Model Error Estimation and Uncertainty Quantification of Machine Learning Interatomic Potentials

Khachik Sargsyan,
Logan Williams, Habib N. Najm

Sandia National Laboratories, Livermore, CA

Error control in first-principles modelling
CECAM-EPFL, Lausanne, Switzerland
June 20-24, 2022
Acknowledgements

- Aidan Thompson, Mary Alice Cusentino, Mitchell Wood, Ember Sikorski (SNL), Katherine Johnston (SNL, U Washington)
- DOE, Office of Science,
  - Fusion Energy Sciences (FES)
  - Advanced Scientific Computing Research (ASCR)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA-0003525.
Outline

- UQ for machine learning interatomic potentials (MLIAP)
  - ... for uncertainty propagation
  - ... for active learning
  - ... for model selection

- Bayesian approach
  - More focus on linear regression models:
    Spectral Neighbor Analysis Potential (SNAP)
  - Importance of noise model, embedded model error construction
  - Relation to variational inference
ML Interatomic Potentials (MLIAP): supervised ML

- Partition the interatomic interaction energy into individual contributions of the atoms: \( E_{\text{total}} = \sum_{i=1}^{N} E_i \)

- Assume flexible functional forms of each such contribution:
  - Function of positions of the neighboring atoms
  - \( O(100) \) parameters

- Require the energy, forces and/or stresses predicted by a MLIAP to be close to those obtained by a quantum mechanical model on some atomic configurations (a.k.a. training set)
MLIAP - desired features

- Good input descriptors
- Accurate, fast-to-evaluate, analytic derivatives
- High-dimensional, flexible functional form
- Transferable/generalizable to unseen atomic configurations
- Account for physics:
  - invariant with respect to translation, rotation, and reflection of the space, and also permutation of chemically equivalent atoms
- Locality (depend on surrounding atoms only within a finite cut-off radius), but remain smooth with respect to atoms entering and leaving the local neighborhood
- **Equipped with uncertainty estimate**
  - for active learning, for MD propagation, ...
Enabling parametric fits with uncertainties

\[ y \approx f_c(x) \]
Focus on SNAP (Left end of the figure)

<table>
<thead>
<tr>
<th>Descriptors</th>
<th>Complex (e.g. bispectrum)</th>
<th>Simple (e.g. cartesian)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAP Models</td>
<td>Linear Regression</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>UQ Methods</td>
<td>Bayesian</td>
<td>Approx.</td>
</tr>
<tr>
<td></td>
<td>Analytical</td>
<td>Variational</td>
</tr>
</tbody>
</table>


\[
f(q) = \sum_{k=0}^{K} c_k B_k(q)
\]

- Linear expansion in parameters \(c\).
- Bayesian inference: both MCMC and analytical posterior PDFs are feasible.
Given a model \( f(x, c) \) and data \( y_i = y(x_i) \), calibrate parameters \( c \).

- Linear model \( f(x, c) = Bc \) with coefficients \( c \)
- NN model \( f(x, c) = NN_c(x) \) with weights/biases \( c \)

Bayesian least-squares fit:

\[
p(c|y) \propto p(y|c)p(c) \propto \prod_{i=1}^{N} \exp \left(-\frac{(f(x_i, c) - y_i)^2}{2\sigma^2}\right)
\]

... corresponding data model \( y_i = f(x_i, c) + \sigma \epsilon_i \sim \mathcal{N}(0,1) \)

Exact answer for linear models: \( c \sim \mathcal{N} \left( (B^T B)^{-1} B^T y, \sigma^2 (B^T B)^{-1} \right) \)
SNAP uncertainty with Tantalum data set

\[ f(q) = \sum_{k=0}^{K} c_k B_k(q) \]

- Employed FitSNAP https://github.com/FitSNAP/FitSNAP

Assumptions baked in likelihood form are crucial!
- i.i.d. gaussian noise with constant \( \sigma \) is not well founded.

Analytical results:
Mean fit ok, const \( \sigma \) assumption may need rethinking
Elephant in the room: model is assumed to be *the* correct model behind data

\[ y_i = f(x_i, c) + \sigma_i \epsilon_i \]

One gets biased estimates of parameters \( c \) (crucial if the model is physical, and/or \( c \) is propagated through other models)

More data leads to overconfident predictions (we become more and more certain about the wrong values of the data)

More evident when there is no (observational/experimental) data error: e.g. DFT is data, and MLIAP is model
Posterior pushed-forward uncertainty does not capture true discrepancy

\[
y(x) = \sin^4(2x - 0.3)
\]

Cubic fit

\[
y_i \approx \sum_{k=0}^{3} c_k B_k(x)
\]

More data leads to overconfident prediction
Capturing model error in data model (a.k.a. likelihood)

\[ y_i = f(x_i, c) + \delta(x_i) + \sigma_i \epsilon_i \]

**External correction** (Kennedy-O’Hagan):


\[ y_i = f(x_i, c + \delta(x_i)) + \sigma_i \epsilon_i \]

**Internal correction** (embedded model error):

- Allows meaningful usage of calibrated model
- ‘Leftover’ noise term even with no data error
- Respects physics (not too relevant in our context)


- Typically requires uncertainty propagation in the likelihood computation
- For linear regression, we can take some shortcuts (see next)
Embedded Model Error for Linear Regression Models

Conventional (i.i.d. error term):

\[ y_i \approx \sum_{k=0}^{P} c_k B_k(x_i) + \sigma_i \epsilon_i \]

Embed uncertainty in all or selected coefficients:

\[ y_i \approx \sum_{k=0}^{P} (c_k + d_k \xi_k) B_k(x_i) = \sum_{k=0}^{P} c_k B_k(x_i) + \sum_{k=0}^{P} d_k B_k(x_i) \xi_k \]

**Note:**
No formal distinction between internal and external corrections: but the error is now model-informed.
Joint MCMC inference of model parameters and model-error parameters

Conventional:

\[ y_i \approx \sum_{k=0}^{P} c_k B_k(x_i) + \sigma_i \epsilon_i \]

\[ p(c|y) \propto \prod_{i=1}^{N} \exp \left( -\frac{\left( \sum_{k=0}^{P} c_k B_k(x_i) - y_i \right)^2}{2\sigma_i^2} \right) \]

Embedded:

\[ y_i \approx \sum_{k=0}^{P} (c_k + d_k \xi_k) B_k(x_i) = \sum_{k=0}^{P} c_k B_k(x_i) + \sum_{k=0}^{P} d_k B_k(x_i) \xi_k \]

\[ p(c, d|y) \propto p(y|c, d) p(c, d) \]

**Note:** Both likelihood and prior selection are challenging.
Embedded Model Error: Two Approximate Likelihood Options

\[ y_i \approx \sum_{k=0}^{P} (c_k + d_k \xi_k) B_k(x_i) = \sum_{k=0}^{P} c_k B_k(x_i) + \sum_{k=0}^{P} d_k B_k(x_i) \xi_k \]

Option 1: IID

\[
p(c, d | y) \propto \prod_{i=1}^{N} \exp \left( - \frac{\left( \sum_{k=0}^{P} c_k B_k(x_i) - y_i \right)^2}{2 \sum_{k=0}^{K} d_k^2 B_k^2(x_i)} \right)
\]

Option 2: ABC

\[
p(c, d | y) \propto \prod_{i=1}^{N} \exp \left( - \frac{\left( \sum_{k=0}^{P} c_k B_k(x_i) - y_i \right)^2 + \left( \sqrt{\sum_{k=0}^{P} d_k^2 B_k^2(x_i)} - \alpha \right) \left| \sum_{k=0}^{P} c_k B_k(x_i) - y_i \right|^2}{2 \epsilon^2} \right)
\]

**Note:**
Does not have to be MCMC: simply optimize the posterior for \((c, d)\)
Pushed forward predictive uncertainty captures the true discrepancy from the data.

**Synthetic data**

\[ y(x) = \sin^4(2x - 0.3) \]

**Cubic fit**

\[ y_i \approx \sum_{k=0}^{3} c_k B_k(x) \]

- **Classical case**
- **Model error, IID likelihood**
- **Model error, ABC likelihood**

\[ N=100, \text{ Poly order}=3 \]

---

ksargsy@sandia.gov

UQ for MLIAPs

Jun 23, 2022 16 / 25
Uncertainty validation: W-ZrC Dataset

Uncertainty without model error

Uncertainty with model error
Uncertainty validation: two examples

Conventional

Embedded, IID Lik.

Embedded, ABC Lik.

Ta

W-ZrC

ksargsy@sandia.gov
Model Error Wrapup: several challenges and choices

- Embedding type, e.g. additive/multiplicative

\[ y_i \approx \sum_{k=0}^{P} (c_k + d_k \xi_k) B_k(x_i) \quad \text{or} \quad y_i \approx \sum_{k=0}^{P} (c_k + c_k d_k \xi_k) B_k(x_i) \]

- Degenerate (Gaussian) likelihoods: resort to approximate Bayesian computation (ABC) or independent (IID) assumptions
- Difficult posterior PDFs for MCMC, choice of priors for embedding parameters
- Which coefficients to embed the model error in?
- Connect predictive uncertainty and the residual error with an extrapolation metric
- Weighting between energies, forces and stresses
Variational inference is a compromise between Bayesian and Empirical approaches.
**Variational inference in a nutshell**

$$KL(p_1 || p_2) = \int \ln \left( \frac{p_1(x)}{p_2(x)} \right) p_1(x) dx$$

- e.g. Mean-Field Variational Inference (MFVI): ansatz $c \sim \mathcal{N}(\mu, \text{diag}(v))$ and find best $(\mu, v)$, i.e.
- minimize Kullback-Leibler distance to the full Bayesian posterior, $\arg\min_{(\mu, v)} KL(\mathcal{N}(\mu, \text{diag}(v)) || \mathcal{N}(\mu_0, \Sigma))$,
- replaces sampling (MCMC) problem with an optimization problem.
Note the connection between variational inference and embedded model error

- **Variational methods:** \( c \sim N(\mu, \Sigma) \) and optimize \( \mu, \Sigma \).
  - In NN context, this is largely called Bayesian Neural Networks
  - Minimize Kullback-Leibler distance via Stoch. Gradient Descent

- **Embedded model error:** \( c \sim N(\mu, \Sigma) \) and optimize \( \mu, \Sigma \).
  - Minimize Gaussian approximation of output predictions (IID), or
  - Minimize statistics/moment matching criterion (ABC)

**Next:**
Overparameterized linear regression (mimicking NN) challenges mean-field variational inference outside training support.
Polynomial fit: Extrapolation scenario

Order=2

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Extrapolation scenario

Order=3

**Full Bayesian Posterior**
\[ \mathcal{N}(\mu, \Sigma) \]

**Variational Posterior**
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Extrapolation scenario

Order=4

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Extrapolation scenario

Order=5

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Interpolation scenario

Order=2

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/diag(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Interpolation scenario

Order=3

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Interpolation scenario

Order=4

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Polynomial fit: Interpolation scenario

Order=5

Full Bayesian Posterior
\[ \mathcal{N}(\mu, \Sigma) \]

Variational Posterior
\[ \mathcal{N}(\mu, 1/\text{diag}(\Sigma^{-1})) \]

Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
Variational posterior predictions heavily underestimate both interpolative and extrapolative errors, in the overparameterized regimes.
**Summary**

- **Bayesian fit of parameterized MLIAPs**
  - Noise assumptions are crucial

- **Embedded model error**
  - Statistical correction *inside* the model: joint inference of model parameters and the correction
  - Leads to model-driven noise model
  - Meaningful model-error uncertainty capturing the true residual
  - A few shortcuts in linear regression models
  - Choices to make: priors, approximate likelihoods, MCMC sampler, where to embed...

- **Variational inference**
  - Approximate alternative to MCMC for nonlinear, complex models
  - Underestimates the uncertainty for overparameterized models: dangerous when extrapolating!
  - Mechanically similar to embedded model error, except the optimization objective/method (and, potentially, the interpretation!)
Additional Material
Uncertainty Propagation through MD

- PC intro setup; SNAP coefficients form a first order Gauss-Hermite Polynomial Chaos (PC)

\[ E \approx \sum_{k=0}^{P} \left( c_k + d_k \xi_k \right) B_k(x) \]

- Sample SNAP coefficients
- Evaluate MD QoIs
- Build PC for MD QoIs, possibly multilevel/multifidelity
- Evaluate PDF/statistics of QoIs
- Challenges: high-d input, noisy MD simulations
Uncertainty-enabling wrappers over PyTorch modules

Deterministic

torch.nn.module

Probabilistic

wrapper(torch.nn.module)

Option 1: ensemble NN

```python
nn_ens = EnsRegr(torch.nn.module, nens=111)
```

class EnsRegr():
    def __init__(self, nmodule, nens=1, verbose=False):
        self.nmodule = nmodule
        self.nens = nens

Option 2: NN learning with MCMC

```python
nn_mcmc = MCMCREgr(torch.nn.module)
```

class MCMCREgr():
    def __init__(self, nmodule, verbose=False):
        self.nmodule = nmodule
        self.verbose = verbose

Option 3: NN learning with VI

```python
nn_vi = VIRegr(torch.nn.module)
```

class VIRegr():
    def __init__(self, nmodule, verbose=False):
        self.nmodule = nmodule
        self.verbose = verbose

- MCMC struggles with complex NNs; VI underestimates; Ensembles do well
Uncertainty-enabling wrappers over PyTorch modules

MCMC struggles with complex NNs; VI underestimates; Ensembles do well
Model error embedding


MLIAPs


Active learning


Active learning for MLIAPs