focus in finding a g that resembles the likelihood $L(X|\theta)$, in an attempt to use the same density g regardless of the chosen prior.

The importance sampling density $g(\theta|X)$ has multiple parts which will be explained in the next subsections:

$$g(\theta|X) = f_1(\mathbf{Q}) f_2(T|\mathbf{Q}) f_3(\lambda|T, \mathbf{Q})$$

Density for $\mathbf{Q}$

First, we generate a rate matrix $\mathbf{Q}$ by ... (do we do this?).

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

After drawing a rate matrix $\mathbf{Q}$ from $f_1(\mathbf{Q})$, we compute the distance matrix between sequences, and then a Neighbor-Joining (NJ) tree (reference). We then bootstrap the sites to get a bootstrap sample of NJ trees. We then use this sample of NJ trees to compute the clade distributions (**?**), and use these clade probabilities to draw a random topology. (need to double check this, add more details).

Density for λ *given* $T, \mathbf{Q}$

After drawing a random topology T, we initialize all the branch lengths with the NJ distances (and 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

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

where the product is over sites, the two sum 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. (add figure)

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 order the nodes in postorder, and traverse the tree 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:

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

The observed information matrix is negative the inverse of the Hessian matrix evaluated at the MLE. (add figure)

Algorithm 3: Joint Gamma sampling

Goal: 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
$$\alpha_1 = \frac{\mu_1^2}{L_{11}^2} \qquad \lambda_1 = \frac{\mu_1}{L_{11}^2}$$
3) Compute $z_1 = \frac{T_1 - \mu_1}{L_{11}}$
4) Generate $T_2|T_1 \sim Gamma(\alpha_2, \lambda_2)$ with
$$\alpha_2 = \frac{(\mu_2 + L_{21} z_1)^2}{L_{22}^2} \qquad \lambda_2 = \frac{(\mu_2 + L_{21} z_1)}{L_{22}^2}$$

Computation of the 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:

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

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:

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

todo: - describe h functions of interest: indicator of tree - go back and fill details missing, plots, formulas