Bayesian Model Mixing via Taweret

Matthew T. Pratola - today's spokesperson(?) for:

K. Ingles, D. Liyanage, A. C. Semposki, J. C. Yannotty and J. P. O'Neal

twitter: @MattPratola
web: www.matthewpratola.com

Tuesday December 17th, 2024

BAND: Bayesian Analysis of Nuclear Dynamics

http://www.bandframework.github.io

- Integrate diverse software tools into a cohesive framework (Python/Taweret).
  - Physics simulators, Emulation, Calibration, Bayesian Model Mixing (BMM)
- BMM Frameworks
  - Moment (mean) mixing
    - Gaussian Process based mixing (Semposki et al.)
    - Tree-based mixing (Yannotty et al.)
  - Density mixing
    - Linear density mixing (Liyanage et al.)
    - Approximate Likelihood mixing (Ingles et al.)

Taweret

JOSS
The Journal of Open Source Software

Taweret: a Python package for Bayesian model mixing

K. Ingles[1*¶], **D. Liyanage**[2*], **A. C. Semposki**[3*], **and J. C. Yannotty**[4*]

1 Illinois Center for Advanced Study of the Universe & Department of Physics, University of Illinois Urbana-Champaign, USA **2** Department of Physics, The Ohio State University, USA **3** Department of Physics and Astronomy & Institute of Nuclear and Particle Physics, Ohio University, USA **4** Department of Statistics, The Ohio State Univeristy, USA **¶** Corresponding author ***** These authors contributed equally.

General BMM Framework

- A process, Y, is observed, which depends on some p-dimensional inputs $\mathbf{x}$.
- This observation is of some underlying phenomena of interest, $f_\dagger(\mathbf{x})$.
- A simple model that we often adopt linking the observations to the underlying phenomena is

$$Y|\mathbf{x} \sim N\left(f_\dagger(\mathbf{x}), \mathbf{c}(\cdot)\right)$$

- Problem: we don't know $f_\dagger$. In fact, we don't even know a good approximation of it. But we do know how to construct high quality *local* approximations.

BMM – Mean Mixing

- Assume we have K models $\mathcal{M}_1, \ldots, \mathcal{M}_K$ with means $f_1, \ldots, f_K$.
- Average the mean predictions of these models,

$$E[Y|x] = \sum_{k=1}^{K} w_k(\mathbf{x}) f_k(\mathbf{x})$$

- The $w_k(x)$'s are input-varying weight functions.

BMM – Density Mixing

- Assume we have K models $\mathcal{M}_1, \ldots, \mathcal{M}_K$ with predictive densities $p_1, \ldots, p_K$.
- Construct the overall predictive density as the weighted combination,

$$p[Y_0|x_0] = \sum_{k=1}^{K} w_k(\mathbf{x}_0) p_k(Y_0|\mathbf{x}_0, Y)$$

- The $w_k(x) \in [0,1]$, s.t. $\sum_k w_k(\mathbf{x}) = 1$ for any $\mathbf{x}$ are simplex weights.

BMM Methods in Taweret

- *Bivariate Linear Mixing*†
  - Mean and Density methods available
  - Supports 1-dimensional input
  - Supports multiple outputs
  - Supports mixing 2 models
- Weight functions:
  - step: $H(\beta_0 - x)$
  - sigmoid: $exp\left((x - \beta_0)/\beta_1\right)$
  - asymmetric 2-step: $\alpha H(\beta_0 - x) + (1 - \alpha)H(\beta_1 - x)$

† Liyanage, 2023

BMM Methods in Taweret

- *Multivariate Mixing*[††]
  - Mean method available
  - Precision-based weighting in a Gaussian likelihood
  - Supports 1-dimensional input
  - Supports 1-dimensional output
  - Supports mixing K models

- Weight functions:
  - Each model assumed Gaussian form $N\left(f_k(x), \sigma^2(x)\right)$
  - Assumes variance information available
  - Mean and variance functions can be modeled using variety of ML tools, such as Gaussian Processes or Neural Networks.

[††] Semposki et al. (2022a), Semposki et al. (2022b)

BMM Methods in Taweret

- *Multivariate Mixing*††
  - Mean method available
  - Precision-based weighting in a Gaussian likelihood
  - Supports 1-dimensional input
  - Supports 1-dimensional output
  - Supports mixing K models

- Weight functions:
  - Mixed posterior predictive has Gaussisan form

$$N(f_\dagger(x), Z_p^{-1}(x))$$

where $f_\dagger(x) = \frac{1}{Z_p(x)} \sum_{k=1}^{K} \frac{1}{\sigma_k^2(x)} f_k(x)$ and $Z_p(x) = \sum_{k=1}^{K} \frac{1}{\sigma_k^2(x)}$
  - Induced weight functions $w_k(x)$ arise from ratio of precisions.
  - Mean and variance functions can be modeled using variety of ML tools, such as Gaussian Processes or Neural Networks.

†† Semposki et al. (2022a), Semposki et al. (2022b)

BMM Methods in Taweret

- *BART-based Mixing*[†††]
  - Mean method available
  - "Fully Bayesian" learning of weight functions using tree priors
  - Supports > 1 input
  - Supports 1-dimensional output
  - Supports mixing K models

- Weight functions:
  - Tree-basis weight functions, $w_k(\mathbf{x}) = \sum_{j=1}^{m} g_k(\mathbf{x}; T_j, M_j)$ for $k = 1, \ldots, K$ assigned BART priors.
  - Form of prior can support informative prior when some knowledge of variance functions are known, or uninformative prior.
  - Regularization prefers weights on $[0, 1]$ but can deviate to capture discrepancy.

[†††] Yannotty et al. (2023), Yannotty et al. (2024)

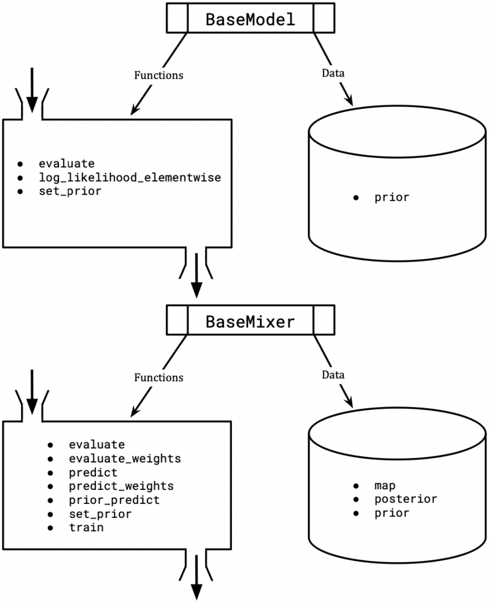
What about those weights?

- Simplex constraint: $w_k(\mathbf{x}) \in [0, 1] \; \forall \; k$ s.t. $\sum_k w_k(\mathbf{x}) = 1$ for any $\mathbf{x}$.
- At any input setting $\mathbf{x}$ there exists a linear combination of models that adequately represents $f_\dagger(\mathbf{x})$.
- What if this is false at some $\mathbf{x}_\star$? The BMM prediction at $\mathbf{x}_\star$ will be poor.

What about those weights?

- Unconstrained weights: $w_k(\mathbf{x})$'s can be any real number, informative prior prefering $w_k(\mathbf{x}) \in [0, 1]$.
- Diagnose inadequate model set in post-processing.
  - e.g. via $\sum_k w_k(\mathbf{x})$
  - e.g. via decomposing $w_k(\mathbf{x})$'s via a constrained and unconstrained projection.

Adding to Taweret



Taweret Workflow

- New physics models are implemented by building off the abstract `BaseModel` class.
  - Essentially, you implement access to a given f_k.
- New model mixing techniques are implemented by building off the abstract `BaseMixer` class.
  - Connects arbitrary model-mixing methods into Taweret.
  - Presents a consisent user interface for training and prediction.

Taweret Workflow – Doing Science

- Three items required:
  - Observations ((X, Y)'s) of the real-world process we are trying to learn about from our models.
  - A set of models $\{\mathcal{M}_1, \ldots, \mathcal{M}_\mathcal{K}\}$ to model the real-world process.
  - Selecting the desired mixing method to combine these two sources of information.

Taweret Workflow – Doing Science

- When doing a new science problem:
  - Implement the models, say $f_1, \ldots, f_K$, by following `BaseModel` (i.e. K model objects to implement).
  - Load the data.
  - Run the mixing method.
  - Interpret the fitted mixed model.

Taweret

Code Blame 403 lines (317 loc) · 10.9 KB

```python
import numpy as np
from scipy.special import factorial
from Taweret.core.base_model import BaseModel

# Polynomial Class Functions


class polynomal_model(BaseModel):
    '''
        Polynomial models class. Used to define a function of the form

        .. math::
            f(x) = c(x-a)^p + b
    '''

    def __init__(self, a=0, b=0, c=1, p=1):
        '''
            Parameters:
            -----------
            :param float a: center parameter.
            :param float b: shift parameter.
            :param float c: scale parameter.
            :param float p: power parameter.

        '''
        self.a = a
        self.b = b
        self.c = c
        self.p = p

    def evaluate(self, x):
        '''
            Evaluate the polynomial at a grid of x's. The standard deviation
            output is set to 1 by default.

            Parameters:
            -----------
            :param np.ndarray x: design matrix.
```

Taweret – Future Directions

- K. Ingles new work on additional density mixing methods, approximate likelihood.
- Tree-based approach for density mixing.
- Extending Multivariate method for higher-dimensional input spaces and multiple outputs.
- All-in-one calibration and mixing.
- Correlated models.

Taweret – Key References

Liyanage, D. (2023). Multifaceted study of ultrarelativistic heavy ion collisions. Ph.D. Thesis, The Ohio State University.

Semposki, A.C., Furnstahl, R. J. and Phillips, D.R. (2022a). Interpolating between small and large-g expansions using Bayesian mdoel mixing. Physical Revieiw C, 106(4), 044002.

Semposki, A.C., Furnstahl, R.J. and Phillips, D.R. (2022b). SAMBA: SAndbox for Mixing using Bayesian Analysis.

Yannotty, J.C., Santner, T.J., Furnstahl, R.J. and Pratola, M.T. (2024). Model mixing using Bayesian additive regression trees. Technometrics, 66(2), 196–207.

Yannotty, J.C., Santner, T.J., Li, B. and Pratola, M.T. (2024). Combining Climate Models using Bayesian Regression Trees and Random Paths. arXiv:2407.13169.

Ingles, K., Liyanage, D., Semposki, A.C. and Yannotty, J.C. Taweret: a Python package for Bayesian model mixing. Journal of Open Source Software, 9(97), 6175.

Taweret Github: https://github.com/bandframework/Taweret