Cross-Entropy Based Importance Sampling for Stochastic Simulation Models

Quoc Dung Cao\textsuperscript{a}, Youngjun Choe\textsuperscript{a,*}

\textsuperscript{a}Department of Industrial and Systems Engineering, University of Washington, Seattle, WA 98195, USA

Abstract

To efficiently evaluate system reliability based on Monte Carlo simulation, importance sampling is used widely. The optimal importance sampling density was derived in 1950s for the deterministic simulation model, which maps an input to an output deterministically, and is approximated in practice using various methods. For the stochastic simulation model whose output is random given an input, the optimal importance sampling density was derived only recently. In the existing literature, metamodel-based approaches have been used to approximate this optimal density. However, building a satisfactory metamodel is often difficult or time-consuming in practice. This paper proposes a cross-entropy based method, which is automatic and does not require specific domain knowledge. The proposed method uses an expectation–maximization algorithm to guide the choice of a mixture distribution model for approximating the optimal density. The method iteratively updates the approximated density to minimize its estimated discrepancy, measured by estimated cross-entropy, from the optimal density. The mixture model’s complexity is controlled using the cross-entropy information criterion. The method is empirically validated using a numerical study and applied to a case study of evaluating the reliability of wind turbine using a stochastic simulation model.

Keywords: Monte Carlo, variance reduction, mixture model, cross-entropy

\textsuperscript{*}Corresponding author

Email address: ychoe@uw.edu (Youngjun Choe)
1. Introduction

There are two types of computer simulation models, deterministic simulation models and stochastic simulation models. In a deterministic simulation model, the simulator will produce a fixed output for the same input values. In this paper, we focus on stochastic simulation models. Due to extra stochastic elements in the simulation model, given a fixed input value, the output is a random variable representing the outcome of the simulation. Despite its flexible modeling capacity, stochastic simulation models can be more complex and computationally costly to generate a simulation outcome.

One of the applications of Monte Carlo simulation is to compute an estimator of a system’s expected output. In Monte Carlo simulation, one draws inputs from a suitable probability distribution, calculates the output, and repeats the process multiple times to obtain different outputs. The estimate of the expected output can be the average of those simulated outputs. Monte Carlo simulation is widely used with both deterministic simulation models and stochastic simulation models. As the computing power advances over the past decades, the reliance on Monte Carlo simulation has increased in various engineering disciplines. Particularly in reliability engineering, Monte Carlo simulation is widely used to study the behavior and reliability of a system, often through estimating its failure probability.

However, the reliability evaluation based on Monte Carlo simulation remains challenging. The more closely the simulation model mimics the real system, the more computationally expensive each simulation run is. Moreover, highly reliable systems such as wind turbines or nuclear reactors require many simulation replications to encounter a rare event of interest such as a system failure, which translates to further demand for computational resources. These challenges are exacerbated with stochastic simulation models, which often have to be run multiple times at the same input value. These challenges highlight the importance
of improving computational efficiency of the reliability study with stochastic simulation models.

Among techniques to speed up Monte Carlo simulation in rare event probability estimation, importance sampling (IS) is one of the most promising methods [1], especially with deterministic simulation models [2]. IS can boost the simulation efficiency by reducing the number of required simulation runs to achieve a target variance of failure probability estimator [3]. IS methods are also used with stochastic simulation models in various applications, such as finance [4], insurance [5], reliability [6, 7], communication networks [8], and queueing operations [9, 10, 11].

Theoretically, if the inputs are drawn from the optimal IS density, the variance of the probability estimator will be minimized and we can save the most computational cost. However, the theoretically optimal IS density is generally not implementable in practice and often necessitates some approximations such as a metamodel-based method [12] or the cross-entropy (CE) method [13]. Metamodel-based IS aims to represent the original high fidelity but black-box simulation model by a surrogate model of which outputs can be more efficiently computed. From the surrogate model, the optimal IS density can be approximated to reduce the variance of the failure probability estimator [14]. From another perspective, CE-based IS aims to find an IS density that is as close as possible to the theoretically optimal IS density, where closeness is measured by the Kullback-Leibler divergence [13]. The CE method has been widely used for estimating the optimal IS density with deterministic simulation models. However, to the best of our knowledge, there is no literature exploring the CE method’s application to stochastic simulation models. Thus, this paper proposes a novel method called the cross-entropy based stochastic importance sampling (CE-SIS) to approximate the optimal IS density for stochastic simulation models.

One challenge of the CE-based IS is the dilemma between parametric and nonparametric representation of the candidate optimal IS density. In the standard CE method for deterministic simulation models, the candidate IS density is confined to a parametric family. Some of the popular parametric IS densities are
multivariate Bernoulli and multivariate Gaussian, thanks to their simple random variate generation and convenient updating formulae for CE minimization. However, parametric IS can be too rigid to capture the complicated important region and it is often difficult to validate the parametric model assumptions [15]. Nonparametric approach can offer flexibility to overcome such limitations. One of the most popular nonparametric density approximation methods is the kernel approach. Besides its flexibility, the kernel approach also has its own drawback. The probability density model is not as parsimonious as (a mixture of) parametric density models. This lack of parsimony often makes subsequent inference and analysis of the resulting model computationally intensive [16, 17].

For stochastic simulation models, the parametric IS approach needs a strong assumption but the variance of the IS estimator converges to the optimal variance faster than the nonparametric IS approach [18]. The nonparametric IS approach requires weak assumptions but the variance reduction rate is not as fast as the parametric IS approach [18]. To achieve a good balance between flexibility and efficiency in the CE-SIS method, we express the candidate IS density for stochastic simulation models using a mixture of parametric distributions. We particularly focus our study on the Gaussian mixture model due to its flexibility to model a smooth distribution.

The proposed method is validated with a numerical example and a case study on wind turbine reliability evaluation. In the case study, our method is benchmarked against a state-of-the-art metamodel-based IS method in the literature. Approximating the optimal IS density automatically without relying on expert domain knowledge (which is often necessary for metamodel construction), the CE-SIS method demonstrates comparable results. This shows the advantage of the CE-SIS method in situations when building a high-quality metamodel is difficult or time-consuming; the CE-SIS method would conveniently provide substantial computational saving potentially more than metamodel-based methods.
2. Background

The crude Monte Carlo (CMC) method samples the input, $\mathbf{X} \in \mathbb{R}^p$, from a known probability density function, $f(x)$ [1]. From the input $\mathbf{X}$, a simulator generates the output, $Y \in \mathbb{R}$. If the simulation model is deterministic, $Y$ is a deterministic function of $\mathbf{X}$, i.e., $Y = g(\mathbf{X})$. For the stochastic simulation model, $Y$ is stochastic for a given $\mathbf{X}$. Due to the random vector $\epsilon$ within the simulation model, it generates different outputs $Y$ even at a fixed input $\mathbf{X}$.

In reliability engineering, a quantity of interest is the so-called failure probability, $P(Y > l)$, where $l$ is a pre-specified threshold on a system that fails if $Y$ exceeds $l$. Note that any failure event set can be expressed as \{\(Y > l\}\} using a transformation. To estimate the failure probability, the following CMC estimator is most commonly used,

$$\hat{P}_{CMC} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(Y > l), \quad (1)$$

where $n$ is the number of total simulation replications. The estimator in (1) is an unbiased estimator for $P(Y > l)$. For the deterministic simulation model, the failure probability is equivalent to $E_f[\mathbb{I}(g(\mathbf{X}) > l)]$, where $\mathbb{I}()$ is the indicator function and the subscript, $f$, appended to the expectation operator, $E$, denotes that the expectation is taken with respect to $f$, the density from which $\mathbf{X}$ is drawn. For the stochastic simulation model, the failure probability is the expectation of the conditional failure probability, expressed as $E_f[P(Y > l | \mathbf{X})]$.

In a highly reliable system, the failure probability can be very low. It may take many simulation runs until a failure occurs. In some cases, each simulation run can be computationally very expensive. In order to obtain a good estimator of the failure probability, the number of simulation runs can be large. To save the computational resource, IS changes the sampling distribution of $\mathbf{X}$ from $f(x)$ to another distribution $q(x)$ that makes the failure events more likely. Sampling $\mathbf{X}$ from $q(x)$ makes the estimator in (1) no longer unbiased.

For the deterministic simulation model, to make the failure probability esti-
mator unbiased, the IS estimator becomes

\[
\hat{P}_{DIS} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(Y_i > l) \frac{f(X_i)}{q(X_i)},
\]  

(2)

where \(X_i, \ i = 1, \ldots, n,\) is sampled from \(q(x)\) instead of \(f(x)\) [1]. \(Y_i\) is the output from the simulator corresponding to the input \(X_i\). The variance of \(\hat{P}_{DIS}\) is minimized if \(X\) is sampled from the optimal IS density [2]

\[
q_{DIS}(x) = \frac{\mathbb{I}(g(x) > l) f(x)}{\mathbb{P}(Y > l)}.
\]  

(3)

Here, the indicator function \(\mathbb{I}(g(x) > l)\) can be evaluated only by running the simulator since the function \(g(x)\) is unknown. The denominator \(\mathbb{P}(Y > l)\) is the unknown quantity that we want to estimate. Thus, \(q_{DIS}\) is not implementable in practice, and approximation of \(q_{DIS}\) is required. As mentioned above, for the deterministic simulation model, the optimal IS density can be approximated using methods such as a metamodel-based method or the CE method. Hereafter, we call the IS method for a deterministic simulation model \(DIS\).

For the stochastic simulation model, to capture the extra randomness, the unbiased IS estimator becomes

\[
\hat{P}_{SIS} = \frac{1}{m} \sum_{i=1}^{m} \left( \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbb{I}(Y_j^{(i)} > l) \right) \frac{f(X_i)}{q(X_i)},
\]  

(4)

where \(m\) is the number of distinct \(X_i\) values and \(Y_j^{(i)}\) is the \(j\)th output from \(N_i\) replications at the distinct input, \(X_i\). Such \(N_i\) replications at the same \(X_i\) value capture the additional randomness \(\epsilon\) within the simulation model. Hereafter, we call the IS method for a stochastic simulation model \(SIS\).

Similar to \(DIS\), the variance of \(\hat{P}_{SIS}\) is desired to be as small as possible. Minimizing the variance requires two steps [12]:

(a) sampling \(X\) from

\[
q_{SIS}(x) = \frac{1}{C_q} f(x) \sqrt{\frac{1}{n} s(x) (1 - s(x)) + s(x)^2},
\]  

(5)

where \(s(x)\) is \(\mathbb{P}(Y > l \mid X = x)\) and \(C_q\) is the normalizing constant;
(b) allocating replications to \( X_i \) by

\[
N_i = n \frac{\sqrt{\frac{n(1-s(X_i))}{1+(n-1)s(X_i)}}}{\sum_{j=1}^{m} \sqrt{\frac{n(1-s(X_j))}{1+(n-1)s(X_j)}}}, \quad i = 1, \ldots, m.
\]  

Since the conditional probability, \( s(x) \), is unknown in practice, the IS method for a stochastic simulation model requires the approximation of \( s(x) \), \( q_{SIS} \), and \( N_i \). This paper will focus on how the CE method can be extended to approximate \( q_{SIS} \). In the next subsection, we first review how the CE method is applied to approximate \( q_{DIS} \).

2.1. CE Method for DIS

The CE method is originally developed to find the density that best approximates the optimal density of DIS [13]. The standard parametric CE method limits the search space for the optimal IS density \( q^*(x) \) to a pre-specified parametric family (e.g., Gaussian, Poisson, gamma, etc.), \( \{q(x; \theta) : \theta \in \Theta(d) \subset \mathbb{R}^d\} \), and seeks the density \( q(x; \theta^*) \) that is closest to the optimal density \( q^*(x) \). The closeness is measured by the Kullback-Leibler divergence [13],

\[
D(q^*(x), q(x; \theta)) = \int q^*(x) \ln q^*(x) \, dx - \int q^*(x) \ln q(x; \theta) \, dx.
\]  

This quantity is always non-negative and takes zero if and only if \( q^*(x) = q(x; \theta^*) \) almost everywhere. Thus, minimizing \( D(q^*(x), q(x; \theta)) \) over \( \theta \in \Theta \) leads to \( q(x; \theta^*) = q^*(x) \) if \( q^* \) belongs to the same parametric family as \( q(x; \theta^*) \).

Minimizing \( D(q^*(x), q(x; \theta)) \) in (7) over \( \theta \) is equivalent to minimizing its second term, known as the cross-entropy (CE)

\[
C(q^*(x), q(x; \theta)) = -\int q^*(x) \log q(x; \theta) \, dx,
\]  

because the first term in (7) is a constant over \( \theta \). As shown in (3), the optimal IS density can be expressed as \( q^*(x) \propto h(x)f(x) \), where \( h(x) \) is \( \mathbb{I}(g(x) > l) \) for
DIS. The CE method aims to equivalently minimize

$$\mathcal{C}(\theta) = - \int h(x)f(x) \log q(x; \theta) \, dx$$

$$= -E_f[h(X) \log q(X; \theta)]$$

$$= - \int h(x) \frac{f(x)}{q(x; \theta')} \log q(x; \theta)q(x; \theta') \, dx \quad (9)$$

$$= - \int h(x)w(x; \theta') \log q(x; \theta)q(x; \theta') \, dx$$

$$= -E_q[h(X)w(X; \theta') \log q(X; \theta)], \quad (10)$$

where $\theta, \theta' \in \Theta(d)$, and $f(x)$ is the original density of $X$. The likelihood ratio $f(x)/q(x; \theta')$ is denoted by $w(x; \theta')$. Note that the equality in (9) holds under the condition that $q(x; \theta') = 0$ implies $h(x)f(x) = 0$. This condition can be easily satisfied by ensuring that the support of $q$ includes the support of $f$ even if $h$ is unknown. The condition is always satisfied by the Gaussian mixture model.

In practice, the CE method finds $\hat{\theta}$ that minimizes the following IS estimator of (10),

$$\bar{\mathcal{C}}(\theta) = - \frac{1}{n} \sum_{i=1}^{n} h(X_i)w(X_i) \log q(X_i; \theta), \quad (11)$$

where $X_i, i = 1, \ldots, n$, is sampled from $q(x; \hat{\theta}')$. Note that by writing $w(X_i)$, we suppress the notation for dependence of $w$ on $\hat{\theta}'$ in (11). Using this IS estimator, the CE method minimizes the CE iteratively:

Step 1. Sample $X_i, i = 1, \ldots, n$, from $q(x; \hat{\theta}')$. At the first iteration, $q(x; \hat{\theta}')$ can be flexible (e.g., $f$ is commonly used).

Step 2. Find $\hat{\theta} = \arg\min_{\theta \in \Theta} \bar{\mathcal{C}}(\theta)$, where $\bar{\mathcal{C}}(\theta)$ is in (11).

Step 3. Set $\hat{\theta}' = \hat{\theta}$ and start the next iteration from Step 1 until some stopping criterion is met.

This procedure iteratively refines $q(x; \hat{\theta})$. However, the refinement is limited, as the parameter search space, $\Theta$, is restricted by a pre-defined parametric family.
Some studies [16, 17] explore nonparametric approaches to allow greater flexibility on the candidate IS density than the standard CE method. However, as mentioned before, the flexibility comes with costs: finding the optimal density [17] or sampling from the optimized density [16] is computationally challenging. Bridging between the two extremes of the spectrum (a parametric density with \( d \ll n \) or a nonparametric density with \( d \approx n \), where \( d \) is the number of parameters in a candidate IS density and \( n \) is the number of simulation replications), a few studies [15, 19, 20] recently consider the mixture of parametric distributions, where \( d \) can vary between 1 and \( n \). This approach is particularly desirable for engineering applications because (a) it can be as flexible as we want; (b) it is easy and fast to sample from the mixture of parametric distributions; and (c) the mixture IS density provides an insight into the engineering system (e.g., means of mixture components often coincide with the so-called ‘hot spots’, where the system likely fails.). Particularly in [21], the candidate distribution is expressed by the Gaussian mixture model (GMM) and an expectation–maximization (EM) algorithm is used to minimize the cross-entropy estimator in (11).

2.2. Gaussian Mixture Model and EM Algorithm

The GMM of \( k \) mixture components takes the following form:

\[
q(x; \theta) = \sum_{j=1}^{k} \alpha_j q_j(x; \mu_j, \Sigma_j),
\]

(12)

where the component weights, \( \alpha_j \), \( j = 1, \ldots, k \), are positive and sum to one. The \( j \)th Gaussian component density, \( q_j \), is specified by the mean, \( \mu_j \), and the covariance \( \Sigma_j \). Thus, the parameter vector of GMM \( \theta \) denotes \( (\alpha_1, \ldots, \alpha_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k) \). Since \( \theta \) is a function of \( k \), the GMM model can be explicitly written as \( q(x; \theta(k)) \).

For simplicity, we will write the GMM as \( q(x; \theta) \) unless we should express different models in terms of \( k \).

To minimize (11), the gradient of (11) with respect to \( \theta \) is set to zero:

\[
-\frac{1}{n} \sum_{i=1}^{n} h(X_i)w(X_i) \nabla_{\theta} \log q(X_i; \theta) = 0.
\]

(13)
This leads to the updating equations as derived in [20]:

\[
\alpha_j = \frac{\sum_{i=1}^{n} h(X_i)w(X_i)\gamma_{ij}}{\sum_{i=1}^{n} h(X_i)w(X_i)}, \tag{14}
\]

\[
\mu_j = \frac{\sum_{i=1}^{n} h(X_i)w(X_i)\gamma_{ij}X_i}{\sum_{i=1}^{n} h(X_i)w(X_i)\gamma_{ij}}, \tag{15}
\]

\[
\Sigma_j = \frac{\sum_{i=1}^{n} h(X_i)w(X_i)\gamma_{ij}(X_i - \mu_j)(X_i - \mu_j)^T}{\sum_{i=1}^{n} h(X_i)w(X_i)\gamma_{ij}}, \tag{16}
\]

where

\[
\gamma_{ij} = \frac{\alpha_j q_j(X_i; \mu_j, \Sigma_j)}{\sum_{j'=1}^{k} \alpha_{j'} q_{j'}(X_i; \mu_{j'}, \Sigma_{j'})}. \tag{17}
\]

As the name suggests, the right-hand sides of the ‘updating’ equations (14), (15), (16) involve \(\theta = (\alpha_1, \ldots, \alpha_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k)\) either explicitly or implicitly through \(\gamma_{ij}\). As such, the updating equations are interlocking with each other and cannot be solved analytically. Thus, by starting with an initial value for \(\theta\) on the right-hand sides of the updating equations, the left-hand sides are computed and plugged back to the right-hand sides iteratively until the convergence is reached. This optimization procedure is a version of the EM algorithm that alternates between the expectation step (computing \(\gamma_{ij}\)) and the maximization step (updating \(\theta\)).

A common challenge in using GMM to approximate the IS density is balancing between the estimated model’s KL divergence (a closeness measure between the approximated IS density and the optimal one) and the model complexity. The studies [15, 19, 20] that consider mixture models point out the difficulty associated with the choice of the number of mixture components, \(k\). They either assume that \(k\) is given [15, 20] or follow a rule of thumb based on “some understanding of the structure of the problem at hand” [19]. Overcoming the need of picking \(k\) a priori when using GMM to approximate the optimal DIS density, Choe [21] proposes the cross-entropy information criterion (CIC) to select \(k\) automatically.

2.3. Cross-Entropy Information Criterion
In order to balance between the cross-entropy estimate and the model complexity, the CE estimator is minimized and the model complexity $k$ is concurrently penalized. CIC for deterministic simulation models takes the following form:

$$
CIC_{DIS}^{(t)}(d) = \hat{C}_{DIS}^{(t-1)}(\hat{\theta}) + \hat{K}_{DIS}^{(t-1)} \frac{d}{\sum_{s=0}^{t-1} m(s)},
$$

(18)

where the CE estimator $\hat{C}_{DIS}^{(t-1)}(\hat{\theta})$ and $\hat{K}_{DIS}^{(t-1)}$ are calculated using all the aggregated data up to iteration $(t - 1)$:

$$
\hat{C}_{DIS}^{(t-1)}(\hat{\theta}) = -\frac{1}{\sum_{s=0}^{t-1} m(s)} \sum_{s=0}^{t-1} \sum_{i=1}^{m(s)} h(X_i) w(X_i; \hat{\theta}^{(s)}) \log q(X_i; \hat{\theta})
$$

(19)

$$
\hat{K}_{DIS}^{(t-1)} = \frac{1}{\sum_{s=0}^{t-1} m(s)} \sum_{s=0}^{t-1} \sum_{i=1}^{m(s)} h(X_i) w(x_i; \hat{\theta}^{(s)}).
$$

(20)

Note that $d = (k-1)+k(p+p(p+1)/2)$ is the dimension of $\hat{\theta}$ and proportional to $k$, where $p$ denotes the dimension of the input $X$. The second term of the CIC in (18) penalizes the model complexity by being linearly proportional to $d$.

The CIC is an asymptotically unbiased estimator of the cross-entropy (up to a multiplicative constant) under one key assumption that there exists $\theta^*$ such that $q^*(x) = q(x; \theta^*)$, and several other regularity conditions (see [21] for details) that are similarly required for the Akaike information criterion (AIC) [22] to be an asymptotically unbiased estimator of the true log-likelihood. Since for stochastic simulation models, $h(x)$ is unknown, it has to be estimated through $\hat{h}(x)$. We will present the CIC formulation for SIS in the next section.

3. Methodology

This section proposes a model selection solution to the estimation of the optimal IS density for stochastic simulation models. We use the GMM to express candidates for the optimal density. The CIC will automatically determine the mixture order $k$ of the GMM and control the model complexity. Among different
mixture models, we choose the GMM since it is the most widely used mixture model thanks to its ability to model any smooth distribution. In general, the proposed CE method is not limited to just the GMM. It can be extended to other mixture models as long as we can minimize a CE estimator. In the convenient case of the GMM, an EM algorithm can be applied to estimate the GMM parameters. Moreover, it is computationally efficient to sample from the GMM.

It is natural to apply the concept of CIC in DIS to SIS. The expression of CIC involves \( h(x) \), which is known in DIS but not in SIS. Hence we use the estimator \( \hat{h}(x) \) which is a function of \( \hat{s}(x) \), the estimated probability of failure at a particular \( x \) value.

### 3.1. Approximations Necessary for Implementation

To implement the proposed method, we need to know \( h(x) \) for computing CIC, evaluating the EM algorithm equations (14)-(17) to solve for \( \hat{\theta} \), and minimizing the CE estimator. For SIS, \( h(x) = \sqrt{s(x)(1-s(x))/n + s(x)^2} \) needs to be estimated because \( s(x) \) is unknown. We estimate \( s(X_i) \) by

\[
\hat{s}(X_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} I(Y_j^{(i)} > l)
\]

and then estimate \( h(x) \) by plugging in \( \hat{s}(x) \):

\[
\hat{h}(x) = \sqrt{\hat{s}(x)(1-\hat{s}(x))/n + \hat{s}(x)^2}.
\]

As the sample size grows large, \( \hat{h}(x) \) asymptotically converges to \( h(x) \). Specifically, if \( n \) increases faster than \( m \), then the convergence holds by the law of large numbers and the continuous mapping theorem.

SIS also needs to allocate \( N_i \) replications at each \( X_i \), as explained in Section 2. In Appendix 1, we show that for a large \( n \gg \max_{i=1}^{m} (1-s(X_i))/s(X_i) \), the optimal \( N_i \) in (6) is approximately proportional to \( \sqrt{w(X_i) - \hat{P}_{SIS}} \). Thus, we decide \( N_i \) based on this approximation. If \( w(X_i) - \hat{P}_{SIS} \leq 0 \), we assign \( N_i = 1 \), to ensure the unbiasedness of \( \hat{P}_{SIS} \) in (4).

### 3.2. Aggregated Failure Probability Estimation

We aggregated use the samples obtained in all of the CE iterations. Instead of \( \hat{P}_{SIS} \) in (4), we compute the failure probability estimator \( \hat{P}_{SIS}^{(t)} \) using all the
aggregated data up to iteration $t$:

$$\hat{P}_{SIS}^{(t)} = \frac{1}{t+1} \sum_{s=0}^{t} \frac{1}{m(s)} \sum_{i=1}^{m(s)} \frac{1}{N_i^{(s)}} \sum_{j=1}^{N_i^{(s)}} \mathbb{I}(Y_{ij}^{(s)} > l) \frac{f(X_i^{(s)})}{q(X_i^{(s)}; \hat{\theta}^{(s)})}, \quad (23)$$

where $X_i^{(s)}$ is the $i^{th}$ random input at iteration $s$, $m(s)$ is the number of $X_i^{(s)}$ drawn at iteration $s$, and $N_i^{(s)}$ is the number of simulation runs at each $X_i^{(s)}$.

The idea of data aggregation leads to the aggregated CIC formulation for stochastic simulation models as follows:

$$\text{CIC}_{SIS}^{(t)}(d) = \bar{C}_{SIS}^{(t-1)}(\hat{\theta}) + \hat{K}_{SIS}^{(t-1)} \frac{d}{\sum_{s=0}^{t-1} m(s)}, \quad (24)$$

where the CE estimator $\bar{C}_{SIS}^{(t-1)}(\hat{\theta})$ and $\hat{K}_{SIS}^{(t-1)}$ are calculated using all the aggregated data up to iteration $(t-1)$:

$$\bar{C}_{SIS}^{(t-1)}(\hat{\theta}) = -\frac{1}{\sum_{s=0}^{t-1} m(s)} \sum_{s=0}^{t-1} \sum_{i=1}^{m(s)} \hat{h}(X_i) w(X_i; \hat{\theta}) \log q(X_i; \hat{\theta}) \quad (25)$$

Noting that $\hat{K}_{DIS}^{(t-1)}$ in (18) is an unbiased DIS estimator of the failure probability, we similarly set

$$\hat{K}_{SIS}^{(t-1)} = \hat{P}_{SIS}^{(t-1)}. \quad (26)$$

3.3. Summary of the Proposed Method (CE-SIS)

For the EM initialization, the grid search of the optimal number of mixture components $k^{(t)*}$, the stopping criteria of the EM algorithm to minimize the quantity $\bar{C}_{SIS}^{(t-1)}(\hat{\theta})$ in (25), and whole division of the replication allocation $N_i^{(t)}$, we follow the same procedures in [21]. The details of implementation are provided in Appendix 2.

We propose the CE-SIS pseudo-code in Algorithm 1:

**INPUT:**

- Initialize the GMM $q(x; \hat{\theta}^{(0)})$ for the first iteration (e.g., GMM with $k = 1$ and the identity covariance matrix).
• Given a simulation budget, \( n \), we pick the actual simulation runs at each iteration, \( n(t) \), such that \( \sum_{t=0}^{\tau} n(t) = n \), where \( \tau \) denotes the number of total CE iterations.

• Fix the number of \( X \) values to draw, \( m(t) \), for each iteration \( t \geq 1 \) such that, for example, \( m(t) \approx 0.3n(t) \), as adopted in [12]. According to [12], SIS should be generally insensitive to the ratio \( \frac{m(t)}{n(t)} \). Note that at the first iteration \( t = 0 \), we set \( m(0) = n(0) \) or equivalently \( N_i(0) = 1 \), to explore the input space as much as possible.

• Pick the maximum number of components of the GMM, \( k_{\text{max}} \), for each iteration \( t \). \( k_{\text{max}} \) can depend on \( t \) since it is bounded by a function of the available sample size.

4. Numerical Example

In this section, we use a numerical example to evaluate the performance of the CE-SIS algorithm presented above. Assuming that the data generating process is unknown, we compare the performance of the CE-SIS algorithm with the state-of-the-art metamodel-based approach in [3].

Cannamela et al. [23] originally develop a deterministic simulation model example, which is later modified in [12] into a stochastic simulation model example. We test the CE-SIS algorithm with the latter numerical example. Its data generating structure is as follows:

\[
X \sim \mathcal{N}(0, 1), \quad Y | X \sim \mathcal{N}(\mu(X), \sigma^2(X)),
\]

where the mean and the standard deviation are:

\[
\mu(X) = 0.95X^2 (1 + 0.5 \cos(5X) + 0.5 \cos(10X)),
\]
\[
\sigma(X) = 1 + 0.7 |X| + 0.4 \cos(X) + 0.3 \cos(14X).
\]

To approximate the optimal density in (5) and the allocation in (6), the metamodel-based method in [3] also needs to estimate the conditional probability \( s(X) = P(Y > l \mid X) \) using a metamodel. The metamodel for the
Algorithm 1 Approximating the optimal IS density for SIS

$t = 0$

while $t \leq \tau$ do

if $t = 0$ then

$N_i^{(0)} = 1$ for $i \in \{1, \ldots, m^{(0)}\}$

for $i \in \{1, \ldots, m^{(0)}\}$ do

Sample $X_i^{(0)}$ from $q(x; \hat{\theta}^{(0)})$, an initial GMM

Generate the data using the simulator: $D = D^{(0)} = \{(X_i^{(0)}, Y_{i1}^{(0)})\}$

end for

Compute $\bar{P}_{SIS}^{(0)}$ in (23)

else

for $l \in \{k_{\text{min}}^{(t)}, \ldots, k_{\text{max}}^{(t)}\}$ do

Using EM: $\hat{\theta}^{(t)}(l) = \arg\min_{\theta(l) \in \Theta(d(l))} \bar{P}_{SIS}^{(t-1)}(\hat{\theta}(l))$ in (25)

Let $k_{SIS}^{(t-1)}(l) = P_{SIS}^{(t-1)}(l)$ in (26)

Compute $\text{CIC}_{SIS}^{(t)}(d(l))$ in (24)

end for

Pick $k^{(t)*} = \arg\min_l \text{CIC}_{SIS}^{(t)}(d(l))$

$\hat{\theta}^{(t)} \leftarrow \hat{\theta}(k^{(t)*})$

for $i \in \{1, \ldots, m^{(t)}\}$ do

Sample $X_i^{(t)}$ from $q(x; \hat{\theta}^{(t)})$

Compute $\sqrt{w(X_i^{(t)}) - \bar{P}_{SIS}^{(t-1)}}$

end for

for $i \in \{1, \ldots, m^{(t)}\}$ do

Compute $N_i^{(t)} = \max \left(1, \text{round} n^{(t)} \frac{\sqrt{w(X_i^{(t)}) - \bar{P}_{SIS}^{(t-1)}}}{\sum_{i=1}^m \sqrt{w(X_i^{(t)}) - \bar{P}_{SIS}^{(t-1)}}} \right)$

end for

for $i \in \{1, \ldots, m^{(t)}\}$ do

for $j \in \{1, \ldots, N_i^{(t)}\}$ do

Generate the data using the simulator: $D^{(t)} = \{X_i^{(t)}, Y_{ij}^{(t)}\}$

end for

end for

Aggregate data: $D = \bigcup_{s=0}^t D^{(s)}$

Compute $\bar{P}_{SIS}^{(t)}$ in (23)

end if

$t \leftarrow t + 1$

end while
conditional distribution $P(Y \mid X)$ is set as the normal distribution with the parameters, mean and standard deviation, modeled as cubic spline functions in the generalized additive model for location, scale and shape (GAMLSS) framework [24]. The metamodel is built using a pilot dataset of 600 simulation replications and the IS experiment is based on 1000 simulation replications.

To ensure that both methods use the same simulation budget, for the CE-SIS method, we use $n^{(0)} = 600$ and $n^{(t)} = 100$ for $t = 1, \ldots, 10$. The experiment is repeated 500 times to obtain the results in Table 1. Since we have full knowledge about the underlying process, we also show in Table 1 the result obtained from the optimal SIS density that uses the true $s(X)$ function.

The performance metric we use is the CMC ratio, which is defined as

$$\text{CMC ratio} = \frac{n}{n_{CMC}},$$

where $n$ is the total number of simulation replications (e.g., 600 + 1000 for metamodel-based SIS, 600 + 1000 for CE-SIS, and 1000 for the optimal SIS) and $n_{CMC}$ is the total number of simulation replications required for the crude Monte Carlo simulation to achieve the same standard error as each method, which is calculated as

$$n_{CMC} = \frac{\hat{P}(1 - \hat{P})}{S.E.^2}.$$

Note that $S.E.$ is the standard error of each method. $\hat{P}$ is the probability estimate from the optimal SIS, which is most accurate, i.e., 0.00996. In Table 1, the smaller CMC ratio translates to a smaller number of replications used in each row’s method divided by the number of replications necessary for CMC in (1) to achieve the standard error in the row, which means better computational saving for the same accuracy level.

It appears that the metamodel-based approach is slightly better than the CE-SIS method. The result is not surprising in the following context. First, the metamodel represents one of the best metamodels that can be built in practice since the correct distribution family (i.e., Gaussian) is used for $Y \mid X$ for building the metamodel. In reality, the underlying data generating distribution is unknown and likely much more complex, rendering the construction of a
Table 1: Comparison between CE-SIS, metamodel-based SIS, and the optimal SIS

| Method     | Mean     | Standard Error | CMC Ratio |
|------------|----------|----------------|-----------|
| CE-SIS     | 0.00972  | 0.00073        | 8.65%     |
| Metamodel  | 0.01003  | 0.00068        | 7.50%     |
| Optimal SIS| 0.00996  | 0.00052        | 2.74%     |

good metamodel difficult, especially when the pilot dataset is small. In fact, the main advantage of the CE-SIS method is the ability to automatically come up with the best approximated importance sampling density. In other words, scalability is the main advantage of CE-SIS method, especially when we want to develop an off-the-shelf software package for SIS. Second, the difference between the standard errors of two methods, 0.00005, is small (i.e., 0.5% of the failure probability), which means the CE-SIS method can achieve performance close to the best possible models.

5. Case Study

This section presents an application of the CE-SIS method to the case study on wind turbine reliability evaluation. We compare the performance between the CE-SIS method and the metamodel-based method [12], using the same CMC ratio metric in Section 4.

The reliability of a wind turbine is subject to stochastic weather [25]. To incorporate the randomness into the reliability evaluation of a turbine design, the international standard, IEC 61400-1 [26], requires the turbine designer to use stochastic simulations. We evaluate the reliability of a wind turbine using computationally expensive aerodynamic simulators (each simulation run takes roughly 1 minute on a typical PC available nowadays) developed by the U.S. National Renewable Energy Laboratory. Specifically, TurbSim is used to generate a 3-dimensional time-marching wind profile, and FAST is used to simulate a turbine’s structural bending moment responses [12, 27, 28]. Each simulation represents a 10-min simulated operation of the turbine. Our simulation setup
is identical with the benchmark setting used in [29].

We are interested in estimating the probability of a failure event which is defined as the bending moment at a turbine’s blade root exceeding a specified threshold. In particular, we study two bending moment types: edgewise and flapwise. We estimate the probability that a bending moment exceeds a threshold \( l \), where \( l = 8,600 \) kNm for edgewise bending moment and \( l = 13,800 \) kNm for flapwise bending moment, both of which occur with around 5% probability.

The input wind speed, \( X \), is drawn from the truncated Rayleigh density [12, 30] defined as:

\[
f(x) = \frac{f_R(x)}{F_R(x_{out}) - F_R(x_{in})},
\]

where \( f_R \) is the untruncated Raleigh density with the shape parameter, \( 10\sqrt{2/\pi} \), and \( F_R \) is the corresponding cumulative distribution function. The cut-in and cut-out wind speed are \( x_{in} = 3 \) m/s and \( x_{out} = 25 \) m/s, respectively, denoting the range of wind speeds for which a wind turbine operates.

In the metamodel-based method in [12], the conditional probability \( s(X) = P(Y > l \mid X) \) is approximated using a nonhomogenous Generalized Extreme Value (GEV) distribution:

\[
P(Y \leq y \mid X = x) = \begin{cases} 
    \exp \left( - \left( 1 + \frac{\xi (y - \mu(x))}{\sigma(x)} \right)^{-1/\xi} \right), & \text{for } \xi \neq 0 \\
    \exp \left( -\exp \left( -\frac{y - \mu(x)}{\sigma(x)} \right) \right), & \text{otherwise,}
\end{cases}
\]

where \( \mu(x) \) and \( \sigma(x) \) are the location and scale parameter functions, respectively, modeled with cubic smoothing spline functions under the GAMLSS framework. The shape parameter \( \xi \) is fixed as a constant, with estimated value \( \hat{\xi} = -0.0359 \) \((-0.0529)\) for the edgewise (flapwise) case. The goodness-of-fit of the GEV distribution is tested using the Kolmogorov-Smirnov test. The metamodel is built using a pilot sample of 600 simulation replications and in addition, the metamodel-based method uses 1000 (2000) replications for the failure probability estimation for edgewise (flapwise) bending moment.

In the CE-SIS method, we allocate the same simulation budget as the metamodel-based method. We use \( n^{(0)} = 600 \) and \( n^{(t)} = 100 \) (200) for \( t = \)

\[18\]
1, . . . , 10 for the edgewise (flapwise) case.

Table 2 compares the results based on 50 repetitions of each method. The CE-SIS method has slightly smaller (larger) standard error than the metamodel-based method for the edgewise (flapwise) bending moment. Accordingly, both methods save the similar level of computational resource compared to CMC, as indicated by the CMC Ratio. In the metamodel-based method, the metamodel is carefully built by fitting a nonhomogeneous GEV distribution to the pilot data. We can see that the performance of the CE-SIS method is comparable to that of the metamodel-based SIS with a high quality metamodel. Note that since CE-SIS is an automated method, it can be particularly helpful when building a metamodel is difficult. We also observe that the computational saving in the edgewise case is more substantial than the flapwise case. This is because the approximated SIS density is not very different from the original density in the flapwise case (this phenomenon is first observed and discussed extensively in [12], to which an interested reader is referred). Nevertheless, we see that the CE-SIS method, without requiring domain knowledge about the underlying process, can still capture this information satisfactorily.

Table 2: Comparison between the metamodel-based SIS and the CE-SIS method for the case study

| Response | Method   | Mean  | Standard Error | CMC Ratio |
|----------|----------|-------|----------------|-----------|
| Edgewise | Metamodel| 0.0486| 0.0018         | 7.0%      |
|          | CE-SIS   | 0.0486| 0.0015         | 4.9%      |
| Flapwise | Metamodel| 0.0514| 0.0028         | 32%       |
|          | CE-SIS   | 0.0535| 0.0030         | 37%       |

6. Conclusion

We propose a method called the cross-entropy based stochastic importance sampling (CE-SIS) which can efficiently construct an importance sampling density for a stochastic simulation model. The CE-SIS method uses an EM algorithm to minimize the estimated cross-entropy from a candidate IS density to
the optimal IS density while penalizing the model complexity concurrently. The method automatically refines the estimated IS density, thus not requiring the specific domain knowledge for building a metamodel, which is often difficult or time-consuming in practice. We focus on a candidate IS density expressed as a Gaussian mixture model, which is both flexible and computationally efficient, while the extension to other mixture models is possible. The application of the cross-entropy information criterion allows a sound choice of the mixture model complexity for the CE-SIS method. By aggregating all the data from previous iterations and effectively allocating the number of replications at each input value, the CE-SIS method utilizes the available information efficiently under a given simulation budget. The numerical example and case study show the advantage of the CE-SIS over the metamodel-based SIS and crude Monte Carlo simulation.
Appendix 1: Approximation of $N_i$

We seek an asymptotic approximation of the optimal allocation size, 

$$N_i = n \frac{\sqrt{\frac{n(1-s(X_i))}{1+(n-1)s(X_i)}}}{\sum_{j=1}^{m} \sqrt{\frac{n(1-s(X_j))}{1+(n-1)s(X_j)}}}, \quad i = 1, \ldots, m.$$ 

First, we show $s(X_i) \approx \hat{p}_{SIS} w(X_i)$. For a large $n \gg \max_{i=1}^{m} (1 - s(X_i))/s(X_i)$, we can approximate

$$q_{SIS}(X_i) = \frac{1}{C_q} f(X_i) \sqrt{\frac{1}{n} s(X_i) (1 - s(X_i)) + s(X_i)^2}$$

$$\approx \frac{1}{C_q} f(X_i) \sqrt{s(X_i)^2}$$

$$= \frac{1}{C_q} f(X_i) s(X_i)$$

for any $i = 1, \ldots, m$. This asymptotic approximation may be not good for some $N_i$ if $s(X_i)$ is close to zero. However, in that case, $q(X_i)$ is small too, and such $X_i$ is unlikely to be sampled in the first place. Therefore, we can approximate

$$s(X_i) \approx C_q \frac{q_{SIS}(X_i)}{f(X_i)}$$

$$\approx \frac{C_q}{w(X_i)},$$

where the second approximation is based on the fact that the estimated IS density approximates the optimal density.

Furthermore, for a large $n \gg \max_{i=1}^{m} (1 - s(X_i))/s(X_i)$, we can also approximate

$$C_q = \int_{X_i} f(x) \sqrt{\frac{1}{n} s(x) \cdot (1 - s(x)) + s(x)^2} \, dx$$

$$\approx \int_{\{x: s(x) > 1/(n+1)\}} f(x) \sqrt{\frac{1}{n} s(x) \cdot (1 - s(x)) + s(x)^2} \, dx$$

$$\approx \int_{\{x: s(x) > 1/(n+1)\}} f(x) s(x) \, dx$$

$$\approx \hat{p}_{SIS},$$

21
where $X_f$ is the support of $f$. Thus, it follows that

$$s(X_i) \approx \frac{\hat{P}_{SIS}}{w(X_i)}.$$  

Therefore, for a large $n$,

$$N_i \propto \sqrt{\frac{n (1 - s(X_i))}{1 + (n - 1) s(X_i)}}$$

$$\approx \sqrt{\frac{1 - s(X_i)}{s(X_i)}}$$

$$\approx \sqrt{\frac{1 - \hat{P}_{SIS}}{\hat{P}_{SIS}}}$$

$$\propto \sqrt{w(X_i) - \hat{P}_{SIS}}.$$  

Although it does not happen frequently, if $w(X_i) - \hat{P}_{SIS} \leq 0$, then we set the corresponding $N_i$ as 1, the smallest allocation possible to maintain the unbiasedness of the SIS estimator.

**Appendix 2: Implementation details of the CE-SIS algorithm**

Since the EM algorithm will lead to a local optimum for non-convex problems, we use 10 random initializations of $\theta$ and choose the best minimizer $\hat{\theta}$ of $\bar{C}_{SIS}^{(t-1)}(\hat{\theta})$ in (25) to reduce the impact of initial guess of $\theta$ on the algorithm’s performance [31].

By monitoring the condition numbers of the Gaussian components’ covariances [31], the number of components $k$ can also be controlled to prevent over-fitting issue within the EM algorithm. We choose a singularity threshold of $10^5$ for the covariance matrix. Exceeding the threshold will signify an ill-conditioned matrix. If more than a half of the 10 initializations observe ill-conditions, we stop the EM algorithm and consider $k_{\text{max}}$ is reached for that iteration.

Checking the convergence of the EM algorithm is through monitoring the reduction of $\bar{C}_{SIS}^{(t-1)}(\hat{\theta})$ in (25). In the numerical study in Section 4, iterating the
updating equations in the EM algorithm is stopped if the reduction of $\bar{C}_{SIS}^{(t-1)}(\hat{\theta})$ is less than 1% or a specified maximum number of iterations is reached.

For the $N_i^{(t)}$ allocation at each iteration $t$, we round the numerical $N_i^{(t)}$ to the nearest integer. The difference between $\sum_{i=1}^m N_i^{(t)}$ and $n^{(t)}$ due to rounding is spread across the largest $N_i^{(t)}$’s equally to make sure we strictly stay within the simulation budget. The purpose of spreading across the largest $N_i^{(t)}$’s (as opposed to, say, the smallest ones) is to minimize the deviations from the prescribed $N_i^{(t)}$’s. Note that SIS is generally insensitive to variations of $N_i^{(t)}$’s [12].

References

[1] D. P. Kroese, T. Taimre, Z. I. Botev, Handbook of Monte Carlo Methods, New York: John Wiley and Sons., 2011.

[2] H. Kahn, A. W. Marshall, Methods of reducing sample size in Monte Carlo computations, Journal of the Operations Research Society of America 1 (5) (1953) 263–278.

[3] Y. Choe, H. Lam, E. Byon, Uncertainty quantification of stochastic simulation for black-box computer experiments, Methodology and Computing in Applied Probability. URL https://doi.org/10.1007/s11009-017-9599-7

[4] P. Glasserman, J. Li, Importance sampling for portfolio credit risk, Management Science 51 (11) (2005) 1643–1656.

[5] S. Asmussen, K. Binswanger, B. Højgaard, Rare events simulation for heavy-tailed distributions, Bernoulli 6 (2) (2000) 303–322.

[6] P. Heidelberger, Fast simulation of rare events in queueing and reliability models, ACM Transactions on Modeling and Computer Simulation (TOMACS) 5 (1) (1995) 43–85.
[7] M. Balesdent, J. Morio, L. Brevault, Rare event probability estimation in the presence of epistemic uncertainty on input probability distribution parameters, Methodology and Computing in Applied Probability 18 (1) (2016) 197–216.

[8] C. Chang, P. Heidelberger, S. Juneja, P. Shahabuddin, Effective bandwidth and fast simulation of ATM intree networks, Performance Evaluation 20 (1) (1994) 45–65.

[9] J. S. Sadowsky, Large deviations theory and efficient simulation of excessive backlogs in a GI/GI/m queue, IEEE Transactions on Automatic Control 36 (12) (1991) 1383–1394.

[10] J. Blanchet, P. Glynn, H. Lam, Rare event simulation for a slotted time M/G/s model, Queueing Systems 63 (1-4) (2009) 33–57.

[11] J. Blanchet, H. Lam, Rare-event simulation for many-server queues, Mathematics of Operations Research 39 (4) (2014) 1142–1178.

[12] Y. Choe, E. Byon, N. Chen, Importance sampling for reliability evaluation with stochastic simulation models, Technometrics 57 (3) (2015) 351–361.

[13] R. Rubinstein, The cross-entropy method for combinatorial and continuous optimization, Methodology and Computing in Applied Probability 1 (2) (1999) 127–190.

[14] V. Dubourg, B. Sudret, F. Deheeger, Metamodel-based importance sampling for structural reliability analysis, Probabilistic Engineering Mechanics 33 (2013) 47–57.

[15] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, P. L’Ecuyer, The cross-entropy method for optimization, Machine Learning: Theory and Applications, V. Govindaraju and C. R. Rao, Eds, Chennai: Elsevier 31 (2013) 35–59.

[16] R. Rubinstein, A stochastic minimum cross-entropy method for combinatorial optimization and rare-event estimation, Methodology and Computing in Applied Probability 7 (1) (2005) 5–50.
[17] Z. I. Botev, D. P. Kroese, T. Taimre, Generalized cross-entropy methods with applications to rare-event simulation and optimization, Simulation 83 (11) (2007) 785–806.

[18] Y. Chen, Y. Choe, Oracle importance sampling for stochastic simulation models, arXiv preprint arXiv:1710.00473.

[19] H. Wang, X. Zhou, A cross-entropy scheme for mixtures, ACM Transactions on Modeling and Computer Simulation 25 (1) (2015) 6:1–6:20.

[20] N. Kurtz, J. Song, Cross-entropy-based adaptive importance sampling using gaussian mixture, Structural Safety 42 (2013) 35–44.

[21] Y. Choe, Information criterion for minimum cross-entropy model selection, arXiv preprint arXiv:1704.04315.

[22] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6) (1974) 716–723.

[23] C. Cannamela, J. Garnier, B. Iooss, Controlled stratification for quantile estimation, The Annals of Applied Statistics 2 (4) (2008) 1554–1580.

[24] R. A. Rigby, D. M. Stasinopoulos, Generalized additive models for location, scale and shape, Journal of the Royal Statistical Society: Series C (Applied Statistics) 54 (3) (2005) 507–554.

[25] E. Byon, L. Ntaimo, Y. Ding, Optimal maintenance strategies for wind power systems under stochastic weather conditions, IEEE Transactions on Reliability 59 (2) (2010) 393–404.

[26] International Electrotechnical Commission, IEC/TC88, 61400-1 ed. 3, Wind Turbines - Part 1: Design Requirements. (2005).

[27] J. M. Jonkman, M. L. Buhl Jr., FAST User’s Guide, Tech. Rep. NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, Colorado (2005).
[28] B. J. Jonkman, TurbSim user’s guide: version 1.50, Tech. Rep. NREL/TP-500-46198, National Renewable Energy Laboratory, Golden, Colorado (2009).

[29] P. Moriarty, Database for validation of design load extrapolation techniques, Wind Energy 11 (6) (2008) 559–576.

[30] Y. Choe, Q. Pan, E. Byron, Computationally efficient uncertainty minimization in wind turbine extreme load assessments, ASME Journal of Solar Energy Engineering 138 (4) (2016) 041012–041012–8.

[31] M. A. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 381–396.