

Fisher Matrix Bootstrap Algorithm Description

Your Name

March 27, 2024

1 Introduction

The Fisher Information Matrix (FIM) is a symmetric positive definite matrix that quantifies the amount of information that a set of observations carries about the parameters of a statistical model. It is denoted by $I(\theta)$, where θ represents the vector of parameters.

Mathematically, $I_{ij} = -E[\frac{\partial^2 \ln f_{\theta_0}(X)}{\partial \theta_i \partial \theta_j}]$, where f is the likelihood function.

It can be seen that the ij^{th} entry of the FIM is $I_{ij} = cov[\frac{\partial \ln f_{\theta_0}(X)}{\partial \theta_i}, \frac{\partial \ln f_{\theta_0}(X)}{\partial \theta_j}]$. Therefore, the FIM is the covariance matrix between the score functions (derivatives of the log-likelihood function with respect to the parameters θ_i and θ_j). As a covariance matrix, the eigenvalues and eigenvectors of FIM then indicate the principal components and the principal directions of the variances. The principal directions thus give us the θ^* that represents high value data the model should focus on. On contrary, the inverse of the Fisher matrix, denoted as $I^{-1}(\theta) = cov[\hat{\theta}_i, \hat{\theta}_j]$, provides estimates of the variances and covariances of the parameters.

In applied statistics, the bootstrap is a tool that allows inference by assessing the variability of the statistics of interest directly from the data without relying on asymptotic theory. In situations where asymptotic theoretical derivations are difficult and/or can only be done under quite restrictive assumptions, the bootstrap is particularly appealing. In the context of bootstrapping with Fisher matrices, it's crucial to employ techniques that preserve their inherent characteristics, particularly positive definiteness and symmetry, during resampling. For the specific scenario where we have two Fisher matrices, A and B , corresponding to identical parameters, bootstrapping can be achieved by manipulating the eigenvalues and eigenvectors. These manipulations involve utilizing the original matrices, A and B , as sources for the modifications.

Directions in parameter space where the log-likelihood function varies the most are represented by the eigenvectors of the FIM. Each eigenvector represents a direction with the greatest variance in the parameter space. The corresponding eigenvalues indicate the amount of variance along each eigenvector direction. Eigenvectors associated with larger eigenvalues represent directions where the parameters are estimated with greater precision. In other words, larger eigenvalues correspond to smaller variances, indicating higher precision

in estimating the corresponding parameters. Conversely, smaller eigenvalues imply larger variances and lower precision. Eigenvalues close to zero suggest that the data provides weak information for estimating the associated parameters, leading to high uncertainty (high σ^2). This implies that imposing strong constraints on these parameters might be challenging.

Considering two provided Fisher information matrices, we can construct a hypothetical scenario where the precision of the estimated parameters exhibits variations that adhere to the characteristics observed in the two given cases. By swapping the eigenvectors and eigenvalues of two observed Fisher matrixes, we can alter precision and cause variation.

2 Algorithm Description

1. Eigenvalue Decomposition:

Perform eigenvalue decomposition on both matrices A and B. This will give: For A: Eigenvalues $\Lambda_A = (\lambda_A^1, \lambda_A^2, \dots, \lambda_A^n)$ and corresponding eigenvectors $V_A = (v_A^1, v_A^2, \dots, v_A^n)$. For B: Eigenvalues $\Lambda_B = (\lambda_B^1, \lambda_B^2, \dots, \lambda_B^n)$ and corresponding eigenvectors $V_B = (v_B^1, v_B^2, \dots, v_B^n)$.

2. Swapping masks generation:

To define the order in which eigenvalues will be swapped, we create a mask vector \mathbf{m} of size n (where n is the number of eigenvalues). When n is small, these mask vectors can be efficiently generated by converting a sequence of integers from 0 to $2^n - 1$ into binary masks.

3. Swap Eigenvalues and Eigenvectors:

Loop through the mask vector \mathbf{m} :

For each index i in \mathbf{m} : Select the eigenvalue: $\lambda_C^i = \begin{cases} \lambda_A^i & m(i) = 0 \\ \lambda_B^i & m(i) = 1 \end{cases}$
 (where $m(i)$ is the value at index i in the permutation vector). Select the corresponding eigenvectors: $v_C^i = \begin{cases} v_A^i & m(i) = 0 \\ v_B^i & m(i) = 1 \end{cases}$.

4. Orthogonalization (Optional):

Given a new positively definite symmetric matrix, C , its eigenvalues and eigenvectors satisfy the property $C = V_C \Lambda_C V_C^T$, where V_C is an orthogonal matrix ($V_C V_C^T = E$) and Λ_C is a diagonal matrix containing the eigenvalues of C . When manipulating eigenvalues and eigenvectors during bootstrapping, the resulting eigenvector combination (V_C) typically loses its orthogonality. While orthogonalization techniques can be applied to restore orthogonality, this process might alter certain eigenvalues.

In the context of bootstrapped matrices (V_C), QR decomposition allows us to express C as $C = Q_C R_C \Lambda_C R_C^T Q_C^T$, where Q_C is orthogonal and R_C

is upper triangular. Notably, the central term in this formula, $R_C \Lambda_C R_C^T$ (denoted as $\hat{\Lambda}_C$), can be interpreted as a new diagonal matrix containing biased eigenvalues. Therefore, the primary motivation for employing orthogonalization techniques during bootstrapping is to minimize alterations to the original eigenvalues. To achieve orthogonalization, we can remove the upper triangular matrix R_C . This simplifies the equation to $Q_C \Lambda_C Q_C^T$, where Q_C now represents a set of orthonormal eigenvectors corresponding to the potentially biased eigenvalues in $\hat{\Lambda}_C$ but applied to the original Λ_C .

5. Reconstruction:

Following the manipulation of eigenvalues and eigenvectors, the modified matrix (denoted as C) can be reconstructed in two ways:

(a) Orthogonalized Version:

For an orthonormal set of eigenvectors, the reconstruction takes the form: $Q_C \Lambda_C Q_C^T$, where Q_C represents the obtained orthonormal eigenvectors and Λ_C is the diagonal matrix containing the original eigenvalues.

(b) Original Form: If the original structure with potentially non-orthogonal eigenvectors is preferred, the reconstruction can be expressed as: $C = V_C \Lambda_C V_C^T$, where V_C denotes the manipulated eigenvectors and Λ_C remains the diagonal matrix of biased eigenvalues.

6. Aggregation:

The generated bootstrap replicates, denoted as $C_0, C_1, \dots, C_{2^n-1}$ serve as the basis for aggregation. A straightforward approach involves calculating the mean of these replicates $\hat{C} = \frac{1}{2^n} \sum_{i=0}^{2^n-1} C_i$. However, the statistical properties of this aggregated statistic, particularly its bias and variance, warrant further research to ensure its reliability for inference.

3 Discussion

One of the challenges in implementing the matrix bootstrap algorithm is to ensure that the resulting matrices retain their essential properties. This includes preserving symmetry, positive definiteness, and orthonormal basis of eigenvectors. The algorithm addresses these challenges by selecting eigenvectors from the source matrices and optionally applying an orthonormalization routine.