Methods to Predict Bias in Criticality Safety Applications using MCNP6 and Whisper

Michael E. Rising, Pavel Grechanuk, and Forrest Brown

Monte Carlo Algorithms, Codes and Applications
Los Alamos National Laboratory

OECD/NEA WPNCS SG-2 Meeting

Paris, FR

July 5, 2018

Outline

- **Motivation**
- **Background**
  - Upper Subcritical Limit
  - MCNP6 / Whisper
- **Bias Prediction Methods**
  - Extreme Value Theory
  - GLLS Method
  - Machine Learning
- **Generalized Results**
- **Application Toward SG-2 Blind Benchmark Study**
- **Conclusions & Future Work**

Motivation

- **Participate in WPNCS EGUACSA Phase V, now WPNCS SG-2**
  - Blind applications used to study bias and bias uncertainty calculations
- **Use new MCNP6 / Whisper-1.1 features**
  - MCNP6 computes nuclear data sensitivity profiles
  - Whisper-1.1 contains:
    - Catalogue of benchmarks with sensitivity profiles
    - Nuclear data covariance matrices
    - Ability to calculate correlations between application and benchmarks
    - Use of extreme value theory to compute bias and bias uncertainty
    - Estimate of margin of subcriticality using generalized linear-least squares method
- **Machine learning is current "hot topic"**
  - Summer student (P. Grechanuk) very interested in this…
- **Look at various methods to calculate bias for blind benchmark study**

Background
Upper Subcritical Limit

- **To consider a simulated system subcritical, the computed k_{eff} must be less than the Upper Subcritical Limit (USL):**

$$K_{calc} < USL$$

$$USL = 1 + (\text{Bias}) - (\text{Bias uncertainty}) - MOS$$

$$MOS = MOS_{data} + MOS_{code} + MOS_{application}$$

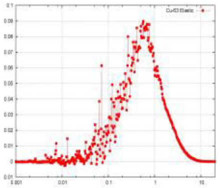
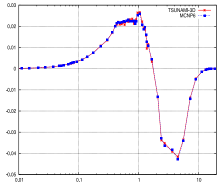
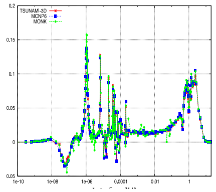
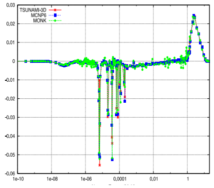
- **The bias and bias uncertainty are at some confidence level, typically 95% or 99%.**
  - These confidence intervals may be derived from a normal distribution, but the normality of the bias data must be justified.
  - Alternatively, the confidence intervals can be set using non-parametric methods.

Background
MCNP6 / Whisper (1)

- The **sensitivity coefficient** is the ratio of relative change in k-effective to relative change in a system parameter:

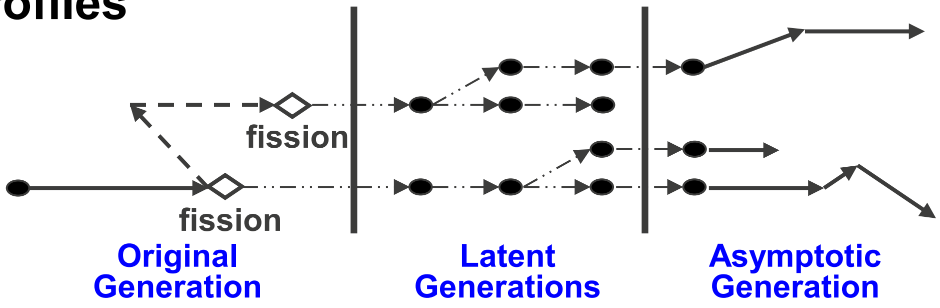
$$S_{k,x} = \frac{dk/k}{dx/x} = -\frac{\left\langle \psi^{\dagger}, \left(\Sigma_x - \mathrm{S}_x - k^{-1}\mathrm{F}_x\right)\psi \right\rangle}{\left\langle \psi^{\dagger}, k^{-1}\mathrm{F}\psi \right\rangle}$$

- $S_{k,x}(E)$ is the **sensitivity profile**, that includes all isotopes, reactions, & energies for a system:

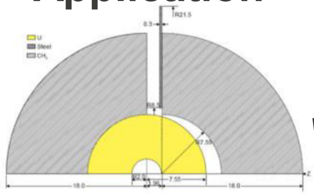
etc.

- **MCNP6 & Scale/Tsunami Monte Carlo** can use the Iterated Fission Probability method to compute adjoint-weighted integrals for the sensitivity profiles

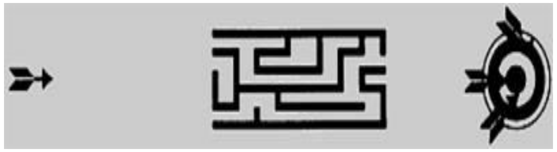


Background
MCNP6 / Whisper (2)

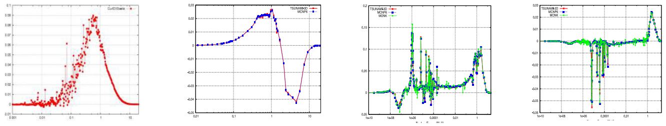
Application



Nuclear Cross-section Data

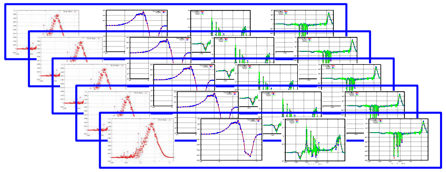


MCNP6 or Scale/Tsunami
Monte Carlo
Criticality Calculation



Application
Sensitivity Profile

Nuclear Cross-section Covariance Data



Catalog of sensitivity profiles for 1000s of experiments

SU-based Analysis

Whisper or Tsurfer

Pattern matching – application sensitivity profile vs catalog

Select similar experiments

Statistical analysis to determine bias & uncertainty & extra margin

USL
Upper Subcritical Limit for criticality safety analysis

Background
MCNP6 / Whisper (3)

- **Whisper**
  - Statistical analysis code to determine baseline USLs
  - Uses sensitivity profiles from continuous-energy MCNP6
  - Uses covariance data for nuclear cross-sections

- **Using Whisper**

  Run MCNP6 for an Application, & get Application sensitivity profile, S_A

  Run Whisper:

  ① Automated, physics-based selection of benchmarks that are neutronically similar to the application, ranked & weighted
  - Compare Application S_A to each of the Benchmark sensitivities $S_{B(i)}$
  - Select most-similar benchmarks (highest S_A-$S_{B(i)}$ correlation coefficients)

  ② Bias + bias uncertainty from Extreme Value Theory

  ③ Margin for nuclear data uncertainty estimated by GLLS method

Bias Prediction Methods

- **Extreme Value Theory in Whisper-1.1**
  - Statistical analysis to estimate calculational margin (CM)
  - Based on most-similar benchmarks selected
- **Generalized Linear-Least Squares Method in Whisper-1.1**
  - Use benchmark sensitivities & cross-section covariance data
  - Estimate the MOS for nuclear data uncertainties
  - Based on entire catalogue of benchmarks
- **Machine Learning using Decision Trees (not in Whisper-1.1)**
  - Use benchmark sensitivities (no nuclear data covariance data used)
  - Regression used to predict target function
  - Based on entire catalogue of benchmarks

Bias Prediction Methods
Extreme Value Theory (1)

- **Whisper uses a nonparametric statistical approach to determining the calculational margin (bias + bias uncertainty)**
  - Does not rely on assumption that ($k_{calc} – k_{bench}$) is normally distributed for the set of benchmarks
  - Based on Extreme Value Theory (EVT)
    - The addition of less-relevant benchmarks cannot reduce the calculational margin
    - Irrelevant benchmarks (i.e., low c_k) will not non-conservatively affect results
    - Accounting for weighting avoids overly conservative calculational margin
- **Whisper uses EVT to find the value of a calculational margin that bounds the worst-case bias to some probability of a weighted population**
- **Notes:**
  - There is the fundamental assumption that for a single benchmark, the bias for that benchmark is normally distributed, according to the experimental uncertainty & Monte Carlo statistics
  - There is no assumption of normality across the collection of benchmarks, however.

Bias Prediction Methods
Extreme Value Theory (2)

- **Let** $\beta_J = k_{calc\ J} - k_{bench\ J}$ **and** $\sigma^2_J = \sigma^2_{bench\ J} + \sigma^2_{calc\ J}$
  - For convenience, the X_J below are opposite in sign to β_J
- **For a set of N benchmarks, let X_J be a random variable normally distributed about β_J with uncertainty σ_J. The cumulative distribution function (CDF) for X_J is**

$$F_J(x) = \mathrm{Prob}(X_J < x) = \frac{1}{\sqrt{2\pi}\cdot\sigma_J}\int_{-\infty}^{x}\exp\left[-\frac{1}{2}\left(\frac{y+\beta_J}{\sigma_J}\right)^2\right]dy = \frac{1}{2}\left[1+\mathrm{erf}\left(\frac{x+\beta_J}{\sqrt{2\sigma_J^2}}\right)\right]$$

  - Note: $+\beta_J$, due to opposite sign
- **Let the random variable X be the maximum (opposite-signed) bias for the benchmark collection:**

$$X = \max\{ X_1, \ldots, X_N \}$$

- **The cumulative distribution function (CDF) for X is**

$$F(x) = \mathrm{Prob}(X \le x) = \prod_{J=1}^{N} F_J(x)$$

Bias Prediction Methods
Extreme Value Theory (3)

- **When benchmarks are weighted, the following form is used for $F_J(x)$**

$$F_J(x) = (1 - w_J) + \frac{w_J}{2}\left[1 + \mathrm{erf}\left(\frac{x + \beta_J}{\sqrt{2\sigma_J^2}}\right)\right]$$

- **For all benchmarks J = 1, ..., N, Whisper computes**
  - Benchmark weight, w_J
  - Bias, β_J
  - Bias uncertainty, σ_J
- **Those quantities & the weighted $F_J(x)$ determine F(x):** $F(x) = \prod_{J=1}^{N} F_J(x)$
- **Whisper determines the calculational margin (bias + bias uncertainty) by numerically solving:**

  F(CM) = .99 (.99 is default, user opt)

Note: CM is the calculational margin that bounds the worst-case benchmark bias & bias uncertainty with probability .99 (default)

Bias Prediction Methods
Extreme Value Theory (4)

- **Bias & bias uncertainty**

  USL = 1 - CM - MOS

  = 1 + bias - bias-uncertainty - $\Delta_{\text{non-conserv}}$ - MOS

  - ANSI/ANS-8.24:

    "Individual elements (e.g., bias and bias uncertainty) of the calculational margin need not be computed separately. Methods may be used that combine the elements into the calculational margin."

- **Whisper computes CM by numerically solving F(CM) = .99**

- **Whisper computes bias & bias uncertainty numerically as:**

$$\text{bias} = -\int_{-\infty}^{\infty} x \cdot f(x)\,dx = -\int_{-\infty}^{\infty} xF(x)\sum_{J=1}^{N} w_J \frac{f_J(x)}{F_J(x)}\,dx$$

$$\sigma_{\text{bias}} = CM + \text{bias}$$

Note: If the bias is non-conservative (positive), then the CM is adjusted so that no credit is taken for non-conservative bias (CM = CM + bias)

Bias Prediction Methods
GLLS Method (1)

- **Goal is to minimize discrepancies between simulated and measured k_{eff} while constrained by the nuclear data covariance matrices**

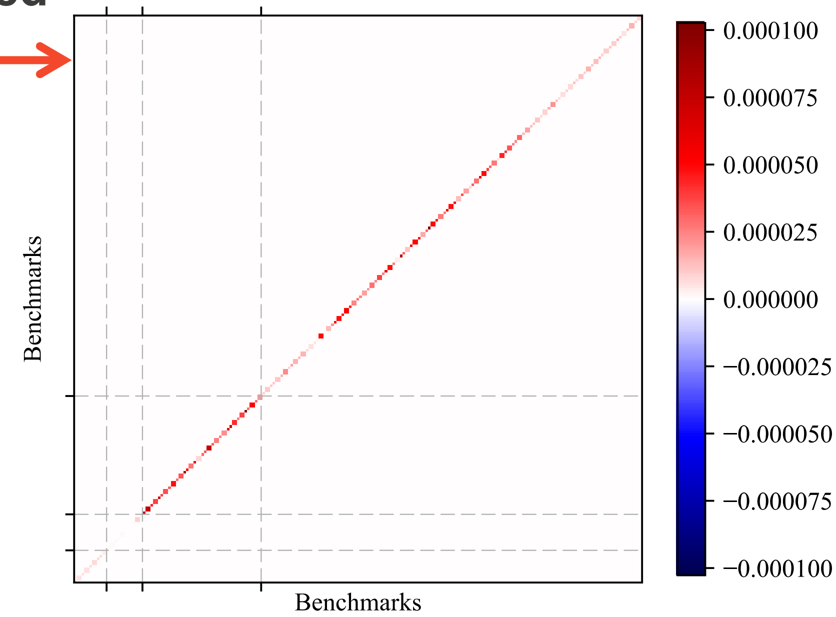
$$\chi^2 = \mathbf{\Delta k}^T \mathbf{C}_{\mathbf{mm}}^{-1} \mathbf{\Delta k} + \mathbf{\Delta}\sigma^T \mathbf{C}_{\sigma\sigma}^{-1} \mathbf{\Delta}\sigma$$

$\mathbf{\Delta k} =$ **Discrepancy between posterior (adjusted) and measured k_{eff}**

$\mathbf{C}_{\mathbf{mm}} =$ **Covariance matrix of measured benchmarks**

$\mathbf{\Delta}\sigma =$ **Difference between prior and posterior nuclear data**

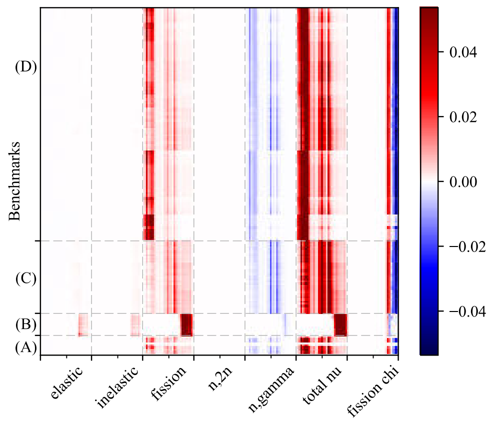
$\mathbf{C}_{\sigma\sigma} =$ **Covariance matrix of nuclear data (previous slide)**



Bias Prediction Methods
GLLS Method (2)

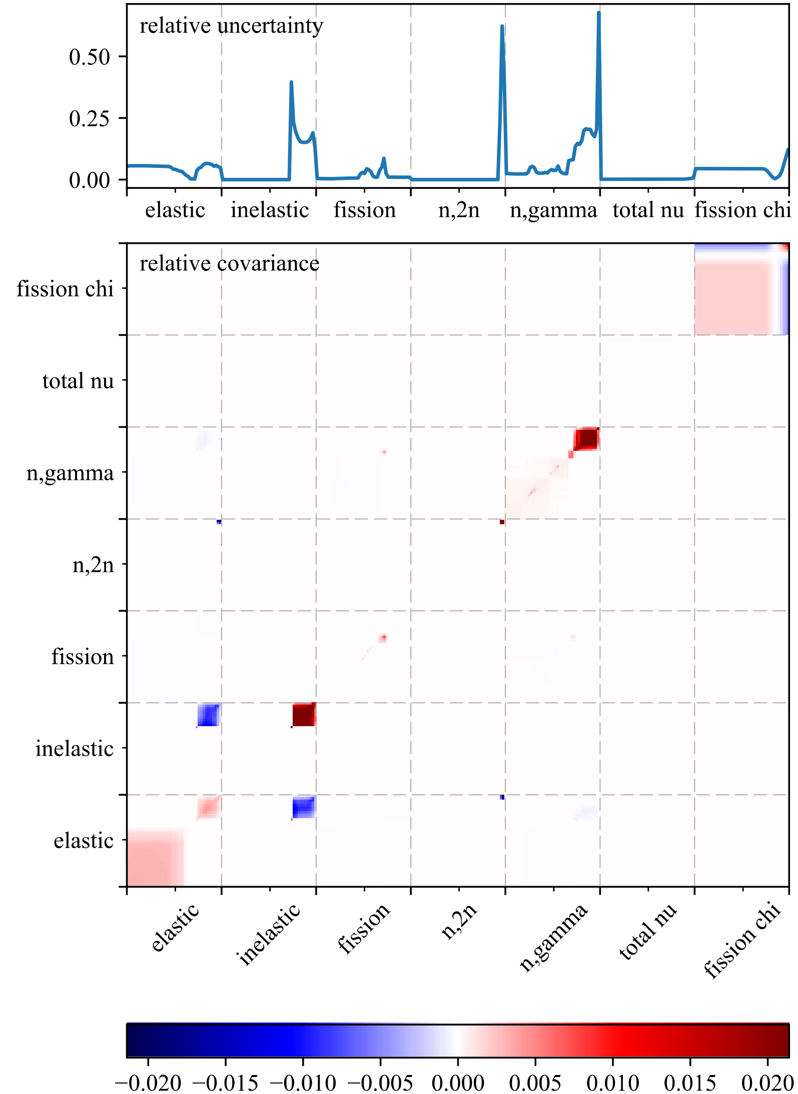
- **With the sensitivity profiles defining how each benchmark k_{eff} changes with respect to the nuclear data,**

$$S_{i,j} = \frac{\sigma_j}{k_i}\frac{\partial k_i}{\partial \sigma_j}$$



- **Linear error propagation, "sandwich" rule,**

$$\mathbf{C_{kk}} = \mathbf{S}\mathbf{C}_{\sigma\sigma}\mathbf{S}^T$$



Bias Prediction Methods GLLS Method (3)

- Covariance of the prior discrepancies

$$\mathbf{C_{dd}} = \mathbf{C_{kk}} + \mathbf{C_{mm}}$$

- The final results of the GLLS minimization process, improved agreement between simulation and measurement

$$\mathbf{\Delta k} = \mathbf{C_{mm}} \mathbf{C_{dd}^{-1}} \mathbf{d}$$

- Reduced nuclear data induced uncertainties in benchmarks, used to compute the portion of the margin of subcriticality from to nuclear data

$$\mathbf{C'_{kk}} = \mathbf{S} \mathbf{C}'_{\sigma\sigma} \mathbf{S}^T$$

- Nuclear data and uncertainty adjustments,

$$\mathbf{\Delta}\sigma = -\mathbf{C}_{\sigma\sigma} \mathbf{S}^T \mathbf{C_{dd}^{-1}} \mathbf{d} \qquad \mathbf{C}'_{\sigma\sigma} = \mathbf{C}_{\sigma\sigma} - \mathbf{C}_{\sigma\sigma} \mathbf{S}^T \mathbf{C_{dd}^{-1}} \mathbf{S} \mathbf{C}_{\sigma\sigma}$$

Bias Prediction Methods
Machine Learning (1)

- Machine learning algorithms can be used to find "hidden" patterns in data that are not necessarily obvious
- Can be used to classify data
- In this case, we want to "predict" something: given x, what is f(x)

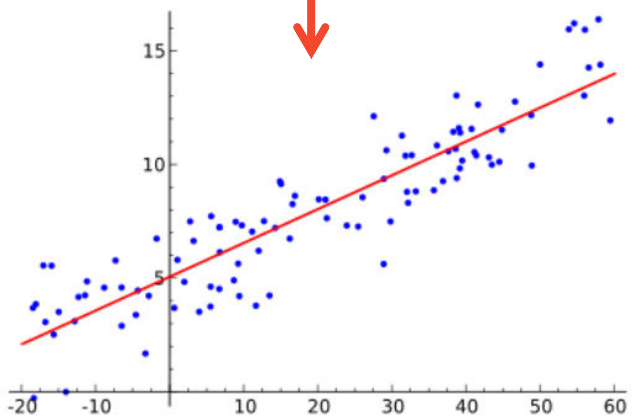


Image obtained from Wikipedia's Linear Regression page

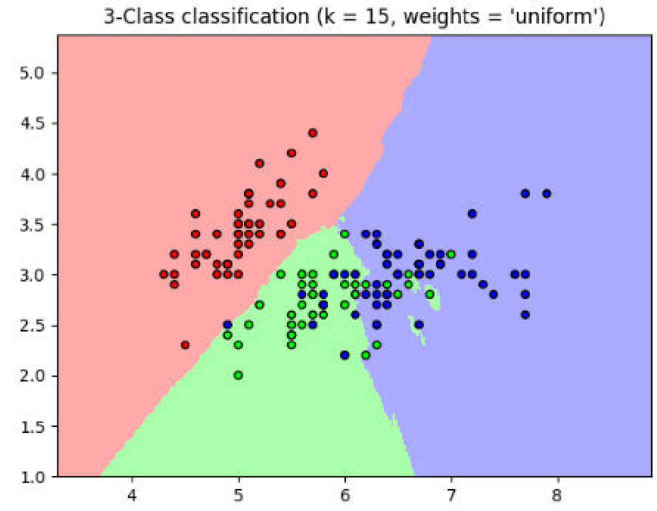
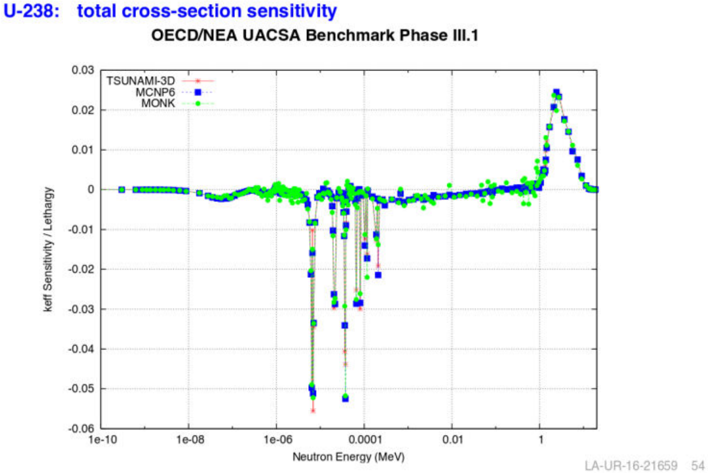


Image obtained from https://sebastianraschka.com/faq/docs/evaluate-a-model.html

- Some nomenclature: features = x, labels = f
- The objective is to predict bias

Bias Prediction Methods
Machine Learning (2)

- **Prediction of Bias using Sensitivity Profiles**
  - Sensitivity profiles are readily available, $S_{i,j}$
  - Bias known for Whisper benchmarks, $\beta_i = k_{calc\ i} - k_{bench\ i}$
  - Goal: predict bias, $\beta_i(S_{i,j})$



- **Decision Trees**
  - A tree-like model of decisions based on the features
  - All features are considered to split the data
  - Splits are chosen to maximize a cost function (i.e. mean-square error)
  - More important features are found near the top

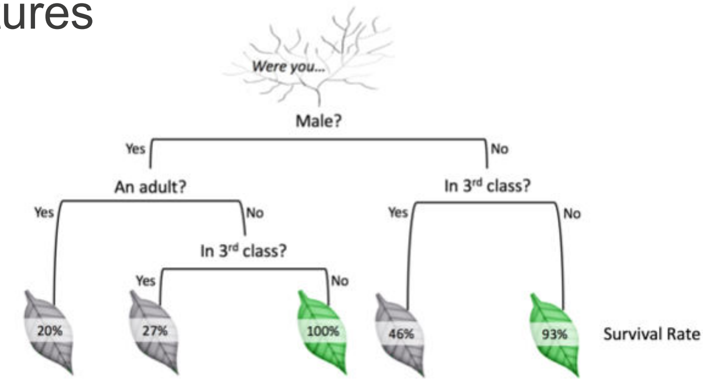


Image obtained from
https://algobeans.com/2016/07/27/decision-trees-tutorial

- **Random Forest (random subset of features)**
- **Adaboost (iteratively improve poor predictors)**

Generalized Results (1)

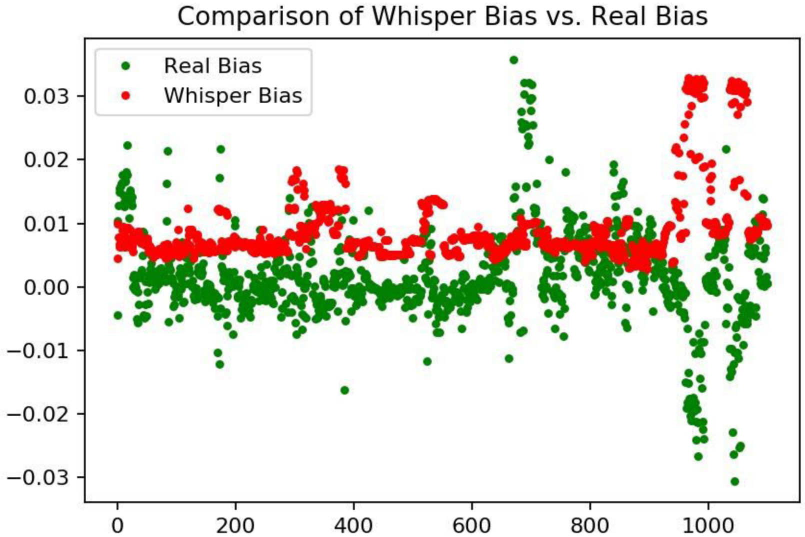
- **Use Whisper benchmark suite for which the bias is known**

- **Whisper results**
  - Remove each benchmark and use as application
  - Bias from extreme value theory

- **GLLSM results**
  - Apply method constrained by covariance data
  - Bias = prior – posterior k_{eff}

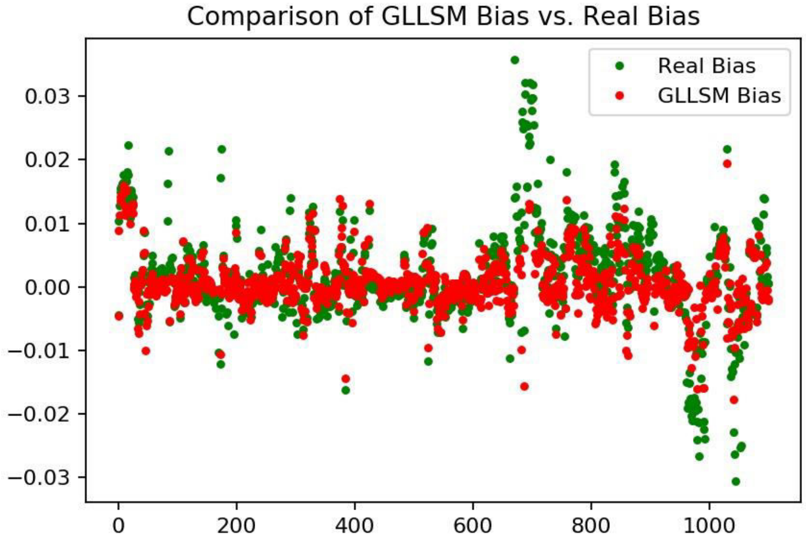
- **Machine Learning results**
  - Train bias function using sensitivity profiles
  - Bias = $\beta(S_j)$

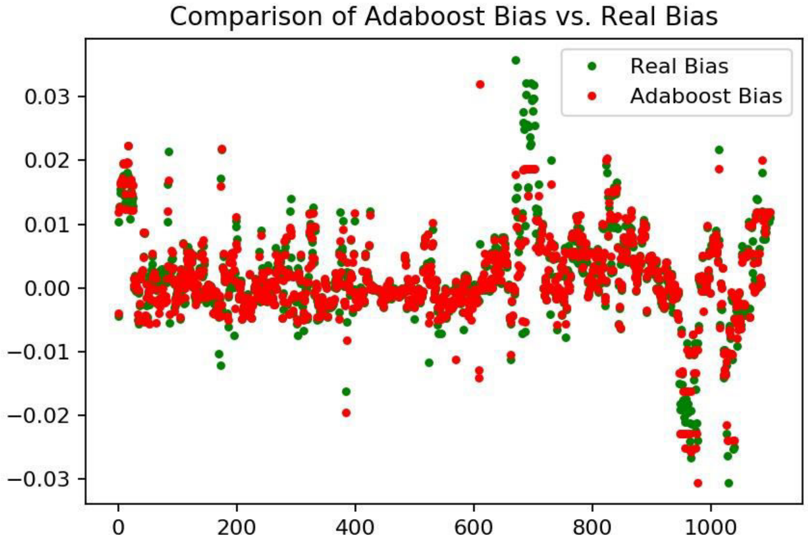
Generalized Results (2)



Bias Accuracy Metrics

Model	RMSE	MAE
Whisper	0.01329	0.00906
GLLSM	0.00959	0.00645
R. Forest (I)	0.00499	0.00350
AdaBoost (I)	0.00498	0.00352
R. Forest (D)	0.00572	0.00397
Adaboost (D)	0.00537	0.00374





Application Toward SG-2 Blind Benchmark Study (1)

- **Whisper-1.1 Results**
  - For all cases
    - Bias & Bias Unc. ~1%
    - MOS_{ND} ~0.15% (1-sigma)
    - $max(c_k)$ 0.79-0.92
    - best matches generally PCM-2, MCI-5, PCI-1
- **Does not include BFS benchmarks**

Case	Bias	Bias Unc.	MOS_{ND}	$\max(c_k)$
C01	0.00911	0.01190	0.00129	0.87696
C02	0.01050	0.01116	0.00134	0.83712
C03	0.00966	0.01097	0.00150	0.86452
C04	0.00912	0.01177	0.00142	0.86187
C05	0.01049	0.01116	0.00137	0.82235
C06	0.00897	0.01100	0.00165	0.92323
C07	0.00919	0.01204	0.00111	0.90301
C08	0.01044	0.01122	0.00128	0.88176
C09	0.00962	0.01105	0.00158	0.89738
C10	0.00806	0.01136	0.00119	0.86025
C11	0.00996	0.01112	0.00133	0.79246
C12	0.00894	0.01099	0.00153	0.87734
C13	0.00946	0.01185	0.00138	0.90423
C14	0.01054	0.01111	0.00132	0.85682
C15	0.00970	0.01085	0.00141	0.87708

Application Toward SG-2 Blind Benchmark Study (2)

- **Comparison of Bias Estimates**
  - For most cases
    - EVT Bias > GLLSM Bias
    - Additional conservatism from EVT?
    - Remember EVT calculates CM, bias & bias unc. inferred from CM
  - For all cases
    - EVT bias > ML bias
    - GLLSM bias > ML bias
- **GLLSM and ML bias uncertainties not reported / computed**
- **Does not include BFS benchmarks**

Case	EVT	GLLSM	ML
C01	0.00911	0.01124	0.00763
C02	0.01050	0.00828	0.00583
C03	0.00966	0.00741	0.00544
C04	0.00912	0.01152	0.00785
C05	0.01049	0.00875	0.00612
C06	0.00897	0.00735	0.00537
C07	0.00919	0.00885	0.00682
C08	0.01044	0.00705	0.00589
C09	0.00962	0.00640	0.00553
C10	0.00806	0.01161	0.00770
C11	0.00996	0.00836	0.00611
C12	0.00894	0.00752	0.00579
C13	0.00946	0.01150	0.00766
C14	0.01054	0.00854	0.00571
C15	0.00970	0.00713	0.00489

Conclusions & Future Work

- Investigated the Whisper-1.1 EVT and GLLSM methods
- Proposed a new use of ML methods, attempting to best estimate bias
- For the ML methods, used nuclear data sensitivity profiles as features [x] to predict bias [f(x)]
- Generalized results for Whisper-1.1 benchmarks suggest the ML algorithms perform most accurately compared to EVT and GLLSM in predicting bias
- Need to predict bias of BFS benchmarks to observe how well these methods perform for more relevant benchmarks
- Investigate adding BFS benchmarks to Whisper-1.1 to better cover these blind application cases and re-train ML algorithms

Questions?

Extra Slides (1)

- **From the machine learning methods, feature importances can be used to identify what nuclear data is cause for poor bias predictions**

Thermal (0 - 0.625 ev)	Intermediate (1.0 ev - 0.1 Mev)	Fast (0.4 Mev - 20 Mev)
6000.80c n,gamma, 0.014562	**92233.80c n,gamma, 0.018457**	**92233.80c fission, 0.015264**
92233.80c total nu, 0.011437	**92233.80c fission, 0.015724**	**92233.80c inelastic, 0.013543**
92233.80c n,gamma, 0.010641	**92233.80c total nu, 0.012844**	**92233.80c n,gamma, 0.012739**
92234.80c n,gamma, 0.009479	**92234.80c n,gamma, 0.011945**	**92233.80c total nu, 0.012644**
1001.80c n,gamma, 0.009069	**94239.80c n,gamma, 0.011687**	9019.80c inelastic, 0.010355
poly.20t inelastic, 0.008879	6000.80c n,gamma, 0.008924	6000.80c elastic, 0.009997
be.20t elastic, 0.008204	**94239.80c total nu, 0.008325**	**92233.80c fission chi, 0.008758**
94239.80c n,gamma, 0.007522	**94239.80c fission, 0.008208**	**92234.80c total nu, 0.008494**
94239.80c fission, 0.007427	6000.80c elastic, 0.007817	**92234.80c fission, 0.008008**
9019.80c n,gamma, 0.007201	**92232.80c total nu, 0.006668**	1001.80c elastic, 0.007938

Extra Slides (2)

- From the machine learning methods, the comparison between the computed feature importances and the nuclear data uncertainties is very suggestive (remember nuclear data covariances are not used…)

^{233}U uncertainties and importances