Importance Sampling for Rare Event Estimation with Computer Experiments

Keith Dalbey and Laura Swiler
Optimization and Uncertainty Quantification Department
Sandia National Laboratories

Brian Williams and Rick Picard
Statistical Sciences Group, Los Alamos National Laboratory

SIAM Conference on Uncertainty Quantification
April 5, 2012
Raleigh, NC
Overview

• Background on Importance Sampling
• Two approaches
  – Sequential Importance Sampling (LANL)
  – Gaussian Process guided Adaptive mixture Importance Sampling (SNL)
• Role of surrogates
• NEAMS (Nuclear Energy Advanced Modeling & Simulation Program) Example
• Issues/Conclusions
Rare Event Estimation

• Assume:
  – Outputs obtained from “black-box” computational model.
  – We assume the uncertain variables are represented by $\theta$, and the uncertainty has been characterized (e.g. based on experimental data, previous calibration, etc).

• Want to estimate: $\Pr_{\theta}[y(\theta) > T_\alpha] = \alpha_T$

• Importance sampling is a method used to sample random variables from different densities than originally defined.

• These importance sampling densities are constructed to pick “important” values of input random variables to improve the estimation of a statistical response of interest, such as a mean or probability of failure.

• For black-box simulations, we cannot generally assume that the optimal importance sampling density will be normal, have a standard parametric form, or allow exact calculation (including normalizing constant).
Importance Sampling

- **Objective:** Calculate probability of failure $\alpha_T = \Pr_\theta [y(\theta) > T_\alpha]$ when $\theta$ is randomly distributed with PDF $\rho(\theta)$

$$\alpha_T = \int I(\theta) \rho(\theta) d\theta$$

where the indicator function $I(\theta) = 1$ if failure occurs and 0 otherwise.

- Above integral equivalent to

$$\alpha_T = \int I(\theta) \frac{\rho(\theta)}{g(\theta)} g(\theta) d\theta$$

which is approximated by Importance Sample Mean

$$\hat{\alpha}_T = \frac{1}{n} \sum_{i=1}^{n} I(\theta_i) \frac{\rho(\theta_i)}{g(\theta_i)}$$

Sampled from $g(\theta)$
How to choose importance sampling density?

• **Challenge:** “There is no general recipe (for choosing an importance density), and the issue remains largely a matter of art in the literature. Most parametric distributions fail to include \( g \) (the optimal importance sampling density) as a member.”

• **Ideal Importance Density:** \( g(\theta) \) ideally chosen to increase the likelihood of observing desired rare events
  
  – Minimum variance importance density has property:

  \[
g(\theta) \propto I(\theta)\rho(\theta)
  \]

  – Is unknown because normalization constant (the probability of failure we need to calculate) is unknown

How to proceed? Two approaches:

1. Sequential importance sampling
2. Gaussian process guided importance sampling
Sequential importance sampling

- Oversample region of parameter space producing rare events of interest
- Sequentially refine importance distributions for improved inference:
  Choose $g(\theta)$ by iterative refinement

1. Sample from initial importance density $g^{(1)}(\theta) = \rho(\theta)$ (or alternative based on physics knowledge) and determine which variables are sensitive for producing extreme output values.
   - Calibrate to a particular quantile such as 0.0001 or specified percentile.
2. For sensitive variables from (1), estimate parameters of distribution family selected for importance sampling.
   - For insensitive parameters, keep joint (or conditional) distribution based on (1).
   Combine these distributions (e.g. independence assumptions) to obtain $g^{(2)}(\theta)$.
3. Repeat (2) until the updated importance distribution “stabilizes” in its approximation of the minimum variance importance distribution.
   - Means and variances of sensitive parameters are now estimated via their conditional distributions given $[y(\theta) > T_\alpha]$. 
Sequential importance sampling

Quantile vs. Percentile Estimation

– Obtain samples \( \{\theta^{(j)}, j=1,\ldots,N\} \) from the density \( g(\theta) \)
– Calculate importance weights \( w_i = \rho(\theta_i)/g(\theta_i) \)
– Evaluate computational model on samples
– Quantile Estimation: Sort code evaluations and choose an appropriate order statistic from the (weighted) list (based on fixed percentile \( \alpha_T \))
  - e.g. if \( \alpha_T = 0.001 \) and \( N=10,000 \), choose the 11-th largest value
  - Not unique, e.g. choose any value between the 10-th and 11-th largest values
– Percentile Estimation: Code runs exceeding fixed quantile

Normalization Issues

– Weights may be unnormalized; consider normalizing by the sum \( \sum_{i=1}^{n} \rho(\theta_i)/g(\theta_i) \)
  - e.g. The original density may be a posterior distribution from a Bayesian analysis, with unknown normalizing constant
  - Unfortunately, simulation studies suggest gross inaccuracies likely when using normalized weights
Sequential importance sampling

• Diagnostic measures for importance sampling

Variance Reduction Factor

\[ VRF = \frac{\text{Brute Force Variance}}{\text{Variance of IS quantity}} \]

Computational Efficiency

\[ CE = VRF \times \frac{\text{Run Time Brute Force}}{\text{Run Time IS}} \]
Sequential importance sampling

• Many applications will require code surrogates
  – Many computational models run too slowly for direct use in brute force or importance sampling based rare event inference
  – Use training runs to develop a statistical surrogate model for the complex code (i.e., the emulator)
  – Choose design augmentation that minimizes integrated mean square error with respect to the currently estimated importance distributions for sensitive parameters
  – Sequential design improves surrogate quality in region of parameter space indicated by importance distributions
  – Importance distributions and VRFs stabilize quickly, while percentile/quantile estimates may converge more slowly because of emulator bias
Gaussian process adaptive importance sampling

Uses a Mixture of Importance Densities

- The mixture is constructed by a weighted set of individual densities. The mixture weights, $w_j$, are the fraction of the total number of samples, $N$, drawn from component density $\rho_j$

$$g(\theta) = \rho^M(\theta) = \sum_{j=0}^{j} w_j \rho'_j(\theta)$$

- Almost as good” as only the best density in the mixture

- The individual densities are: $\rho'_0(\theta) = u(\theta)$, $\rho'_j(\theta) \propto E_j[I(\theta)]\rho(\theta)$

- Gaussian Process surrogate can estimate the indicator function (importance density) at not yet sampled points

- The GP Expected Indicator (a real number between 0 and 1) is:

$$E_j[I(\theta)] = \frac{1}{2} \left( 1 + erf \left( \frac{T_\alpha - E_j(y(\theta)))}{\sqrt{2Var_j(y(\theta))}} \right) \right)$$
Gaussian process adaptive importance sampling

Procedure
• Take initial set of LHS samples from $\rho'_0(\theta) = u(\theta)$
• Construct initial GP
• For j=1:J
  1. Generate 10K or 100K samples of current GP$_j$
  2. Estimate normalization constant $\beta_j$
  3. Generate one draw from $\rho'_j(\theta)$
  4. Evaluate function at the sample draw
  5. Rebuild the GP after adding this new sample draw point
  6. $j = j + 1$, go to 1 and continue
• At end of process, use overall formula $\hat{\alpha}_T = \frac{1}{n} \sum_{i=1}^{n} I(\theta_i) \frac{\rho(\theta_i)}{\rho_M(\theta_i)}$

Rebuild GP after each importance sample $\rightarrow$ an adaptive mixture of importance densities (GPAIS)
Conceptual Challenges in GPAIS

Challenges:

1. Points can pile up when $\alpha_T$ is small $\rightarrow$ GP’s correlation matrix, $R$, becomes ill conditioned (numerically singular). **Solution:** add a small nugget to diagonal of $R$

2. How to determine proportionality constant $\beta$

$$\rho_j(\theta) = \beta_j E_j[I(\theta)] \rho(\theta)$$

$$\beta_j = \frac{1}{\int E_j[I(\theta)] \rho_j(\theta) d\theta}$$

3. How to draw from IMPLICITLY DEFINED importance density

**Joint solution of 2. & 3.:** evaluate GP emulator a large number of times; from this ensemble estimate the normalization constant & select one importance sample
How to Estimate $\beta$

- Evaluate emulator j’s expected indicator at $n$ points $\theta_k$ ($1 \leq k \leq n$) and calculate
  \[ f_k = E_j(I(\theta_k)) \frac{\rho(\theta_k)}{u(\theta_k)} \]

- How large does $n$ need to be?
- If $S = \sum_{k=1}^{n} f_k \geq C$ then the standard error in the normalization constant for this GP looks like $\alpha_T/S^{0.5}$, so $C = 25 \rightarrow S \approx 20\%$ of $\alpha_T$. The proportionality constant looks like
  \[ \beta \approx \frac{1}{\frac{1}{n} \sum_{k=1}^{n} f_k} \]

- However, if $S = 0$ after a very large $n$, can “give up” & draw from the same distribution (“uniform”) as samples used to build initial GP
Case study: R7 Virtual Reactor

- Simple thermal-hydraulics loop that represents a simplified plant
- The loop is 10m tall and 10m wide.
- The loop has 8 pipes, 4 elbows, a pump, the core, a heater and heat exchanger (HX), a pressurizer, etc.
- The working fluid is water at high temperature and pressure, using single phase flow.
- The power output of this reactor is set to a nominal value of 1.25 MW.
Case study: R7 Virtual Reactor
Pressurizer Failure Scenario

Cumulative Distribution Function (CDF)
based on 10,000 LHS Samples

Prob (PCT ≤ 700 = 0.9964)
Prob (PCT > 700 = 0.0036)
This is what we want to estimate.
Case study: R7 Virtual Reactor Pressurizer Failure Scenario

- Six dimensional problem
- Good results from initial testing: adaptive sampling working
- Estimates with 240 samples vs. 10K, generating 15-30% of samples in failure region

<table>
<thead>
<tr>
<th>CCDF</th>
<th>10000 LHS Samples</th>
<th>Bootstrap Estimate of Std Dev of CCDF</th>
<th>GPAIS: 240 Samples 60 Initial/180 Adaptive</th>
<th>GPAIS Estimate of Std Dev of CCDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob(PCT &gt; 680)</td>
<td>0.0101</td>
<td>0.001023</td>
<td>0.0090</td>
<td>0.001389</td>
</tr>
<tr>
<td>Prob(PCT &gt; 690)</td>
<td>0.0071</td>
<td>0.000809</td>
<td>0.0058</td>
<td>0.000334</td>
</tr>
<tr>
<td>Prob(PCT &gt; 700)</td>
<td>0.0036</td>
<td>0.000628</td>
<td>0.0034</td>
<td>0.000515</td>
</tr>
<tr>
<td>Prob(PCT &gt; 710)</td>
<td>0.0012</td>
<td>0.000361</td>
<td>0.0011</td>
<td>0.000233</td>
</tr>
<tr>
<td>Prob(PCT &gt; 720)</td>
<td>0.0001</td>
<td>0.000099</td>
<td>0.0002</td>
<td>0.000090</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CCDF</th>
<th>Average Fraction Failed</th>
<th>Average Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prob(PCT &gt; 680)</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>Prob(PCT &gt; 690)</td>
<td>0.34</td>
<td>0.38</td>
</tr>
<tr>
<td>Prob(PCT &gt; 700)</td>
<td>0.36</td>
<td>0.39</td>
</tr>
<tr>
<td>Prob(PCT &gt; 710)</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>Prob(PCT &gt; 720)</td>
<td>0.12</td>
<td>0.48</td>
</tr>
</tbody>
</table>
Case study: R7 Virtual Reactor
Pressurizer Failure: Quantile Inference

Importance Distributions

Variance Factors

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0001</td>
<td>1737</td>
<td>4994</td>
</tr>
<tr>
<td>0.00001</td>
<td>2953</td>
<td>7536</td>
</tr>
<tr>
<td>0.000001</td>
<td>1749</td>
<td>3159</td>
</tr>
</tbody>
</table>
Summary

- Importance sampling improves the estimation of percentiles and quantiles relative to a typical Monte Carlo sampling approach.
- The benefits of importance sampling increase as percentiles become more extreme.
- Surrogates are necessary for codes with long run time.
- Iterative refinement improves importance distributions in relatively few iterations.
- Developing a generalized approach to importance sampling for black-box applications, especially in high-dimensions, can be challenging:
  - There may not be good parametric forms for the importance distribution and/or parametric forms may need to account for correlations.
  - Estimating normalization constants can be difficult and small errors in these constants can potentially affect results significantly.
Two approaches

• Sequential Importance Sampling
  – Can handle both percentile and quantile estimation
  – Human-in-the-loop to estimate sequential importance densities although this can be fully automated; payoff is substantially improved inference with relatively small cost of very few iterations for iterative refinement.
  – Does not require a surrogate although one may be used if necessary.
  – Investigating issues of surrogate bias affecting inference results; potential convergence issues for percentile/quantile estimates.

• Gaussian Process guided Adaptive mixture Importance Sampling
  – Currently can only handle percentile estimation
  – Does not require a human-in-the-loop; the updating is automated.
  – Requires a set of Gaussian process surrogates
  – Investigating issues of normalization constant estimation, sampling on surrogate to generate next point.
  – Can also be used for calculating moments through integration