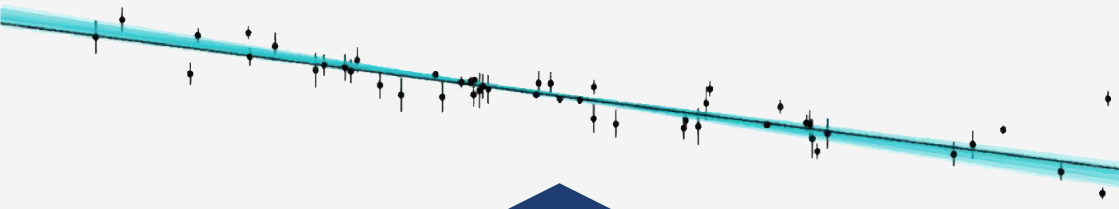
UNIVERSITY OF SURREY



TOWARDS A CROSS-RESEARCH PLATFORM FOR HOSTING BAYESIAN DATA-FITTING TOOLS

Conference Proceedings
July 2021

SPONSOR

Institute of Advanced Studies (IAS)

The Institute of Advanced Studies (IAS) at the University of Surrey sponsors workshops and Fellowships at the 'cutting edge' of science, engineering, social science and the humanities. Through this scheme the Institute fosters interdisciplinary collaborations and encourages a flow of international scholars to visit, enjoy their stay at Surrey and leave behind excellent ideas and innovations.

ias.surrey.ac.uk

Organisers:

Dr Payel Das, University of Surrey, Department of Physics

Dr Alex Shestopaloff, Alan Turing Institute and Queen Mary University of London

Prof David Lloyd, University of Surrey, Department of Mathematics

INTRODUCTION

Scientists and engineers enhance their understanding of the world by fitting models to data. The model parameters are however inherently uncertain due to observational errors in the data, and structural uncertainties in the model. Bayesian sampling methods offer an approach to quantifying the uncertainty in model parameters by inferring the full posterior probability distribution of the model as a function of its parameters. In static systems, the models are usually quick to run for each set of parameters. Real systems such as interacting galaxies, the weather, or flu epidemics often change rapidly with time, requiring more complex models. Applying Bayesian methods to these problems can be computationally prohibitive.

A virtual one-day workshop that ran on Wednesday 14th July, brought together 30+ stakeholders in both research and industry to facilitate knowledge exchange in the quantification of uncertainty in complex data-fitting problems and methods for Bayesian sampling and optimization.

There was a combination of pedalogical talks and research talks in two sessions:

1. Complex data-fitting problems and uncertainty analysis
2. Bayesian sampling and optimization

Three keynote speakers gave the pedagogical talks:

1. Introduction to uncertainty and sequential data assimilation (Dr Naratip Santitissadeekorn from the University of Surrey)
2. The essentials of Markov Chain Monte Carlo (Dr Alex Shestopaloff from The Alan Turing Institute and Queen Mary University)
3. An Introduction to Nested Sampling (Dr Josh Speagle from Harvard University)

Here are the conference proceedings from the workshop.

VIDEO RECORDINGS

Video recordings of the presentations can be accessed via **Youtube**.

Naratip Santitissadeekorn

INTRODUCTION TO UNCERTAINTY AND SEQUENTIAL DATA ASSIMILATION

UNCERTAINTY QUANTIFICATION

From the Bayesian viewpoint, uncertainty is subjective, which typically arises from lack of complete knowledge of the phenomenon we wish to understand. The essence of Bayesian uncertainty quantification is to use a probability distribution to describe our uncertainty, which will be reassigned a new probability distribution in light of available evidences or data. The basic Bayes's rule can be applied to assimilate the available data for the probability reassignment. In the next section, we will consider a recursive method for data assimilation.

SEQUENTIAL DATA ASSIMILATION (DA)

The **Sequential Data Assimilation** uses the Bayes's rule to recursively incorporate information contained in a data stream y_k to update uncertainty of unobserved state x_k, where k is a discrete time step. The main elements of this problem is the state-space model of x and y:

Hidden-Markov model: $\mathbf{x}_k \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$

Observation model: $\mathbf{y}_k \sim p(\mathbf{y}_k \mid \mathbf{x}_k)$

An example of a state-space model is the random walk model:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \rightarrow p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k-1}, \mathbf{Q})$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{r}_k, \qquad \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \rightarrow p(\mathbf{y}_k \mid \mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R})$$

Based on these assumptions, a recursive learning algorithm can be developed to compute the probability density $p(x_k \mid y_{1:k})$, where $y_{1:k}$ is the information up to time step k. Some key assumptions are required to enable a recursive scheme:

- Markov property of states (M)

$$p(\mathbf{x}_k \mid \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \Rightarrow p(\mathbf{x}_{0:T}) = p(\mathbf{x}_0) \prod_{k=1}^{T} p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$$

- Conditional independence of observations (C)

$$p(\mathbf{y}_k \mid \mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k \mid \mathbf{x}_k) \Rightarrow p(\mathbf{x}_{1:T} \mid \mathbf{x}_{0:T}) = \prod_{k=1}^{T} p(\mathbf{y}_k \mid \mathbf{x}_k)$$

continued ▶

The sequential DA performs two main steps:

1. Prediction step: Given $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$, we can apply the Chapman-Kolomogrov equation and use (M) to show that

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1}$$

2. Update step: Given the observation $\mathbf{y}_k$ we can use (C) to show that

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \propto p(\mathbf{y}_k \mid \mathbf{x}_k)\, p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$$

Notice that this scheme starts with $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$ and ends with $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$; hence, it provides a way to construct a recursive inference algorithm. Both of the above steps can be difficult to implement. Some special cases have an explicit formulation (e.g. the Celebrated Kalman filtering where the normal model of both $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ and $p(\mathbf{y}_k \mid \mathbf{x}_k)$ makes $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ a normal distribution as well. In general cases, the implementation of the above two steps use a Monte Carlo approach. The important sampling (IS) is used to develop a sequential DA algorithm, called particle filter (PF). Given $\{\mathbf{x}_{k-1}^{(1)}, \ldots, \mathbf{x}_{k-1}^{(m)}\} \sim p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$, the PF aims to construct $\{\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(m)}\} \sim p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$. Suppose that we have a set of particles $\mathbf{x}_{k-1}^{(j)}$ for $j = 1, \ldots, m$. At time k, the PF will carry out the following steps:

1. Draw $\mathbf{x}_k^{(j)}$ from a trial distribution $g(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)})$.
2. Compute the incremental weight $u_k^{(j)} \propto \dfrac{p(\mathbf{x}_k^{(j)} \mid \mathbf{x}_{k-1}^{(j)})\, p(\mathbf{y}_k \mid \mathbf{x}_k^{(j)})}{g(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)})}$
3. New weight: $w_k^{(j)} = w_{k-1}^{(j)}\, u_k^{(j)}$

It is clear that the above algorithm is recursive. In practice, it further requires a resampling to combat the weight degeneracy problem where only a small number of particles has a significant weights. The PF suffers from the curse of dimensionality and the resolution will require a creative design of the trial distribution, which will be problem-dependent. In addition, many problems in practice include unknown (fixed) parameters θ. Since x_k depends on θ at all time step, the joint state $[x_k,]$ is not Markovian and PF cannot be directly applied. To circumvent this issue, we may allow the parameters to move and write an augmented state $[x_k, \theta_k]$, where $\theta_k = \theta_{k-1} + \eta$ for some stochastic variable η and let x_k to depend only on θ_k so that the PF can be applied.

Xilu Wang

TRANSFER LEARNING-BASED SURROGATE ASSISTED EVOLUTIONARY BI-OBJECTIVE OPTIMIZATION FOR OBJECTIVES WITH DIFFERENT EVALUATION TIMES

ABSTRACT

Various multiobjective optimization algorithms have been proposed with a common assumption that the evaluation of each objective function takes the same period of time. Little attention has been paid to more general and realistic optimization scenarios where different objectives are evaluated by different computer simulations or physical experiments with different time complexities (latencies) and only a very limited number of function evaluations is allowed for the slow objective.

We propose a transfer learning scheme within a surrogate-assisted evolutionary algorithm framework to augment the training data for the surrogate for the slow objective function by transferring knowledge from the fast one.

Specifically, a hybrid domain adaptation method aligning the second-order statistics and marginal distributions across domains is introduced to generate promising samples in the decision space according to the search experience of the fast one.

A Gaussian process model based co-training method is adopted to predict the value of the slow objective and those having a high confidence level are selected as the augmented synthetic training data, thereby enhancing the approximation quality of the surrogate of the slow objective.

1. INTRODUCTION

Various multi-objective evolutionary algorithms (MOEAs) have been developed to solve multi-objective optimization problems (MOPs) [9, 5]. However, MOEAs usually require a large number of objective function evaluations (FEs) to generate satisfying approximations to Pareto fronts (PFs), indicating the difficulty to handle computationally expensive MOPs where FEs involve computationally intensive simulations or costly physical experiments. Surrogate assisted evolutionary algorithms (SAEAs) have emerged to be an effective methodology to overcome the computational obstacle for applying MOEAs to computationally expensive MOPs [11, 4]. Most MOEAs and SAEAs share a common assumption that the computational complexities of all objective functions are similar. However, it is common in many real-world applications that different objective functions require different evaluation times (also referred to as latencies [1]) due to differences in their computational complexities. Such MOPs are first considered by Allmendinger et al. in [2], and the authors further provided a general problem definition for MOPs with non-uniform evaluation times (NET-MOPs) in [1]. As done in [2, 1, 6, 14], we consider an expensive bi-objective optimization problem where one objective function is more computationally expensive than the other, NET-BOPs by short. To simulate NET-BOPs, we use the notations and assumptions as initially proposed in [1, 6], which we recap here for completeness reasons; the notations will also be to discuss related research and the proposed method for dealing with NET-BOPs.

continued ▶

- The slow (or delayed/expensive) objective function is denoted as f_s, while the fast (or non-delayed/cheap) one is denoted as f_f, and the corresponding surrogates for the two objective functions are denoted as GP_s and GP_f, respectively. We assume that the evaluation of the fast and slow objective functions can be done in parallel.
- It is assumed that the computation time for building surrogates and for implementing the genetic operators is negligible compared to that for evaluating the expensive objectives. Consequently, the total computation time available for solving the NET-BOPs is defined by the total budget for FEs.

In this work, we propose and validate a more efficient approach to exploit the transferable knowledge from the additional samples for f_f by integrating a TL strategy into a GP based SAEA for solving NET-BOPs, termed Tr-SAEA. Recall that more new samples $\mathbf{X}_f^{new}$ on the fast objective f_f can be collected than the slow objective owing to the big computational difference between f_f and f_s, when we use AFs to select samples to be evaluated using the true objective functions. We hypothesize that samples selected according to an AF for the fast objective f_f are not only helpful for the optimisation of f_f, but also beneficial for the optimisation of f_s. The reason for this hypothesis is that there is a functional relationship between the values of f_f and f_s (usually a trade-off relation) for solutions on or near the Pareto front [8]. Consequently, it makes sense to leverage knowledge readily available for f_f, i.e., the label-rich source domain in the light of TL, to improve the learning performance of f_s, which is the label-scarce target domain f_s.

2. PROPOSED ALGORITHM

2.1. Notations

- Two GPs for f_s, denoted as GP_{s1} and GP_{s2}, are constructed in the co-training paradigm in Tr-SAEA. The initial population $\boldsymbol{X}$ is evaluated on both f_f and f_s.
- While a very limited number of new samples $D_s^{new} = (\boldsymbol{X}_s^{new}, \boldsymbol{Y}_s^{new})$ is affordable for f_s, more new samples $D_f^{new} = (\boldsymbol{X}_f^{new}, \boldsymbol{Y}_f^{new})$ are allowed ot be evaluated on f_f. We apply the proposed HDA method on D_f^{new} so that some promising candidate solutions $\boldsymbol{X}_s^{new}$ for the target domain f_s are generated. Data $\boldsymbol{X}_f^{new}$ and $\boldsymbol{X}_s^{new}$ are combined as a set of promising samples $\boldsymbol{X}_s^a$ for f_s. Note that the slow objective values associated with $\boldsymbol{X}_s^a$ are unknown; hence, a co-training based SSL algorithm is further proposed to leverage these unlabeled target solutions.
- Let $(\boldsymbol{Y}_{s1}^a, \boldsymbol{\sigma}_{s1}^a)$ and $(\boldsymbol{Y}_{s2}^a, \boldsymbol{\sigma}_{s2}^a)$ denote the predictions of GP_{s1} and GP_{s2} on $\boldsymbol{X}_s^a$, respectively, where $\boldsymbol{Y}$ denotes the predicted objective value and σ denotes the corresponding a confidence level. Subsequently, each of the regression models GP_{s1} and GP_{s2} will identify a subset of $\boldsymbol{X}_s^a$ with a higher confidence level to update each other. In this way, the chosen subsets denoted as D_{t1} and D_{t2} can augment the training data for f_s, transferring knowledge from f_f to f_s.
- $D_{s1} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t1}$ and $D_{s2} = (\boldsymbol{X}, \boldsymbol{Y}_s) + D_s^{new} + D_{t2}$ are defined as the training data sets for GP_{s1} and GP_{s2}, respectively, while $D_f = (\boldsymbol{X}, \boldsymbol{Y}_f) + D_f^{new}$ is defined as the training data set for GP_f.

continued ▶

2.2. Algorithm Framework

Tr-SAEA focuses on how to transfer knowledge embedded in the label-rich source domain f_f to improve the learning of the label-scarce domain f_s in building surrogates. We design two major components in Tr-SAEA, a hybrid domain adaptation (HDA) method and a co-training mechanism, as described by the following step-by-step procedure. The details of the key components in Tr-SAEA will be presented in the following subsections.

- Step 1: Construct surrogate models. Tr-SAEA begins with sampling two different training data sets (D_f^0 and D_s^0) from f_f and f_s in the same way used in [7], where the Latin hypercube sampling (LHS) is employed to initialize the population and a SOEA is performed to consume the additional evaluation budget available for f_f. Subsequently, while a GP_f is trained with D_f^0 for f_f , GP_{s1} and GP_{s2} using different kernel functions with D_s^0 are generated to implement co-training mechanism for f_s.
- Step 2: Select new samples. Similar to standard GP-based SAEAs, a baseline MOEA, which is RVEA [5] in this work, is employed to optimize the NET-BOP for a certain number of generations. Here, the two objectives are predicted by the GPs instead of using the true objective values. The optimized population will then be evaluated according to the AFF [13], and the selection strategy in RVEA is adopted to select u and $u * \tau$ new samples to be evaluated using f_f and f_s, respectively, due to the different evaluation times. In this way, we can obtain $D_s^{new} = (\boldsymbol{X}_s^{new}, \boldsymbol{Y}_s^{new})$ and $D_f^{new} = (\boldsymbol{X}_f^{new}, \boldsymbol{Y}_f^{new})$.
- Step 3: Implement HDA in the objective space to align the second-order statistics and marginal distributions of the source domain (f_f) and target domain (f_s). Motivated by the fact that there is a functional relationship between f_s and f_f on and near the PF, we attempt to find a common latent space to minimize the difference between the two domains. Firstly, CORAL is adopted to construct a transformation matrix (denoted as A) to minimize the domain shift by aligning the second-order statistics of both domains. As CORAL does not consider the distribution alignment, TCA is adopted to further discover a feature representation in a latent space having the same marginal distribution across the two domains. Using the joint matching method yields the mapping $\boldsymbol{Y}'^{new}_s$ of $\boldsymbol{Y}'^{new}_f$ in the latent domains. Subsequently, an SOEA is adopted to exploit promising decision variables $\boldsymbol{X}'^{new}_s$ whose associated slow objective values are close to $\boldsymbol{Y}'^{new}_s$ in the obtained latent space.
- Step 4: Implement the co-training strategy to identify reliable unlabeled data with regard to the predictions of GP_{S1} and GP_{S2} , which will be then used as labelled data for training GP_{S1} and GP_{S2} together with the real data. Let the set of promising samples be $\boldsymbol{X}_s^a = (\boldsymbol{X}_f^{new}, \boldsymbol{X}'^{new}_s)$. The two GPs for f_s separately provide their predictions on $\boldsymbol{X}_s^a$, including the predicted fitness value and a confidence level. Based on the labeling confidence, GP_{s1} and GP_{s2} will select $u * \tau$ unlabeled samples (D_{t1} and D_{t2}) associated with predicted labels from $\boldsymbol{X}_s^a$, and then add these reliable unlabeled samples to their training data sets, respectively.
- Step 5: Update GPs with different training data sets. GP_{s1} and GP_{s2} will be updated with $D_{s1} = (\boldsymbol{X} , \boldsymbol{Y}_s) + D_s^{new} + D_{t1}$ and $D_{s2} = (\boldsymbol{X} , \boldsymbol{Y}_s) + D_s^{new} + D_{t2}$, respectively, while GP_f is updated with $D_f = (\boldsymbol{X} , \boldsymbol{Y}_f) + D_f^{new}$. The process repeats until the maximum number of evaluations is reached.

continued ▶

2.3. Hybrid Domain Adaptation

In Tr-SAEA, a hybrid domain adaptation method is proposed to use the available data drawn from the source domain (f_f) to generate data in the decision space for the target domain (f_s) based on the correlation between the two domains. The key motivation is that there is a functional relationship between f_f and f_s when they are closed to the PF [8]. We adopt CORAL to align the second-order statistics of both domains due to its simplicity and efficiency, and then employ the TCA to match the marginal distributions of both domains in a Reproducing Kernel Hilbert Space (RKHS).

CORAL constructs a transformation matrix A of the source features by minimizing the distance between the second-order statistics across the source and target domains, and the Frobenius norm is adopted as the matrix distance metric, which can be achieved as follows,

$$\min_A \left\| C_{\hat{S}} - C_T \right\|_F^2 = \min_A \left\| A^T C_s A - C_T \right\|_F^2 \quad (1)$$

where $C_{\hat{S}}$, C_S and C_T denote covariance matrices of the transformed source features, the source features, and target features, respectively, and $\| \cdot \|_F^2$ denotes the squared matrix Frobenius norm. Note that, as an alternative of other approaches to domain shift, CORAL avoids subspace projection that can be computationally intensive. Moreover, CORAL is very easy to implement. However, it is not able to account for the distribution alignment.

TCA is adopted to explore a set of common transfer components across the two domains by minimizing the marginal distribution discrepancy in a latent space, while preserving data properties in the original space. To achieve this, TCA minimizes the distance between the means of the source and target data based on the RKHS using the maximum mean discrepancy (MMD) [3] as a marginal distribution measurement criterion, and further enforces the scatter matrix as a constraint [12]. The distance between two distributions $P (Z_S)$ and $P (Z_T)$ can be estimated by MMD in an RKHS [12],

$$\text{Dist} (Z'_S , Z'_T) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \varphi (Z_{S_i}) - \frac{1}{n_2} \sum_{i=1}^{n_2} \varphi (Z_{T_i}) \right\|_{\mathcal{H}}^2 \quad (2)$$

where Z'_S and Z'_T denote the transformed input sets from the source and target domains, n_1 and n_2 denote the number of samples in each domain, and φ denotes the desired transformation mapping. Here, CORAL cooperates with TCA to construct a mapping matrix $A \times W$, allowing us to transfer some promising samples of the source domain to the target domain for the benefit of training the surrogate and guiding the search of f_s. Hence, once $\mathbf{Y}_f^{new}$ is generated, we can map them into the latent space with the help of the mapping matrix $A \times W$ in order to generate useful data $\mathbf{Y}'^{new}_s$ for the target domain. Similar to the method in [10], we further explore the decision space to find new samples $\mathbf{X}'^{new}_s$ and enforce its objective values on f_s to be close to $\mathbf{Y}'^{new}_s$ in the latent space.

2.4 GP-based Co-training Method

With the help of domain adaptation, we are able to identify a set of promising samples $\boldsymbol{X}_s^a$ (in decision space), which are considered to be useful for the target domain (f_s). Here, a GP-based co-training method is proposed in Tr-SAEA to leverage these unlabeled data.

continued ▶

First, in GP-based co-training method, we utilize different kernel functions in the two GPs (GP_{s1} and GP_{s2}) to achieve diversity, instead of requiring sufficient and redundant views or different learning algorithms.

Second, for the selected unlabeled samples $\boldsymbol{X}^a_s$, GP_{s1} and GP_{s2} provide their predictions combined with a confidence level: ($\boldsymbol{Y}^a_{s1}$, $\boldsymbol{\theta}^a_{s1}$) and ($\boldsymbol{Y}^a_{s2}$, $\boldsymbol{\theta}^a_{s2}$), respectively. We then rank the predicted data according to the confidence level in an ascending order. Then, the most reliable predicted samples are selected as the augmented data sets D_{t1} and D_{t2} to enlarge the training sets D_{s1} and D_{s2}, respectively.

3. EXPERIMENTAL RESULTS

We have selected three widely used suites of bi-objective test problems for our experimental study, and extend them to simulate NET-BOPs. The inverted generational distance (IGD) [16] and hypervolume (HV) [15] are adopted to assess the performance of the algorithms.

3.1. Comparison with State-of-the-art methods

From these results, we can see that Tr-SAEA has achieved the best overall performance on the three test suites, followed by HK-RVEA. Overall, the non-surrogate based methods (*Waiting*, *Fast-first*, *Brood* and *Speculative interleaving*) cannot compete with the surrogate-based methods (HK-RVEA, Tr-SAEA, T-SAEA), confirming the observation already made in [6].

4. CONCLUSION

In this paper, an effort is made to solve MOPs where different objective functions take considerably different evaluation times. To more efficiently solve bi-objective optimization problems with one fast objective and one slow objective (NET-BOPs), we integrate a transfer learning scheme into an SAEA framework to augment the training data for the slow objective by transferring knowledge from the fast one, thereby alleviating the search biases caused by different computational budgets needed for evaluating the two objectives. We compare the proposed algorithm, Tr-SAEA, with five state-of-the-art (surrogate and non-surrogate-based) delay-handling strategies, as well as four variants of Tr-SAEA, on three widely used test suites.

4. REFERENCES

1. Allmendinger, R., Handl, J., Knowles, J., 2015. Multiobjective optimization: When objectives exhibit non-uniform latencies. European Journal of Operational Research 243, 497–513.
2. Allmendinger, R., Knowles, J., 2013. 'Hang On a Minute': Investigations on the e˙ects of delayed objective functions in multiobjective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 6–20.
3. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J., 2006. Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22, e49–e57.
4. Carlos A. Coello Coello, S.G., Figueroa Gamboa, J., Guadalupe Castillo Tapia, M., Hernández Gómez, R., 2020. Evolutionary multi-objective optimization: open research areas and some challenges lying ahead. Complex & Intelligent Systems 6, 221–236.
5. Cheng, R., Jin, Y., Olhofer, M., Sendho˙, B., 2016. A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation 20, 773–791.

continued ▶

Test problem	Waiting	Fast-first	Brood	Speculative	HK-RVEA	T-SAEA	Tr-SAEA	Undelayed
	mean std	mean std	mean std	mean std	mean std	mean std	mean std	mean std
DTLZ1	30.2 + 14.3	69.7 + 24.1	28.6 + 10.2	48.1 + 10.5	42.2 + 10.5	21.7 + 11.9	**20.7** 5.38	15.8 5.71
DTLZ1a	14.2 + 8.48	2.62 + 0.25	15.6 + 6.54	28.6 + 8.22	0.52 + 0.18	1.06 + 1.00	**0.20** 0.07	0.32 0.08
DTLZ2	0.24 + 0.05	0.80 + 0.08	0.36 + 0.05	0.38 + 0.03	0.10 + 0.02	0.05 + 0.03	**0.03** 0.01	0.02 0.01
DTLZ3	349 ≈ 83.4	549 + 142	357 + 74.6	462 + 67.5	354 ≈ 41.9	**203** − 100	327 82.1	168 61.9
DTLZ3a	227 + 75.4	546 + 86.9	313 + 83.1	406 + 96.6	14.9 + 5.27	5.34 + 37.5	**3.39** 1.87	13.9 2.32
DTLZ4	0.51 + 0.32	0.78 + 0.11	0.54 + 0.06	0.65 + 0.10	0.23 ≈ 0.11	0.60 + 0.13	**0.16** 0.07	0.05 0.07
DTLZ5	0.27 + 0.06	0.86 + 0.10	0.35 + 0.04	0.39 + 0.03	0.09 + 0.02	0.05 + 0.02	**0.03** 0.03	0.02 0.00
DTLZ6	7.31 + 0.52	8.79 + 0.11	7.63 + 0.44	8.26 + 0.13	4.10 + 0.54	2.56 + 1.21	**0.72** 0.09	2.52 0.04
DTLZ7	4.41 + 0.62	7.53 + 0.39	5.53 + 0.47	5.57 + 0.68	0.06 ≈ 0.05	1.15 + 0.91	**0.03** 0.01	0.04 0.00
UF1	1.01 + 0.14	0.49 + 0.04	0.36 + 0.02	0.42 + 0.05	0.23 + 0.02	**0.19** ≈ 0.02	**0.19** 0.01	0.15 0.04
UF2	0.50 + 0.07	0.58 + 0.09	0.45 + 0.03	0.51 + 0.03	0.15 + 0.02	0.14 ≈ 0.02	**0.12** 0.01	0.09 0.02
UF3	0.97 + 0.08	1.22 + 0.06	0.96 + 0.04	1.08 + 0.07	0.54 ≈ 0.05	**0.19** − 0.08	0.49 0.01	0.44 0.02
UF4	**0.21** ≈ 0.01	0.24 ≈ 0.02	0.23 ≈ 0.00	0.23 ≈ 0.01	0.22 ≈ 0.00	0.23 ≈ 0.02	0.22 0.00	0.12 0.01
UF5	4.75 + 0.42	3.53 + 0.26	2.84 + 0.18	3.25 + 0.16	2.46 ≈ 0.43	2.49 ≈ 0.44	**2.43** 0.28	1.55 0.31
UF6	4.36 + 0.63	2.28 + 0.24	1.69 + 0.13	1.99 + 0.17	1.34 ≈ 0.13	**1.01** − 0.25	1.32 0.39	0.58 0.13
UF7	1.20 + 0.12	0.58 + 0.06	0.38 + 0.03	0.47 + 0.05	**0.21** − 0.04	0.37 ≈ 0.06	0.32 0.11	0.32 0.13
ZDT1	6.37 + 1.63	17.5 + 3.22	11.9 + 1.43	12.8 + 2.49	0.21 + 0.07	1.05 + 0.59	**0.02** 0.01	0.02 0.00
ZDT2	8.30 + 1.89	20.6 + 2.20	11.6 + 2.17	12.9 + 4.21	**0.54** − 0.14	1.05 + 0.56	0.56 0.12	0.02 0.00
ZDT3	6.30 + 1.75	17.3 + 3.01	11.2 + 1.86	13.0 + 2.45	**0.15** ≈ 0.02	0.70 + 0.37	**0.15** 0.21	0.04 0.02
ZDT4	34.6 + 6.47	36.0 + 8.96	18.8 + 5.62	21.8 + 6.69	**2.25** − 1.92	33.0 + 7.70	3.29 2.74	17.2 9.8
ZDT6	9.38 + 0.64	11.3 + 0.29	9.91 + 0.49	10.4 + 0.37	1.11 + 0.49	2.44 + 0.88	**0.61** 0.06	0.49 0.10
+/≈/-	19/2/0	20/1/0	20/1/0	20/1/0	10/8/2	13/5/3		

Table 1 | Statistical results of the IGD values obtained by Waiting, Fast-first, Brood interleaving, Speculative interleaving, HK-RVEA, T-SAEA, Tr-SAEA and Undelayed algorithn with $FE_s^{max} = 200$ and $\tau = 5$.

6. Chugh, T., Allmendinger, R., Ojalehto, V., Miettinen, K., 2018a. Surrogate-assisted evolutionary biobjective optimization for objectives with non-uniform latencies, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM. pp. 609–616.
7. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K., 2018b. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation 22, 129–142.
8. Deb, K., 2001. Multi-objective optimization using evolutionary algorithms. volume 16. John Wiley & Sons.
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation 6, 182–197.
10. Jiang, M., Huang, Z., Qiu, L., Huang, W., Yen, G.G., 2017. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Transactions on Evolutionary Computation 22, 501–514.
11. Jin, Y., Wang, H., Chugh, T., Guo, D., Miettinen, K., 2019. Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation 23, 442–458.
12. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2010. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 199–210.
13. Wang, X., Jin, Y., Schmitt, S., Olhofer, M., 2020a. An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization. Information Sciences 519, 317–331.
14. Wang, X., Jin, Y., Schmitt, S., Olhofer, M., 2020b. Transfer learning for Gaussian process assisted evolutionary bi-objective optimization for objectives with di˙erent evaluation times, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2020), ACM.
15. While, L., Hingston, P., Barone, L., Huband, S., 2006. A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation 10, 29–38.
16. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G., 2003. Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7, 117–132.

Alessandro Pastore

NUCLEAR PHYSICS IN A MACHINE LEARNING ERA

1. INTRODUCTION

To date more than 2400 nuclear masses have been measured [1], but current nuclear models predict the existence of more than 6000 [2]. The detailed knowledge of nuclear masses is key to understand several physical phenomena as the r-process nucleosynthesis [3] or to assess the structure of the crust of a neutron star [4, 5].

Within the literature, it is possible to find several mass models with a typical accuracy, i.e., the root mean square (RMS) deviation of the residuals, spanning from 2 MeV to 500 keV [6]. Although such an accuracy is remarkably good compared to the typical nuclear binding energies, it is important to further improve the accuracy of such models in order to be able to describe correctly the properties of very neutron rich nuclei [7]. In order to achieve such an ambitious goal, several techniques have been employed, but the most promising ones are based on the use of machine learning (ML) methods such as kernel ridge regression [8], radial basis function interpolation [9, 10], neural networks [11–13] or Gaussian Process [14]. By adding these ML on top of a nuclear model, it is thus possible to reduce the discrepancy between the data and the extrapolated values up to less than 200 keV [14].

In the present article, I present the main achievements obtained in Ref. [14] and based on the use of Gaussian Process (GP) [15]. By adjusting the parameters of a GP on the mass residuals of two nuclear functionals: UNEDF0 [16] and UNEDF1 [17], I will show how it is possible to reduce the global root mean square (RMS) deviation of the residuals. This GP method assumes that the residuals originate from some multivariate Gaussian distribution, whose covariance matrix contains some parameters to be adjusted in order to maximise the likelihood for the GP's fit to the residuals.

2. APPLICATION OF GAUSSIAN PROCESS TO NUCLEAR MASSES

A Gaussian process is an infinite-dimensional Gaussian distribution. Similar to how a one dimensional (1D) Gaussian distribution has a mean μ and variance σ^2, a GP has a mean function $\mu(\mathbf{x})$, and a covariance function $k(\mathbf{x}, \mathbf{x}')$, also known as the kernel [18]. $\mathbf{x}$ is a vector of length d representing a point in a d-dimensional input space. Just as we can draw random samples (numbers) from a 1D Gaussian distribution, one can also draw random samples from a GP, which are functions $f(x)$. The kernel $k(x, x')$ tells us the typical correlation between the value of f at any two inputs x and x', and entirely determines the behaviour of the GP (relative to the mean function). For simplicity, one can set a constant mean function of 0. If the data have a non-zero mean, it is always possible to rescale them.

continued ▶

In the current work, I use the following kernel

$$k_{\mathrm{RBF}}(x, x') = \eta^2 \exp\left[-\frac{(N-N')^2}{2\rho_N^2} - \frac{(Z-Z')^2}{2\rho_Z^2}\right] + \sigma_n^2 \delta_{xx'}, \quad (1)$$

where in the present case x = (N, Z), and η^2, ρ_Z, ρ_N are the adjustable parameters. Following Ref. [7], ρ_N and ρ_Z are interpreted as correlation lengths in the neutron and proton directions, while η^2 gives the strength of the correlation between neighbouring nuclei.

The addition of the nugget means that the GP mean now does not necessarily pass directly through each data point. The main role of the nugget is to avoid over-fitting, which manifests itself via a correlation length smaller to the typical separation of the data [13]. For a more detailed discussion on GP and the role of the nugget we refer to Ref. [7].

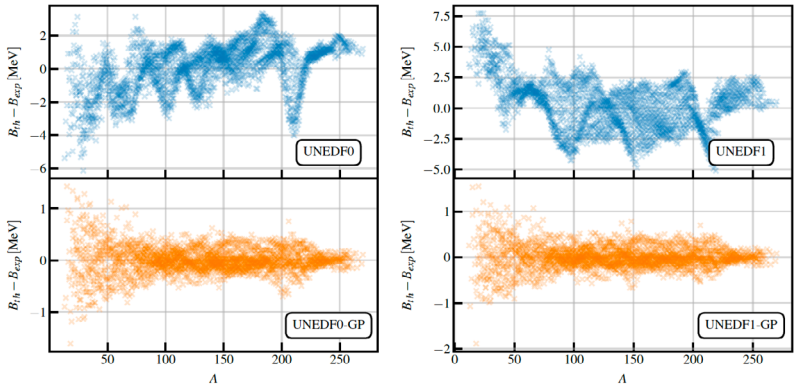


Figure 1 | Upper panels: residuals of the UNEDF0 and UNEDF1 mass models. Lower panels: residuals of the UNEDF0 and UNEDF1 models equipped with the additional GP. See text for details.

Following the procedure applied in Ref. [14], I consider only the 2400 nuclear masses directly provided in the AME2016 database [1]. In the upper panels of Fig. 1, I illustrate the residuals of the two UNEDF0 and UNEDF1 models in the region $8 \leq Z \leq 110$. The RMS of the two models is $\sigma_{\mathrm{UNEDF0}} = 1.43$ MeV $\sigma_{\mathrm{UNEDF1}} = 2.00$ MeV. This value is quite high compared to more sophisticated mass models as discussed in Ref. [6], but one has to bear in mind that UNEDF models reproduce also a large variety of other nuclear observables.

After applying the GP detailed in Eq.1 on the residuals of UNEDF0 and UNEDF1, I obtain more accurate mass models whose residuals are illustrated in the lower panels of Fig. 1. The global RMS goes down to $\sigma_{\mathrm{UNEDF0\text{-}GP}} = 0.419$ MeV $\sigma_{\mathrm{UNEDF1\text{-}GP}} = 0.420$ MeV.

continued ▶

3. CONCLUSIONS

By using a Gaussian process with 4 adjustable parameters fitted to the residuals of the UNEDF mass model, I have been able to create a mass model with a global RMS of less than 500 keV, thus improving the original value by a factor of 3-4. The improvement is roughly independent of the starting point, i.e. the nuclear mass model used to originate the residuals as also shown in Ref. [7]. Having improved the models, one can now use them to perform extrapolations in regions of the nuclear chart where no data are available [5, 14]. At large extrapolations, the trend is always dictated by the model since the GP tends to 0. To avoid such a model dependence, in recent years, some authors suggested to perform a Bayesian Model Averaging in order to obtain more robust extrapolations [19].

ACKNOWLEDGEMENTS

This work has been supported by STFC Grant No. ST/P003885/1.

REFERENCES

1. M. Wang, G. Audi, F. Kondev, W. Huang, S. Naimi, and X. Xu, Chinese Physics C 41, 030003 (2017).
2. J. Erler, N. Birge, M. Kortelainen, W. Nazarewicz, E. Olsen, A. M. Perhac, and M. Stoitsov, Nature 486, 509 (2012).
3. M. Mumpower, R. Surman, D.-L. Fang, M. Beard, P. M¨oller, T. Kawano, and A. Aprahamian, Physical Review C 92, 035807 (2015).
4. N. Chamel and P. Haensel, Living Reviews in relativity 11, 10 (2008).
5. A. Pastore, D. Neill, H. Powell, K. Medler, and C. Barton, Physical Review C 101, 035804 (2020).
6. A. Sobiczewski, Y. A. Litvinov, and M. Palczewski, Atomic Data and Nuclear Data Tables 119, 1 (2018).
7. L. Neufcourt, Y. Cao, W. Nazarewicz, F. Viens, et al., Physical Review C 98, 034318 (2018).
8. X. Wu and P. Zhao, Physical Review C 101, 051301 (2020).
9. N. Wang and M. Liu, Physical Review C 84, 051303 (2011).
10. Z. Niu, H. Liang, B. Sun, Y. Niu, J. Guo, and J. Meng, Science Bulletin 63, 759 (2018).
11. J. W. Clark, in *Scientific applications of neural nets* (Springer, 1999), pp. 1–96.
12. Z. Niu and H. Liang, Physics Letters B 778, 48 (2018).
13. A. Pastore and M. Carnini, Journal of Physics G: Nuclear and Particle Physics 48, 084001 (2021).
14. M. Shelley and A. Pastore, Universe 7, 131 (2021).
15. L. S. Bastos and A. O'Hagan, Technometrics 51, 425 (2009).
16. M. Kortelainen, T. Lesinski, J. Mor´e, W. Nazarewicz, J. Sarich, N. Schunck, M. Stoitsov, and S. Wild, Physical Review C 82, 024313 (2010).
17. M. Kortelainen, J. McDonnell, W. Nazarewicz, P.-G. Reinhard, J. Sarich, N. Schunck, M. Stoitsov, and S. Wild, Physical Review C 85, 024304 (2012).
18. C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning (MIT Press, Cambridge, Mass, 2006), ISBN 978-0-262-18253-9.
19. L. Neufcourt, Y. Cao, W. Nazarewicz, E. Olsen, F. Viens, et al., Physical review letters 122, 062502 (2019).

Tao Chen

SURROGATE ASSISTED CALIBRATION OF COMPUTATIONAL FLUID DYNAMICS MODELS

ABSTRACT

Computational fluid dynamics (CFD) is a simulation technique widely used in chemical and process engineering applications. However, computation has become a bottleneck when calibration of CFD models with experimental data (also known as model parameter estimation) is needed. In this research, the kriging meta-modelling approach (also termed Gaussian process) was coupled with expected improvement (EI) to address this challenge. A new EI measure was developed for the sum of squared errors (SSE) which conforms to a generalised chi-square distribution and hence existing normal distribution-based EI measures are not applicable. The new EI measure is to suggest the CFD model parameter to simulate with, hence minimising SSE and improving match between simulation and experiments. The usefulness of the developed method was demonstrated through a case study of a single-phase flow in both a straight-type and a convergent-divergent-type annular jet pump, where a single model parameter was calibrated with experimental data. This talk is based on a journal article we previously published in the AIChE Journal.

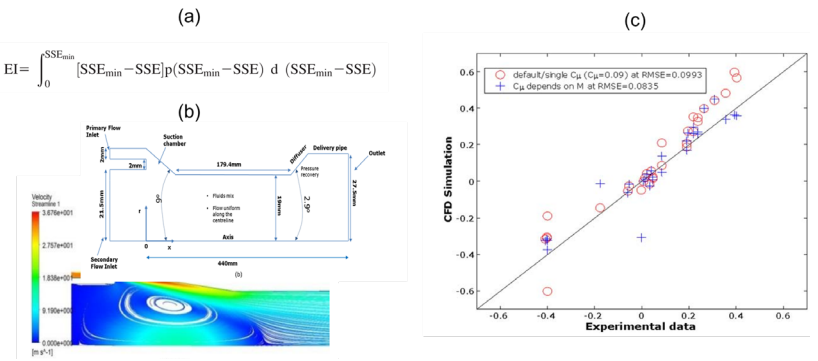


Figure (a) Using expected improvement (EI) of the sum-of-squared (SSE) error as a measure for model calibration under the uncertainty between the surrogate model and the underlying complex simulation. **Figure (b)** Case study on a convergent-divergent jet pump and its typical CFD simulation. **Figure (c)** Illustration of calibration results. Kajero et al., AIChE J, 62: 4308, 2016.

Carmen Calama Gonzalez

BAYESIAN CALIBRATION OF BUILDING ENERGY MODELS

RESEARCH HYPOTHESIS AND OBJECTIVE

The improvement in energy efficiency of existing buildings is key for meeting 2030 and 2050 energy and CO_2 emission targets. Three-quarters of the existing Spanish stock was built prior to energy performance regulations and the current new-built construction rate is less than 2% [1]. Building energy modelling (BEM) plays a crucial role in evaluating the performance of retrofit actions in terms of energy consumption and thermal comfort. Nonetheless, uncalibrated and invalidated BEM may lead to uncertain predictions. This research [2] aims at determining whether or not it is possible to properly calibrate and validate BEM at an hourly basis through on-site measurements to accurately assess the thermal comfort performance in buildings.

METHODOLOGY

A Bayesian calibration approach [3], combined with Morris sensitivity analysis [4,5], is applied to reduce discrepancies between measured and simulated hourly indoor air temperatures. The following steps are taken (Figure 1): Firstly, a BEM of a controlled and monitored environment, a test cell case study, is developed using EnergyPlus. Later, the most influential model parameters on the simulation outputs are obtained through the one-step-at-a-time Morris sensitivity analysis. Only the top-4 most influential parameters are calibrated, since the Bayesian method is computationally prohibitive in a high-dimensional parameter space [ ] and increasing the calibration parameters may lead to inaccuracy and ineffectiveness. Finally, the accuracy of the calibrated model is measured using the uncertainty indices defined in the ASHRAE Guideline [ ]: the Normalized Mean Bias Error (NMBE), the Coefficient of Variation of the Root Mean Square Error (CVRMSE) and the Coefficient of Determination (R^2).

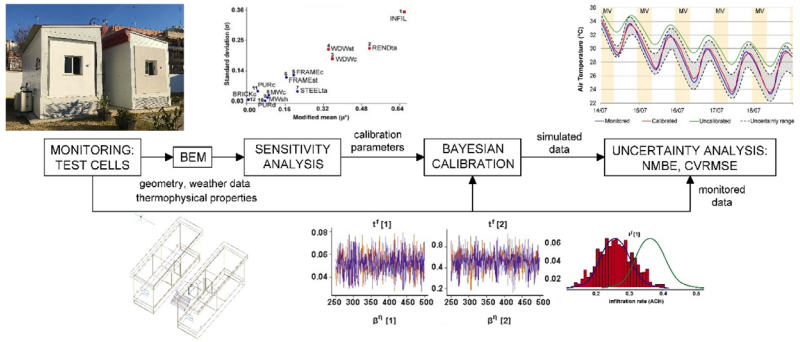


Figure 1 | Methodology followed [2]

continued ▶

In calibration, the number of iterations the algorithm runs and the warm-up argument (steps used to automatically tune the sampler) were set at 500 and 250, respectively. Since the assessment of thermal comfort is normally based on hourly data, it requires a higher precision than that of energy consumption (generally evaluated at monthly resolution: 12 points). Thus, a 24-hour training period is used, with a total of 960 simulated points. Uncertainties associated with model inputs (fixed parameters in the model), model discrepancies due to physical limitations of the BEM (simplifications when compared to the real performance of the building), errors in field observations and noisy measurements are accounted for. The parameter uncertainty was taken into account by specifying a prior distribution for the calibration parameters, which includes the most likely range of possible values, considering building specifications, tests and expert judgement. Simulation runs are used to identify which parameters are most likely to lead to the observed data, updating prior distributions and calculating posterior distributions.

ANALYSIS AND DISCUSSION

Several scenarios are evaluated to determine how different variables may impact the calibration: (1) solar radiation by changing orientation (North and South); (2) mechanical ventilation (MV), which is scheduled as OFF or ON (from 22:00 to 8:00 at 1.75 ACH); and (3) blind aperture levels, considering no window for north orientation (b0) and window half open for south orientation (b50).

Sensitivity analysis reported that the most influential parameters in the north facing cell were infiltration, thermal absorptance of the façade and roof, and conductivity of the facade. In the south facing cell, the top-four parameters were conductivity and solar transmittance of the glazing surface, infiltration and thermal absorptance of the façade. The ventilation rate and fan efficiency were of utmost importance when the MV was ON in both orientations. In calibration, visual inspection of the plots suggested that the sampling algorithm was efficiently exploring the posterior distribution. The potential scale reduction statistic (Rhat) was within 1.0±0.1 in all scenarios. Thus, convergence was successfully achieved. Results of the calibration analysis through the uncertainty indices are shown in Table 1. To check for bias in the evaluation process, these indices are determined using an independent 120-hour dataset with 4,800 simulated points (testing period), different from the training period. Even though uncalibrated models were within the uncertainty ranges of the ASHARE, pre-calibration simulation outputs over-predicted measurements up to 3.2 °C. After calibration, the average maximum temperature difference was reduced to 0.68 °C, improving the results by almost 80%.

Monitored data was within the uncertainty range (95% confidence intervals) of the calibrated model, considering variations within the posterior distribution ranges. With an accuracy of the probes of ±0.5 °C and ±1.0 °C for 10-30 °C and 30-55 °C, respectively, the model is considered to be well calibrated.

continued ▶

Scenario	Calibrated	NMBE (±10%)	CVRMSE (<30%)	R^2 (>0.75)	Max. $T_{difference}$ (°C)	P-value (T-Test)
North MVOFF b0	No	8.92%	10.04%	0.61	2.63	0.49
	Yes	0.44%	0.90%	0.98	0.51	0.82
North MVON b0	No	-7.46%	11.57%	0.72	2.41	0.25
	Yes	-0.22%	1.42%	0.97	0.74	0.70
South MVOFF b50	No	-8.35%	8.57%	0.84	3.31	0.38
	Yes	-0.22%	0.81%	0.98	0.75	0.78
South MVON b50	No	-9.31%	9.81%	0.93	4.42	0.43
	Yes	-0.78%	1.45%	0.98	0.74	0.72

Table 1 | Assessment of the model's accuracy: comparison between uncalibrated and calibrated models [2]

CONCLUSIONS

Implementing a first level statistical calibration-simulation methodology, which combines sensitivity analysis and Bayesian techniques, is proven to improve the level of agreement between on-site measurements and simulated outputs. Applying this method is useful for calibrating and validating indoor hourly temperatures, providing adequate results for thermal comfort assessment. However, results reported may only be applied to simple houses or small single zone units (flats or small offices) with limited ventilation and few wall partitioning. Besides, this research was carried out in an unoccupied, highly controlled environment with free-running conditions (no HVAC systems). Thus, future research should test this methodology in real building, evaluating its viability and accuracy in models with different grades of complexity.

REFERENCES

1. International Energy Agency. Tracking buildings 2020 https://www.iea.org/reports/tracking-buildings-2020 [May 2021]
2. Calama-González, C.M., Symonds, P., Petrou, G., Suárez, R., & León-Rodríguez, Á.L. (2021). Bayesian calibration of building energy models for uncertainty analysis through test cells monitoring. Applied Energy, 282, 116118.
3. Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J R Stat Soc Ser B (Statistical Methodol 2001;63:425–464. https://doi.org/10.1111/1467-9868.00294
4. Morris MD. Factorial sampling plans for preliminary computational experiments. Technometrics 1991;33:161–174. https://doi.org/10.1080/00401706.1991.10484804
5. Campolongo F, Cariboni J, Saltelli A. An effective screening design for sensitivity analysis of large models. Envir Mod Software 2017;22:1509-1518. https://doi.org/10.1016/j.envsoft.2006.10.004
6. Chong A, Menberg K. Guidelines for the Bayesian calibration of building energy models. Energy Build 2018;174:527–547. https://doi.org/10.1016/j.enbuild.2018.06.028
7. American Society of Heating, Ventilating, and Air Conditioning Engineers. Guideline 14-2002, Measurement of Energy and Demand Savings; Technical Report; Atlanta, GA, USA, 2002

Alex Shestopaloff

THE ESSENTIALS OF MARKOV CHAIN MONTE CARLO

1. MOTIVATING EXAMPLE: STATE SPACE MODELS

State space models model the distribution of an observed sequence $y_{1:T} = (y_1, \ldots, y_T)$. Here, Y_t are drawn from an observation density $g(y_t \mid x_t, \theta)$ and X_t is an unobserved Markov process with initial density $\mu(x_1 \mid \theta)$ and transition density $f(x_t \mid x_{t-1}, \theta)$. We focus on inferring $X_{1:T} = (X_1, \ldots, X_T)$, assuming θ is known. Parameter inference can be easily built on top of this, eg,. by alternately inferring $(X_1, \ldots X_T)$ and θ.

1.1. Bayesian inference for state space models

We will infer $X_{1:T}$ by sampling from the posterior density of $X_{1:T}$ given $y_{1:T}$,

$$p(x_{1:T} \mid y_{1:T}) \propto \underbrace{\mu(x_1) \prod_{t=2}^{T} f(x_t \mid x_{t-1})}_{\text{Prior}} \underbrace{\prod_{t=1}^{T} g(y_t \mid x_t)}_{\text{Likelihood}}.$$

No exact solution to this sampling problem is present, except for linear Gaussian models or models with a finite state space. In these cases, we can use the Kalman filter or the forward-backward algorithm. Most often, approximate methods such as Markov Chain Monte Carlo (MCMC) must be used. Why is it difficult to sample from $p(x_{1:T} \mid y_{1:T})$ with MCMC? Strong temporal dependencies amongst the x_t can make sampling inefficient and thus requiring sophisticated MCMC methods.

2. BASICS OF MARKOV CHAINS

Suppose we want to sample $X_{1:T}$ when $X_{1:T}$ has density $p(x_{1:T} \mid y_{1:T})$. To do this, we can construct a Markov chain with transition kernel $K(x_{1:T}, x'_{1:T})$ that leaves p invariant. This means that the transition kernel K satisfies

$$\int p(x_{1:T} \mid y_{1:T}) K(x_{1:T}, x'_{1:T}) \, dx_{1:T} = p(x'_{1:T} \mid y_{1:T}).$$

Provided the Markov chain is ergodic, this condition ensures that its distribution will converge to p. This leads to an approach for drawing samples from P. This is done by simulating the Markov chain. If we run the Markov chain long enough, we will be able to draw samples approximately distributed as p. We would like the Markov chain to produce samples with low autocorrelation time, after adjusting for computation time. We are interested in samples with low autocorrelation time in order to produce estimates with low variance.

continued ▶

3. COMMON MCMC ALGORITHMS

We briefly look at a couple of common MCMC algorithms.

3.1. Metropolis-Hastings

Suppose we are interesting in sampling from $p(x_{1:T} \mid y_{1:T})$. Given $x_{1:T}$, we use a proposal density $q(x^*_{1:T} \mid x_{1:T})$ to draw a candidate $x^*_{1:T}$ for the next point $x'_{1:T}$ in the chain. Then, we accept $x^*_{1:T}$ with probability

$$\min\left[1, \frac{p(x^*_{1:T} \mid y_{1:T})\, q(x_{1:T} \mid x^*_{1:T})}{p(x_{1:T} \mid y_{1:T})\, q(x^*_{1:T} \mid x_{1:T})}\right].$$

If $x^*_{1:T}$ is accepted, set $x'_{1:T} = x^*_{1:T}$ and otherwise set $x'_{1:T} = x_{1:T}$. The choice of q is such that we have a low autocorrelation time.

3.2 Gibbs Sampling

Suppose we have access to conditional densities of each variable, given the other ones. With Gibbs sampling, we update each coordinate x_t in turn by sampling the conditional density $p(x_t \mid x_{t\neq j}, y_{1:T}) \propto f(x_{t+1} \mid x_t)\, f(x_t \mid x_{t-1})\, g(y_t \mid x_t)$. On a general note, it can be beneficial to combine various MCMC updates, provided this is done correctly.

4. CONDITIONAL SEQUENTIAL MONTE CARLO

This is a modern method for sampling state sequences in general state space models (Andrieu, Doucet and Holenstein (2010)). It is an MCMC update of $x_{1:T}$ to $x'_{1:T}$. This approach uses Sequential Monte Carlo (SMC) to create a set of candidate sequences. In the SMC pass, one of the particles at each time t is set to x_t. An advantage of conditional SMC is that it can use *all* particles generated by an SMC pass to construct a set of candidate sequences. Advantages of conditional SMC include: ability to sample state sequences with strong temporal dependencies and it is computationally inexpensive. The disadvantages are: poor scaling to models with high-dimensional states. Also, typical choice of target densities does not consider all observed data.

5. HAMILTONIAN MONTE CARLO

This method is due to Duane, Kennedy, Pendleton and Roweth (1987). Suppose we have a distribution of interest: $\pi(q) = (1/Z)\exp(-U(q))$ with $U(q)$ the "potential energy". We introduce a vector of auxiliary "momentum" variables p with same dimensionality as q, define a "kinetic energy" $K(p)$. Common choice is $K(p) = p^T p/2$ ie., Gaussian momentum variables. The "Hamiltonian" is $H(p,q) = U(q)+K(p)$; density of (q,p) proportional to $\exp(-H(p,q))$, q and p are independent; the marginal density for q is π.

The Hamiltonian dynamics are defined by the differential equations describing the evolution of (q, p) in "time" t.

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}.$$

continued ▶

For $H(q,p) = U(q)+K(p)$ with $K(p) = p^T p/2$,

$$\frac{dq_i}{dt} = p_i, \quad \frac{dp_i}{dt} = -\frac{\partial U}{\partial q_i}.$$

(q, p) evolves to (q^*, p^*) by applying Hamiltonian dynamics. Key properties of this mapping are (1) it leaves H invariant and (2) it preserves volume in (q, p). In most cases these equations cannot be solved exactly so we need to approximate them with some stepsize ϵ. A common choice of discretization is the "leapfrog" algorithm.

The proposal follows a trajectory that is not a random walk. After L leapfrog steps, we expect the proposal to be at a distance ϵL from the initial point. Tuning ϵ is a challenge: on one hand, we want the discretization to be stable, on the other hand we want the proposal to lie far from the initial point. Tuning L is another challenge: we would like the proposed point to be close to independent of the initial point. Finally, the dynamics can exhibit "doubling back", a problem that has been addressed in practice in numerous ways, eg., NUTS of Hoffman and Gelman (2014).

6. REFERENCES

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society B 72, 269?342.

Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987). Hybrid Monte Carlo. Phys. Lett. B 195 216?222.

Hoffman, M. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research 15 1593-1623.

Denis Erkal

FITTING NOISY DATA WITH NOISY MODELS

ABSTRACT

In astrophysics, we sometimes need to compare noisy data with noisy models. I will present one such scenario in which this arises when the model has shot noise since it is an N-body simulation consisting of a discrete set of particles. I will briefly explain the astrophysical motivation of the problem.

I will then explain how we fit these models to the data using an MCMC as long as the shot noise in the model is sufficiently small and discuss some pitfalls. I will also present a toy model which has the same characteristics to better explore this problem.

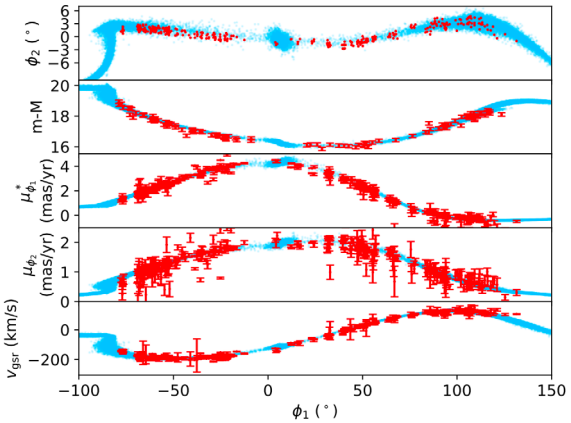


Figure | Here we show an astrophysical example of a noisy model fit to noisy data. The data is shown in red and shows measurements of the 3d position and 3d velocity of stars orbiting the Milky Way. The blue points show an N-body model simulated in a potential similar to that of our Galaxy which can reproduce the trends seen in the data. To perform the fit, we must ensure that the noise in the model (shot noise from a finite number of particles) is much smaller than the uncertainties in the data.

Joaquín García de la Cruz

USING MCMC METHODS FOR FITTING DATA: FROM FAILING TO LEARNING

ABSTRACT

In this talk, I will talk about my personal experience using MCMC methods during the course of my PhD.

Firstly, I will show through different examples how the physical interpretation of the fitted values and their uncertainties was crucial in solving problems in my research. Sometimes, failing to fit the data – especially due to high uncertainties – led me to new insights about a system I was struggling to understand, even taking that project into a whole new direction.

Secondly, I will talk about situations where I still struggle fit the data, and I will share my insights as to why.

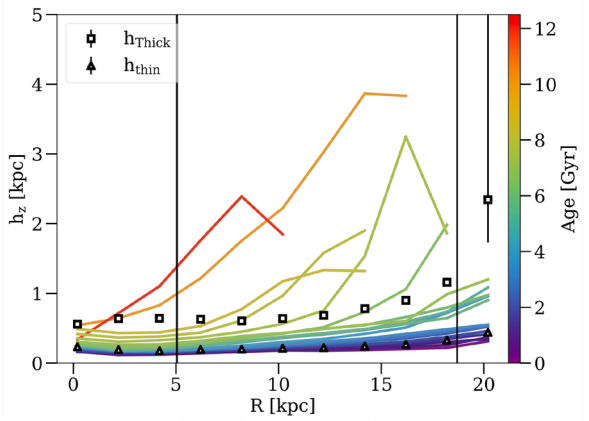


Figure | For a simulated disc galaxy, values of the scale-height against radius for the thin disc (triangles), thick disc (squares), and mono-age populations (solid lines) colour-coded by age. The vertical black line on the left represents where the galactic disc starts. A second vertical black line represents where fits for the thick disc's scale-heights show very high uncertainties.

Minas Karamanis

PARALLEL, BLACK-BOX AND GRADIENT-FREE INFERENCE

ABSTRACT

Modern astronomical and cosmological analyses have been revolutionised by the use of Markov Chain Monte Carlo (MCMC) methods for Bayesian inference and data analysis. However, the characteristics of most astronomical models (e.g. computational cost, intractable derivatives) pose substantial challenges to many state-of-the-art MCMC methods. To this end, we introduce a new method called Ensemble Slice Sampling for parallel, black-box and gradient-free inference and its Python implementation zeus in order to facilitate astronomical analyses more efficiently.

1. INTRODUCTION

Bayesian inference and data analysis have become an integral part of modern science and astronomy in particular. This is partly due to the capacity of Markov Chain Monte Carlo (MCMC) methods to generate samples from posterior distributions. As the amount of collected data and astronomical observations increases so does the complexity and computational cost of the theoretical models that are developed in order to account for those observations. The increased model complexity, both in terms of high dimensionality (e.g. large number of parameters) and non-linearity of the physics involved, has led to posterior distributions that are difficult to sample from even when state–of–the–art methods are used. Peculiar posterior distributions can also emerge in cases in which the data are sparse (e.g. exoplanet radial velocity measurements).

MCMC sampling is often the main computational bottleneck of modern astronomical pipelines. We will argue that this is mostly because of some odd aspects and characteristics of astronomical and cosmological models. First of all, and perhaps most importantly, astronomical models are often quite slow, meaning that one evaluation, for a specific set of parameters, can take from few seconds to a few minutes (e.g. one model evaluation of the CAMB or CLASS Boltzmann codes which are used in most cosmological analyses require 5 − 10 s in a modern CPU). Unlike the models that are used in other fields of study, most astronomical models are not differentiable. This means that gradient–based methods such as Hamiltonian Monte Carlo (HMC) [5] and its variations cannot be applied.

continued ▶

Based on the aforementioned characteristics we can compile a list of properties of the ideal sampler for astronomical applications. In order to handle the high computational cost of the models the ideal MCMC sampler needs to be able to scale favourably with the number of available CPUs. Another constraint comes from the non-differentiable nature of the models. The sampler needs to be able to generate samples from the target distribution without relying on the availability of the score function (i.e. gradient of the log–probability). Finally, in order for the sampler to be versatile and able to sample efficiently from a wide range of target distributions it needs to be able to maintain a high level of sampling efficiency even in cases of high correlation between parameters (i.e. highly skewed or anisotropic distributions) without requiring any hand-tuning from the user.

So far, only a limited number of samplers have been widely applied to astronomical and cosmological problems. Those include: the Random Walk Metropolis (RWM) algorithm (i.e. Metropolis-Hastings with symmetric normal proposal distribution), the Affine Invariant Ensemble Sampler (AIES) using the Stretch move [2], the Differential Evolution Monte Carlo (DEMC) [6] method. Unfortunately, RWM requires great amounts of hand-tuning and even when it is tuned this is done assuming that a single proposal scale is optimal for the whole parameter space. While AIES does not require any hand-tuning it does not scale well with the number of dimensions and it is prone to mode collapse issues. DEMC is optimal only for the case of normal distributions and its efficiency drops rapidly when this condition is violated.

2. METHODS AND RESULTS

Based on the above discussion there is clearly a need for new methods that could complement the existing ones. This is of paramount importance for cases in which the aforementioned methods usually struggle to generate samples (e.g. higher dimensional problems, mildly non-linear correlations, multimodal distributions). To this end we introduce Ensemble Slice Sampling (ESS) [3], a parallel, black–box and gradient–free method for Bayesian inference in correlated and multimodal distributions. ESS is the amalgamation of two separate methods, Slice Sampling and Ensemble MCMC, in such a way as to complement each other. Slice Sampling is based on the idea that instead of sampling from a distribution with density $p(x)$ one can sample uniformly from the area under the graph of $f(x) \propto p(x)$. Slice Sampling is a univariate scheme that has a single hyper–parameter (i.e. the initial length scale) that is auto–tuned during the run. By design, Slice Sampling performs no rejections at the level of the Markov chain. Ensemble MCMC utilises a collection of parallel and interacting chains, called walkers, which preserve the product density $P(x_1, x_2, \ldots, x_N) = p(x_1)p(x_2) \ldots p(x_N)$ without the individual walker trajectories being independent, or even Markovian. ESS functions by performing Slice Sampling updates along directions chosen using the ensemble of walkers. There are arbitrary many ways of defining those direction vectors and each one yields a new algorithm with different characteristics, strengths and weaknesses. However, few of them such as the Differential move or the Global move are general enough that they can

continued ▶

be applied to most problems [3, 4]. The final method is parallel and scales linearly with the number of available CPUs (i.e. for $n_{CPUs} \leq n_{Walkers}/2$, requires no hand-tuning or gradient information, and it is affine invariant, meaning that its performance is insensitive to linear correlations). Empirical tests demonstrated that compared to RWM, Slice Sampling, AIES, DEMC and Sequential Monte Carlo [1], ESS samples at least as efficiently and in many cases more efficiently by a significant margin.

3. DISCUSSION

During the past couple of decades, MCMC methods have experienced a substantial rise in popularity in the fields of astronomy and cosmology. Given the increased sophistication of the astronomical models the aforementioned rise is expected to continue making the development of novel sampling methods ever more important. We argue that ESS is one such method that could facilitate astronomical research for the next decade. A Python implementation of ESS, called zeus [4], is publicly available at **https://github.com/minaskar/zeus** with detailed documentation that can be found at **https://zeus-mcmc.readthedocs.io**.

REFERENCES

1. Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.
2. Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science*, 5(1):65–80, 2010.
3. Minas Karamanis and Florian Beutler. Ensemble slice sampling. *arXiv preprint arXiv: 2002.06212*, 2020.
4. Minas Karamanis, Florian Beutler, and John A Peacock. zeus: A python implementation of ensemble slice sampling for efficient bayesian parameter inference. *arXiv preprint arXiv:2105.03468*, 2021.
5. Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
6. Cajo JF Ter Braak. A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. *Statistics and Computing*, 16(3):239–249, 2006.

Josh Speagle

AN INTRODUCTION TO NESTED SAMPLING

ABSTRACT

Quantifying model uncertainty and performing model selection within a Bayesian framework is becoming an ever-larger part of scientific analysis both within and outside of astronomy. I will present a brief introduction to Nested Sampling, a complementary framework to Markov Chain Monte Carlo approaches that is designed to estimate marginal likelihoods (i.e. Bayesian evidences) and posterior distributions, outline some of their pros and cons, and briefly discuss more recent extensions such as Dynamic Nested Sampling. I will also briefly highlight "dynesty", an open-source Python package designed to make it easy for researchers to applying Nested Sampling approaches to various "black box" likelihoods present in their work.

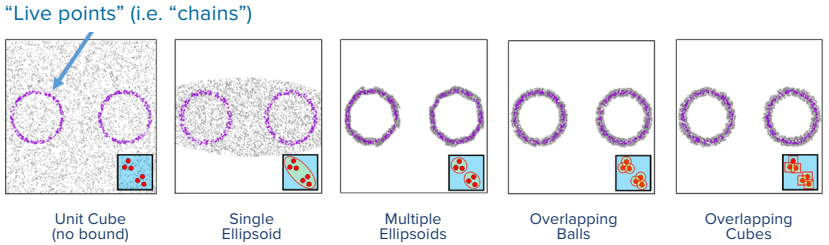


Figure | A schematic figure highlighting the various strategies used to approximate the shape of the constrained prior distribution in Nested Sampling algorithms using the current positions of live points (i.e. chains), shown in purple. Samples from the approximation are shown in grey. A small illustration is also shown in the bottom-right portion of each panel. The highlighted strategies, from left to right, are: no bounds, a single ellipsoid, multiple ellipsoids, many overlapping spheres, and many overlapping cubes.

Payel Das

USING ADAPTIVE HAMILTONIAN MONTE CARLO FOR TRAINING ARTIFICIAL NEURAL NETWORKS

1. BACKGROUND

Astronomical datasets are undergoing a rapid growth in their size and complexity due to past and ongoing surveys. In response to this, astronomers are developing machine learning tools to help extract and analyse the wealth of information within them. Figure 1 shows how the number of abstracts in the astronomy ArXiv that mention 'machine learning' has increased over the last 30 years.

The main challenge in making a machine learning model fully Bayesian however, is the number of parameters that need to be fit. Even a simple artificial neural network (ANN) can have ~ hundred parameters. Hamiltonian Monte Carlo (HMC) is a Bayesian sampling method, perfectly suited to fitting machine learning models because of its ability to deal with a large number of potentially correlated parameters. Neal recognized this potential back in 1995 [5], but it was only after a much later review that HMC became more mainstream [6]. The research objective here is to briefly explain the HMC method and then show its application to an astrophysical problem.

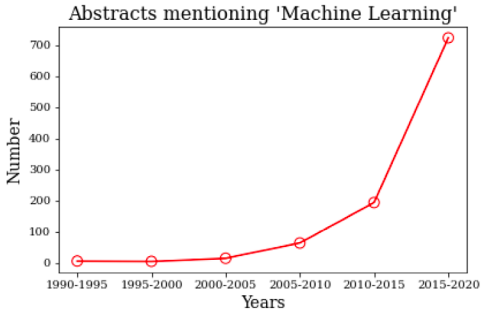


Figure 1 | The number of abstracts in the astronomy ArXiv mentioning "machine learning" in the abstract.

2. ADAPTIVE HAMILTONIAN MONTE CARLO

In uncertainty analysis, we are not interested in the peak posterior density but rather expectation values of the posterior density such as the mean, to which both the posterior density and volume contribute. The posterior density peaks at some $p = p_{peak}$ but the volume element increases as p increases. Therefore their product peaks somewhere between the peak posterior density and a very large volume and is called the 'typical set' [1]. Increasing the number of dimensions rapidly focuses the typical set into a very narrow region. This quickly becomes a challenge to sample from, in particular, with a procedure that is completely random.

continued ▶

In Hamiltonian Monte Carlo (HMC), Hamtilonian dynamics is used to introduce a component of the exploration of the typical set that is deterministic, by solving the equations of motion. To apply HMC, you need to define a phase space defined by positions and momenta. The parameters of the model become positions, and these need to be supplemented by auxiliary momentum parameters, q, to complement each dimension of the target parameter space. We can now lift the target distribution onto the joint probability distribution in phase space by choosing auxiliary momenta randomly, conditioning on the position coordinates,

$$P(p,q) = P(q \mid p)\, P(p)\,. \tag{1}$$

This ensures that the target distribution can be easily recovered if we marginalize out the momentum. Also, the trajectories exploring the typical set of the target distribution can easily be recovered by projecting the trajectories in the phase-space distribution.

We can write the posterior density $P(p,q)$ in terms of the Hamiltonian, $H(p,q)$,

$$P(p,q) = e^{-H(p,q)}\,, \tag{2}$$

where in physics, the value of the Hamiltonian at any point in phase space is the energy there. Rewriting in terms of the Hamiltonian,

$$H(q,p) \equiv -\log P(p,q) \tag{3}$$

$$\equiv -\log \pi(q|p) - \log P(p)\,, \tag{4}$$

where, $-\log \pi(q \mid p)$ is the kinetic energy $K(q,p)$, while $-\log P(p)$ is the potential energy $V(p)$. The typical set in phase space (which can be physically interpreted as e.g. the stable orbit for a satellite orbiting a planet) is then explored by using Hamilton's equations:

$$\frac{dp}{dt} = \frac{\partial H}{\partial p} = \frac{\partial K}{\partial q} \tag{5}$$

$$\frac{dq}{dt} = -\frac{\partial H}{\partial q} = -\frac{\partial K}{\partial p} - \frac{\partial V}{\partial q} \tag{6}$$

There are two key choices in implementing HMC: 1) the choice of the conditional probability distribution over the momentum and 2) the integration time at each step. If we integrate for only a short time, we do not optimize the deterministic exploration of the typical set. However, if we integrate for too long, trajectories eventually return to previously explored neighbourhoods.

continued ▶

The No-U-Turn sampler [NUTS, 3] dynamically changes the integration time by considering the boundaries of a trajectory. The termination criterion is satisfied when any further integration would bring the ends of the trajectory closer together. Empirically, this has been shown to work extremely well [1].

3. ASTROPHYSICAL APPLICATION

There is a class of stars called 'red giants' that can be seen at different epochs of our Milky Way's existence and all over space. They are therefore very good probes for dissecting the formation history of our galaxy. A small sample (~ thousands) of these stars have excellent estimates of their mass from asteroseismology surveys that measure how these stars oscillate. These mass estimates can be combined with stellar evolution models to obtain very precise ages. Martig et al. (2016) [4] showed that the masses can be empirically well predicted from a handful of spectroscopic properties that are being measured for millions of stars. Stellar evolution models can then be used with these mass estimates to predict ages for millions of stars rather than just a thousand. Their simple polynomial model however underestimates ages on the high-age end and as they only probe relations between spectroscopic parameters and masses, they still need to rely on stellar evolution models to get ages. They also do not use new distance information that has become available from the Gaia spacecraft. Here we present a new model for predicting ages from the newest data. The work presented here is discussed in more detail elsewhere [2].

3.1 Building the training set

We build a training sample using red giant stars for which there are current mass constraints from asteroseismology in addition to astrometric (distances to stars) constraints as well as photometric (colours and brightnesses of stars), and spectroscopic (surface gravity, surface temperature and surface abundances of stars) data. We then use a brute-force approach to apply stellar evolution models to estimate the posterior densities of mass, age, distance, and metal content (or metallicity) for the stars in the training sample.

3.2 Building the ANN

An ANN consists of interconnected layers of neurons, which represent linear or non-linear transformations by an 'activation' function. We assume a feedforward ANN, where only neurons in adjacent layers are connected to one another. The first layer is the input layer comprising the same number of neurons as the number of inputs, $n_{\rm in}$. The central layers are hidden layers, each with potentially a different number of neurons per hidden layer, $n_{\rm hid}$. The final layer is the output layer with the same number of neurons as the number of outputs, nhid. We assume linear activation functions for the input and output layers and a tanh activation function for the hidden layer, which maps variables ranging from $-\infty$ to ∞ to a domain extending between -1 to 1. The predicted outputs vector for each star, $\mathbf{y}$, is calculated from the predicted inputs vector for each star, $\mathbf{x}$, by

$$\mathbf{y} = (\mathbf{w}_{\rm h,out}\ \tanh(\mathbf{w}_{\rm h,in} \cdot \mathbf{x} + \mathbf{b}_{\rm h,in})) + \mathbf{b}_{\rm h,out}\,, \qquad (7)$$

where $\mathbf{w}_{h,\rm in}$ is a $n_{\rm hid} \times n_{\rm in}$ matrix of weights, $\mathbf{b}_{h,\rm in}$ is a length-$n_{\rm hid}$ vector of biases, $\mathbf{w}_{h,\rm out}$ is a length-$n_{\rm hid}$ vector of weights, and $\mathbf{b}_{h,\rm out}$ is a length-$n_{\rm out}$ vector of biases.

continued ▶

This architecture has $n_\theta = n_{\rm hid}(n_{\rm in} + 2) + n_{\rm out}$ model parameters.

We investigate which inputs from the astrometric, photometric, and spectroscopic data can be used as predictors of the outputs of the stellar evolution models (mass, age, distance, and metallicity of a star) using the Spearman Rank Correlation Coefficient. We choose nine inputs, ten neurons in the hidden layer, and have four outputs and therefore our ANN (figure 2) has 114 parameters.

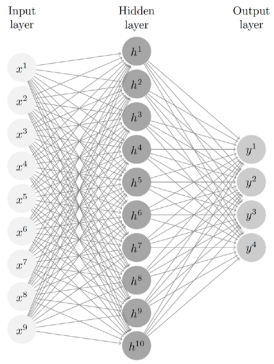


Figure 1 | Architecture of ANN assuming $n_{\rm in} = 9$ (light grey), $n_{\rm out} = 4$ (middle grey), and one hidden layer with $n_{\rm hid} = 10$ neurons (dark grey).

The posterior distributions of the ANN parameters, θ, can be estimated using Bayes' law

$$P(\theta|\tilde{\mathbf{u}}) = \frac{P(\tilde{\mathbf{u}}|\theta)\,P(\theta)}{P(\tilde{\mathbf{u}})} \qquad (8)$$

where $P(\tilde{\mathbf{u}}|\theta)$ is the joint likelihood of all measured stellar properties, $\tilde{\mathbf{u}}$ (includes measured inputs, $\tilde{\mathbf{x}}$, and measured outputs, $\tilde{\mathbf{y}}$), given the model parameters, $P(\theta)$ is the prior on the model parameters, and $P(\tilde{\mathbf{u}})$ is the distribution of the measured stellar properties. $P(\tilde{\mathbf{u}})$ is the same for every model and can be ignored. The likelihood of the star's measured properties, $P(\tilde{\mathbf{u}}|\theta)$, is assumed to be the product of the likeli-hoods of each measured stellar property. We assume Gaussian measurement uncertainties. The NUTS sampler in PyMC3 is used to train the ANN on 80% randomly selected stars. The remaining 20% is used as an independent test of the ANN.

3.3 Making predictions with the ANN

Once we have obtained posterior distributions for the parameters of the ANN, $P(\theta|\tilde{\mathbf{u}})$, we can calculate posterior predictive distributions for selected predicted stellar properties of new stars, $\mathbf{y}_N$, given the training sample, $\tilde{\mathbf{u}}$, and the new measured inputs, $\tilde{\mathbf{x}}_N$, i.e.

$$P(\mathbf{y}_N|\tilde{\mathbf{u}},\tilde{\mathbf{x}}_N) = \iint P(\mathbf{y}_N|\theta,\tilde{\mathbf{x}}_N)\,P(\theta|\tilde{\mathbf{u}})\,P(\tilde{\mathbf{x}}_N|\mathbf{x}_N)\,{\rm d}\theta{\rm d}\mathbf{x}_N\,. \qquad (9)$$

$P(\mathbf{y}_N|\,\theta, \tilde{\mathbf{x}}_N)$ is the probability of the new predicted outputs given some set of model parameters and new true inputs, $P(\theta\,|\,\tilde{\mathbf{u}})$ is the posterior distributions of the ANN model parameters evaluated in Section 3.2, and $P(\tilde{\mathbf{x}}_N|\,\mathbf{x}_N)$ is the distribution of new

continued ▶

predicted inputs given the new measured inputs. Marginalizing the product of these probabilities over the parameters of the ANN and the new inputs give the posterior predictive distributions on the new outputs. Generating the posterior predictive distributions therefore does not re-quire one to engage with the stellar evolution models.

Figure 3 compares the means and standard deviations of the output logarithmic age distributions predicted by the ANN against those of the measured output distributions for the training and testing samples. Applying the ANN takes just over a minute on a single core to calculate posterior predictive distributions for mass, age, distance, and metallicity. This is comparable to the time taken to apply stellar evolution models using a Bayesian approach for just a handful of stars.

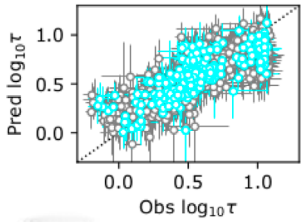


Figure 3 | A comparison between predicted and measured mean logarithmic ages. The predicted output distributions are generated by the ANN and the measured output distributions are generated by the stellar evolution models for the training (grey) and testing (cyan) samples.

4. SUMMARY

Stellar evolution models were applied to a training sample of giant stars with astrometric, photometric, and spectroscopic data, and asteroseismology mass estimates to obtain Bayesian estimates of their masses, ages, distances, and metallicities. Supplementing the astrometric, photometric, spectroscopic data with chemical abundances, we train a Bayesian ANN to learn the relationship between these inputs and mass, age, distance, and metallicity. The ANN on average reproduces the stellar evolution model estimates for mass, age, distance, and metallicity with similar uncertainties, but takes far less time to run.

REFERENCES

1. Betancourt M., 2017, arXiv e-prints
2. Das P., Sanders J. L., 2019, MNRAS, 484, 294
3. Hoffman M. D., Gelman A., 2014, Journal of Machine Learning Research, 15, 1593
4. Martig M., et al., 2016, MNRAS, 456, 3655
5. Neal R. M., 1995, PhD thesis, University of Toronto
6. Neal R. M., 2012, arXiv e-prints, pp arXiv–1206

Linghan Li

APPLICATIONS OF MCMC BAYESIAN SAMPLING METHODS

In this article, four commonly used Bayesian sampling strategies (Metropolis-Hasting MCMC, Affine-invariant ensemble MCMC, Hamiltonian Monte Carlo and Nested sampling) are introduced, and examples given to demonstrate their relative performance.

1. METROPOLIS-HASTING SAMPLER

Metropolis-Hastings is a MCMC method for sampling from the posterior distribution by using a proposal distribution to perform a random jump, then accepting or rejecting proposed moves between current and proposed states with some probability. The proposal function is normally chosen to be symmetric but can be asymmetric if the sampling distribution is truncated or you have prior knowledge of its skew. A test problem of a 2D Gaussian with different correlations (range from 0 to 0.99) is given below to show the results of MH MCMC.

The MH can successfully recover the posterior distribution but the efficiency decreases as the correlation increases to one. In this high correlation case, the proposed next step is very likely to fall down the sharp cliff. The theoretical requirements for using MH are quite minimal. However, it's fundamentally a random walk. There's no logic informing how large and which direction the jumps should be, given the current position. So the limitation for MH MCMC are normally related to low efficiency, as it requires a user-defined step size and it is hard to tune especially in high dimensional case. Furthermore, the step size is fixed, i.e. it is not proportional to the density distribution.

The MH can successfully recover the posterior distribution but the efficiency decreases as correlation trends to one. In this high correlation case, the proposed next step very likely to fall down the sharp cliff. so it stuck at this spot. The theoretical requirements for using MH are quite minimal. However, it's fundamentally a random walk. There's no logic informing how large and which direction the jumps should be given the current position. So the limitation for MH MCMC are normally related to low efficiency. As it requires user defined step size and it is hard to tune especially in high dimensional case. Furthermore, the step size is fixed, it is not proportional to the density distribution.

2. AFFINE-INVARIANT ENSEMBLE SAMPLER

The second scheme is affine invariant ensemble. It use linear transformation to transform the (θ_1, θ_2, ...) coordinate state into a new state space of ensembles, while the state is still proportional to the posterior. If some walkers catch the a probability

continued ▶

maximum, the others will move towards them and explore the surrounding space efficiently, and in such a way that the step size is proportional to the density of the target distribution. It can step around awkward distributiondistributions, and views the densities as equally difficult. The comparecomparison of the MH and affine-invariant ensemble samplers in term of efficiency is shown in Figure 1. This method is robust for many badly-scaled posteriors.

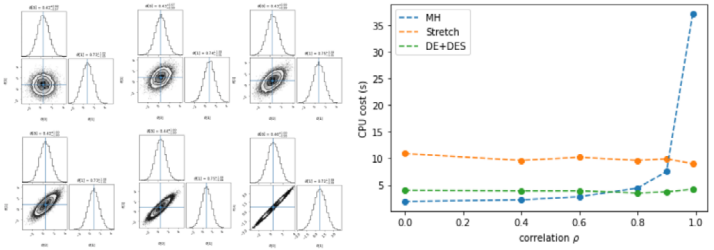


Figure 1 | MH results for the 2D Gaussian with different correlations (left), and their corresponding CPU cost. This is compared with the affine-invariant ensemble scheme (right).

However, for some particular distributions such as a donut shape distribution, if two walkers are in the region of high density, the proposed position is highly unlikely to fall back to the high-density region. As shown in Figure 2, the basic affine-invariant ensemble scheme fails to recover the 25D donut-shape distribution. The proposed move is inefficient and it is not suitable for high-dimensional cases beyond the Gaussian distribution.

3. HAMILTONIAN MONTE CARLO (HMC) SAMPLER

HMC is a variant of MH algorithm and the way in which it differs from standard MH is by a using physics analogy to generate proposals. To understand the principle, we should imagine the path of a frictionless particle on a space which is related to posterior space. Because that space is essentially the inverse of posterior space then we tend to flow towards the regions of the modes. As shown in Figure 2, the results for HMC for sampling from the 25D donut distribution is very promising as it runs much quicker (around 1min) than the affine ensemble sampler.

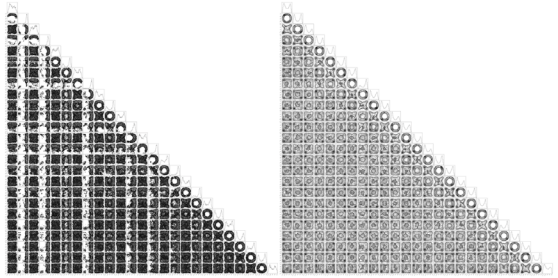


Figure 2 | Posterior distribution recovered by ensemble method (left) and HMC (right)

continued ▶

The latter needs hours. HMC recovers the space very well. 100 leapfrog steps is used, and the integration time delta is tuned until the acceptance rate is 0.688.

However, the aforementioned method finds it difficult to handle multi-modal posterior distribution as it may get trapped at a local maximum. This is where nested samplers come in to play.

4. NESTED SAMPLER

The nested sampling algorithm is different from the MCMC method. It approximates the marginal likelihood directly. An example of 3-modal 5D Gaussians with deep valleys between each modal would be problematic for the three aforementioned schemes but the nested sampler handles it very well.

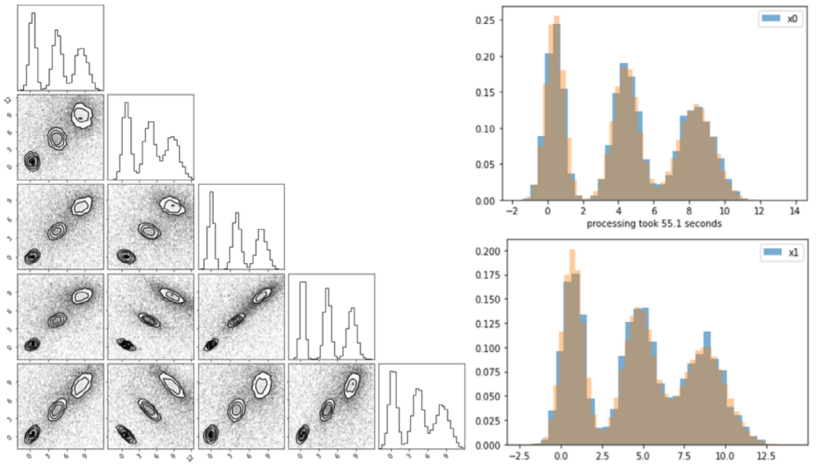


Figure 3 | 3-mode 5D Gaussian distribution generated by the nested sampler, compared to the true distribution (amber).

Another limit of MCMC is that it typically focusses on the peak of the posterior and explores in that vicinity, with low sampling of the tails of the distribution. This is not a problem for parameter estimation, but it can be when calculating evidence. In nested sampling methods, the evidence is immediately obtained by summation.

FACULTY OF ENGINEERING AND PHYSICAL SCIENCE

University of Surrey
Guildford, GU2 7XH, UK

surrey.ac.uk