NPB, MCMC, GPE and other funny acronyms

Complicated methods for simple tasks

A. Pastore, C. Barton, $\sum_{i=1}^{\infty}$ M. Phys$_i$

Department of Physics, University of York, Heslington, York, YO10 5DD, UK

October 30, 2019

Machine Learning et Physique Nucléaire, Orsay, 29-30 Oct. 2019

Introduction

How people *see* machine learning?

What is machine learning (ML)?

EDF fitting...

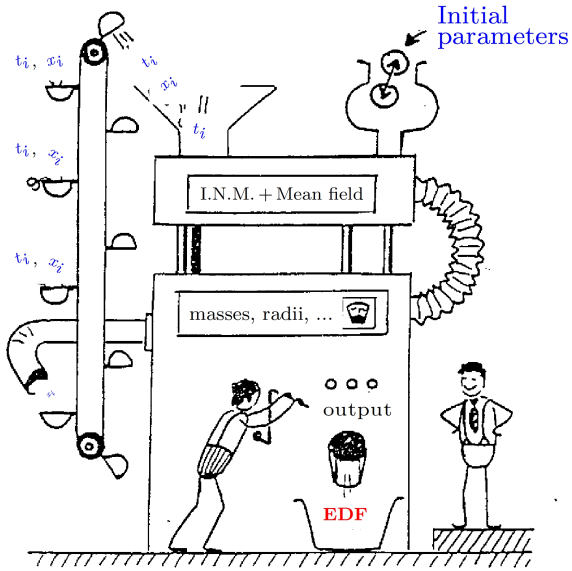


Figure: Picture by J. Dechargé, from "Approches de champ moyen et au-delà", J.-F. Berger, École Joliot-Curie: "Les noyaux en pleine forme", 1991.

ML is essentially a *complicated* parameter estimate.

Nuclear models

Main task in nuclear physics is to adjust parameters in theoretical models.

Example 1: Liquid Drop (LD)

$$B_{th}(N,Z) = a_v A - a_s A^{2/3} - a_c \frac{Z(Z-1)}{A^{1/3}} - a_a \frac{(N-Z)^2}{A} - \delta \frac{\mathrm{mod}(Z,2) + \mathrm{mod}(N,2) - 1}{A^{1/2}} ,$$

Example 2: Duflo-Zucker (DZ)

$$B_{th} = a_1 V_C + a_2(M+S) - a_3 \frac{M}{\rho} - a_4 V_T + a_5 V_{TS} + a_6 s_3 - a_7 \frac{s_3}{\rho} + a_8 s_4 + a_9 d_4 + a_{10} V_P \ .$$

[J. Duflo and A. P. Zuker; Phys. Rev. C 52 (1995) R23]

My (our) goal

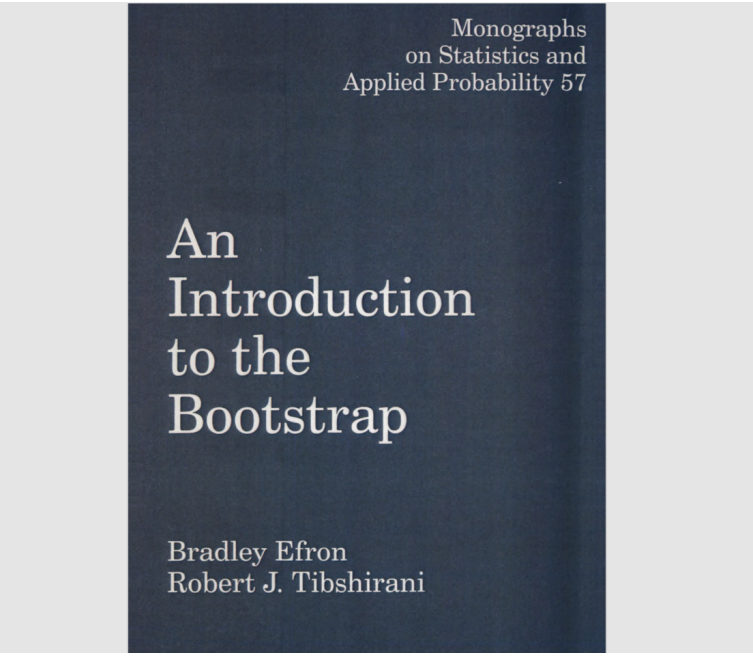
- Estimate the parameters a_i in the best possible way
- Estimate errors and correlations among parameters
- Improve the models

Non Parametric Bootstrap (NPB)

Bootstrap is a simple Monte-Carlo with no *smart* acceptance/rejection method

Hypothesis

A sample data originates from a *population* and they keep its features!



[A.P. *An introduction to bootstrap for nuclear physics*. J. Phys. G 46, 052001(2019)]

Parameter estimate (how NPB does the dirty job for you)

(Classical) Set up

Estimate 5-parameters of LD model This is a *linear* model. We estimate parameters as

$$\chi^2 = \sum_{N,Z \in \text{data-set}} \frac{[B_{exp}(N,Z) - B_{th}(N,Z)]^2}{\sigma^2(N,Z)} \ .$$

($\sigma^2(N,Z) =$ for simplicity)

- Minimise χ^2
- Build Hessian matrix (parameter derivatives) [Numerically dangerous!]
- Build Jacobian matrix for the model around minimum [Numerically dangerous!]
- Require explicit modelling of data-correlations in σ^2 matrix! [Complicated!]
- Error analysis

[Barlow, R. J. Statistics: a guide to the use of statistical methods in the physical sciences . John Wiley & Sons.(1993).]

A simple bootstrap solution

1. We do 1 fit and we obtain residuals

$$\chi^2 = \sum_{N,Z \in \text{data-set}} \left[B_{exp}(N,Z) - B_{th}(N,Z) \right]^2 .$$

$$B_{exp} = B_{th}(\mathbf{x}, \mathbf{p}_0) + \mathcal{E}(\mathbf{x}) ,$$

2. We bootstrap the residuals $\mathcal{E}(\mathbf{x}) \rightarrow \mathcal{E}^*(\mathbf{x})$
3. We create *new* sets of experimental binding energies

$$B*_{exp} = B_{exp} + \mathcal{E}^*(\mathbf{x}) ,$$

4. We fit new masses with our model

$$\chi^2 = \sum_{N,Z \in \text{data-set}} \left[B^*_{exp}(N,Z) - B_{th}(N,Z) \right]^2 .$$

5. Repeat the operation 10^4 times
6. Make nice histograms

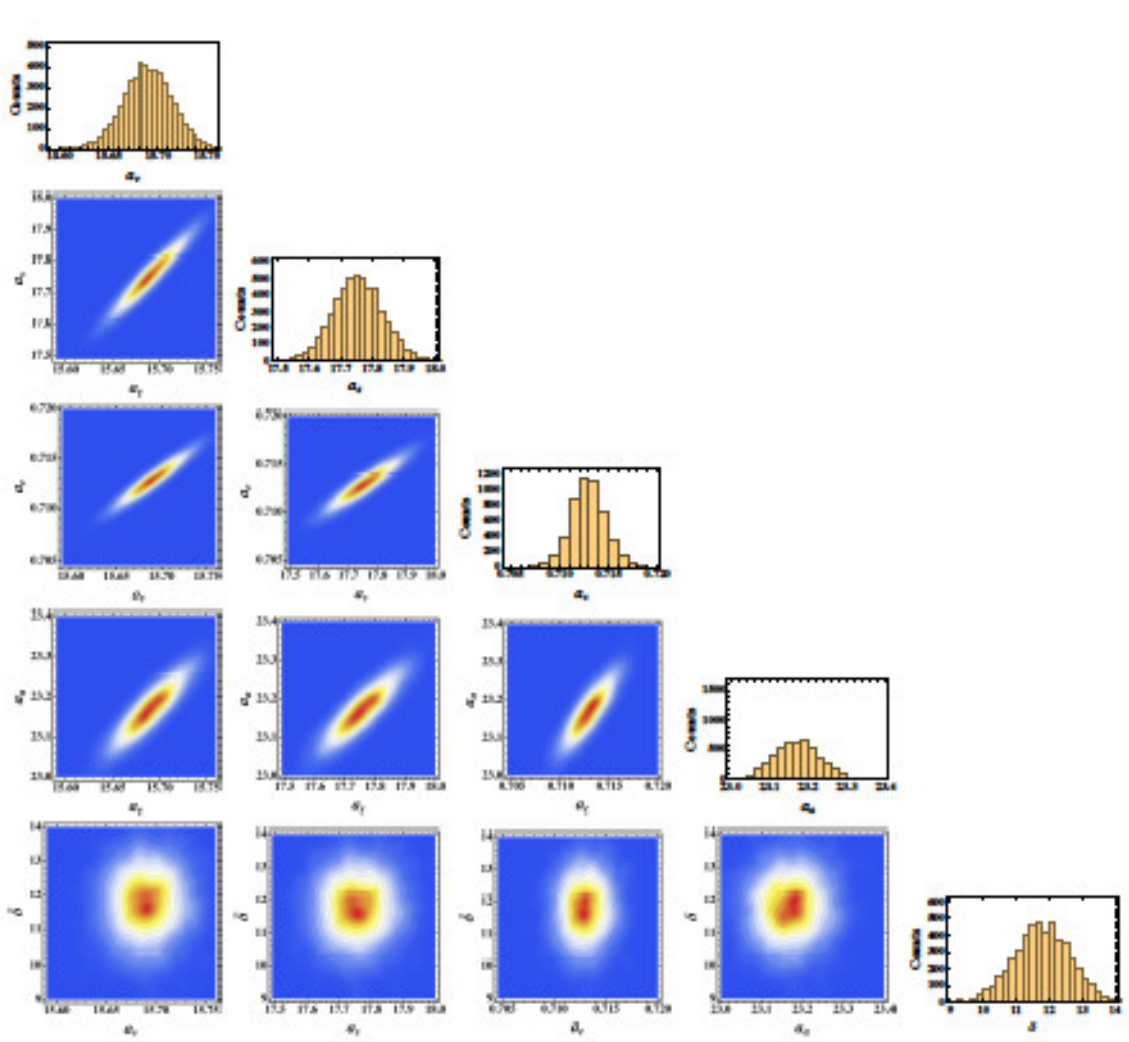
Results

Parameter	[MeV]	Error [MeV]
a_v	15.69	± 0.025
a_s	17.75	± 0.08
a_c	0.713	± 0.002
a_a	23.16	± 0.06
δ	11.8	± 0.9

We get the *same* results using linear fit procedure (good benchmark).

Corner plots for free

The data-set of 10^4 can be seen as a corner plot (no marginalisation!)



Advantages

- I get corner plots for free
- I do not need to calculate derivatives in parameter space! Covariance comes out automatically from 2D histograms!
- I do not need any *parabolic* approximation to do error propagation. I have access to full Monte Carlo error propagation for free! (I have actually 10^4 models I can use now!)

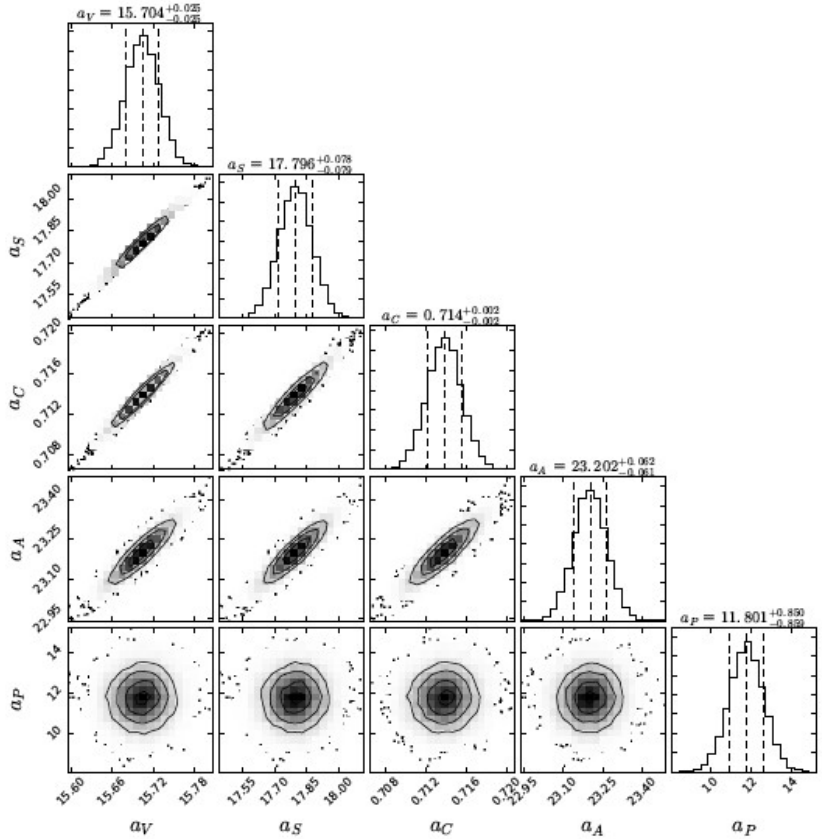
Problems? (not really... let's move on)

- We assumed $\sigma = 1$. Using data dependent sigmas... not easy
- We have an homogenous χ^2. Not the case in EDF fitting

A smarter Monte Carlo

By equipping a *memory* and a *smart* way of choosing (Metropolis) we obtain Markov-Chain-Monte-Carlo (MCMC).

- More efficient than NPB
- More advanced MCMC on the market $\rightarrow$ speed up in the process
- We get same results as NPB

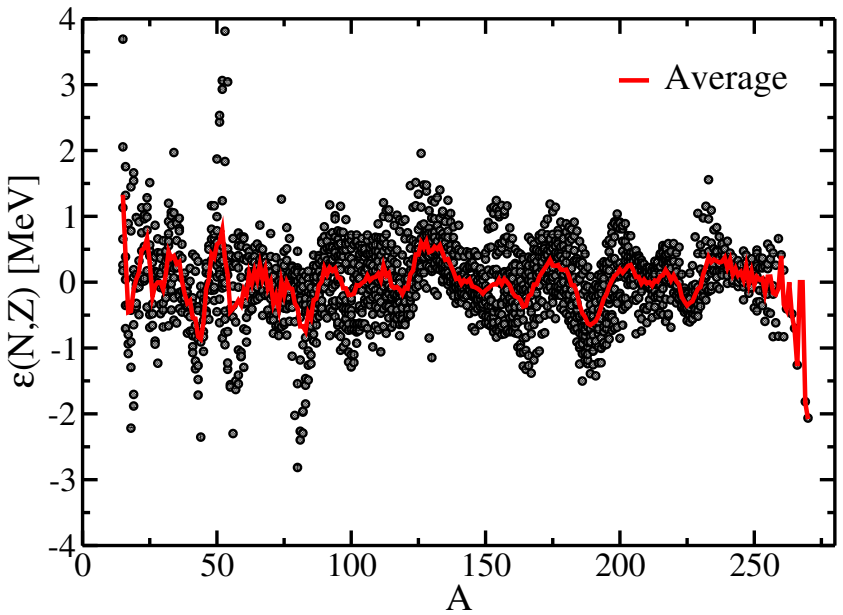


[M. Shelley, P. Becker, A. Gration and AP (2018). *Advanced statistical methods to fit nuclear models.* arXiv:1811.09130.]

Let's go back to our hypothesis

The residuals are assumed to be normally distributed $\mathcal{N}(0, \sigma)$ $\sigma = 0.572$ keV.

$$B_{exp} = B_{th}(\mathbf{x}, \mathbf{p}_0) + \mathcal{E}(\mathbf{x}) \ ,$$

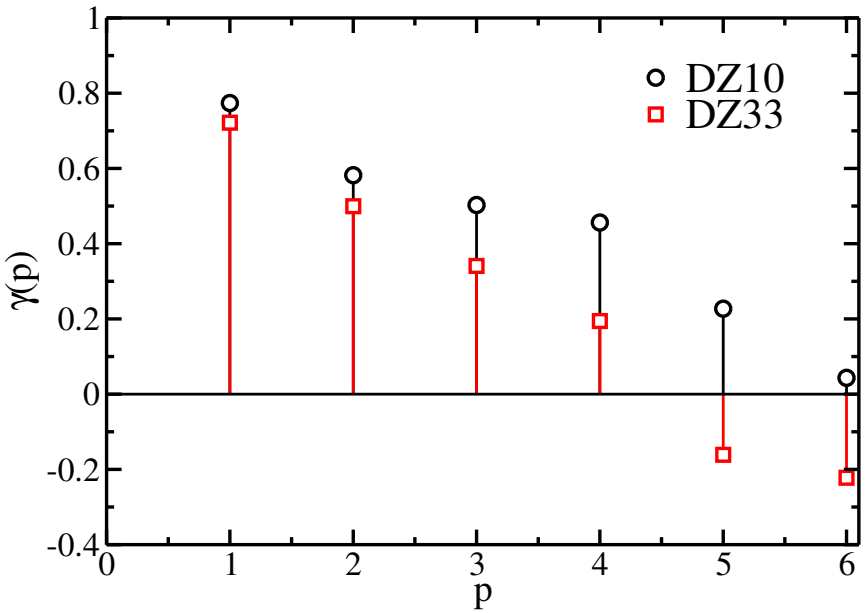


Residuals are not normally distributed (Kolmogorov test)

$$\sigma_A^2 = \frac{1}{N_A} \sum_{Z+N=A} \left(\mathcal{E}(N,Z) - \mathcal{E}_A(A)\right)^2 \ .$$

We work on σ_A^2

We reduce to a 1-D problem



BB in 2D

We have repeated the analysis on the mass table (no averaging) using a BB methods in 2D. The results do not changing remarkably

How to handle correlations in data?

Bootstrap can handle correlations

Several variants:

- Frequency Domain Bootstrap [G. F Bertsch and D. Bingham (2017). *Estimating parameter uncertainty in binding-energy models by the frequency-domain bootstrap*. Phys. rev. lett., 119, 252501. .]
- Block-Bootstrap
- Wild Bootstrap
- ....

MCMC can handle correlations?

It is a question for you! I have no idea.

Block-Bootstrap

Given a data-set composed by n elements $\{X_1, X_2, \ldots, X_n\}$, I consider an integer l satisfying $1 \leq l \leq n$. I define $\mathcal{B}_N$ overlapping blocks of length l as

$$\begin{aligned}
\mathcal{B}_1 &= (X_1, X_2, \ldots, X_l) \\
\mathcal{B}_1 &= (X_2, X_3, \ldots, X_{l+1}) \\
\ldots &= \ldots \\
\mathcal{B}_N &= (X_{n-l+1}, \ldots, X_n)
\end{aligned}$$

where $N = n - l + 1$.
We treat the blocks as uncorrelated. What size of blocks?

Statistic vs Systematic error

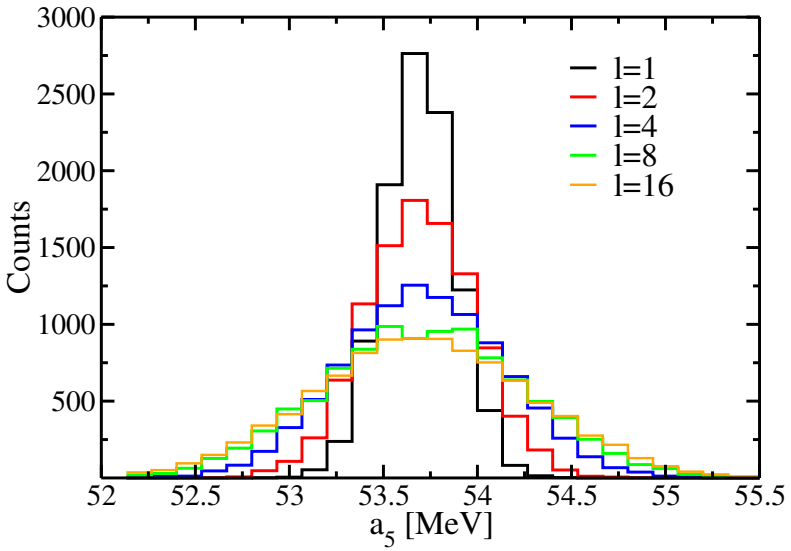
To assess the quality of our estimate we compare theory with experiment

NPB error propagation

	1σ	2σ	3σ
Full chart	13.6%	27.2%	39.5%
$50 \leq A < 150$	14.7%	26.8%	37.2%
$20 \leq Z \leq 50$	11.5 %	22.2%	31.4%
$A \geq 150$	14.8%	30.8 %	45.8%

BB estimate

	1σ	2σ	3σ
Full chart	34.5%	60.4%	77.9%
$50 \leq A \leq 150$	31.8%	55.5%	74.2%
$20 \leq Z \leq 50$	27.9 %	52.8%	71.9%
$A > 150$	39.9%	69.4 %	85.6%



[D. Neil, K. Medler, AP, C. Barton *Impact of statistical uncertainties on the composition of the outer crust of a neutron star* On my desk waiting to go....]

All very nice, but...

Back to square one

$$B_{exp}(N,Z) = B_{th}(N,Z) + \varepsilon(N,Z)$$

A major effort to get the best estimate for $\varepsilon(N,Z)$

We did not touch the residuals. What is the model has a *bias*?

Let's go to square two

$$B_{exp}(N,Z) = B_{th}(N,Z) + f_{ML}(N,Z) + \tilde{\varepsilon}(N,Z)$$

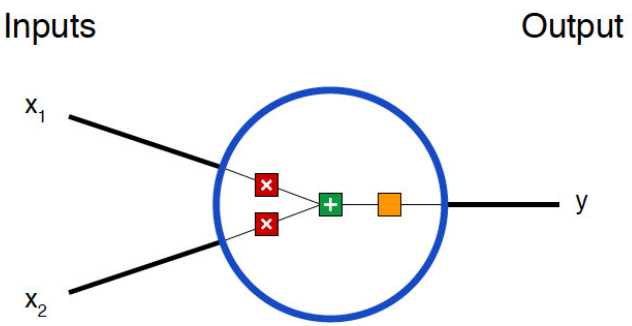
We add a correction to the model $f_{ML}(N,Z) \rightarrow$ Neural Network/ Gaussian Process Emulator

[L. Neufcourt, Y. Cao, W. Nazarewicz and F. Viens (2018). *Bayesian approach to model-based extrapolation of nuclear observables*. Physical Review C, 98(3), 034318.]

Neural Network (NN)

Definition

A NN is a system of connected algorithms (nodes/neurons) designed to mimic the working of a biological brain

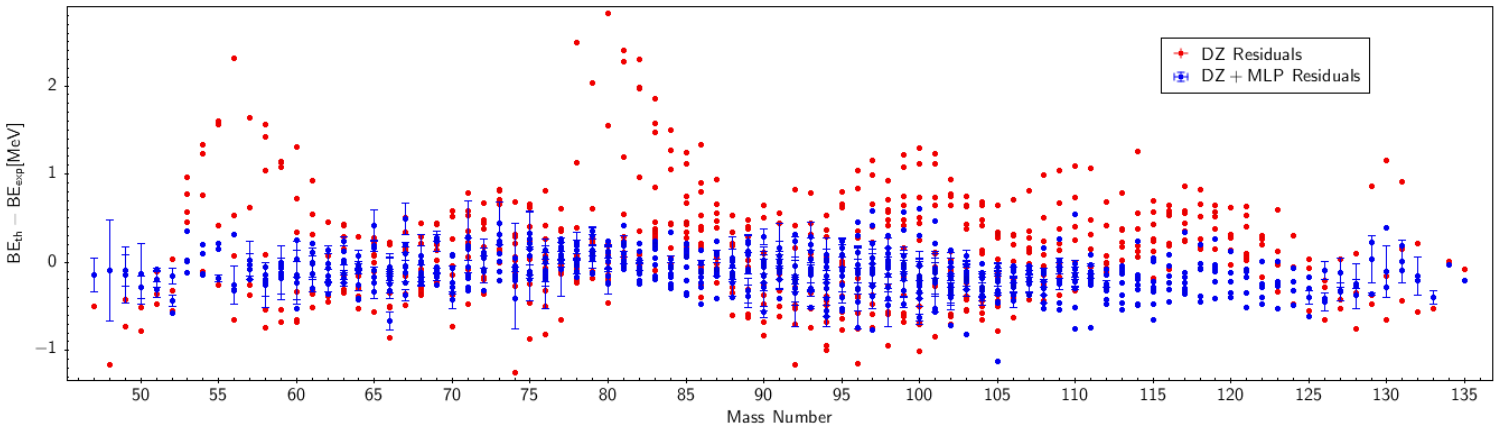


- Take inputs and multiply by weights $x_i \rightarrow x_i w_i$
- Sum $\sum_i x_i w_i$
- Pass to activation function $y = f(\sum x_i w_i + b)$
- Compare output $MSE = \frac{1}{n}\sum_i (y_{true} - y_{pred})^2$
- Find w_i to minimise MSE

[K. Hornik; Neural networks 4 (1991): 251-257 / K. Hornik, M. Stinchcombe, H. White; Neural Networks 2 (1989)359-366.]

DZ+NN

We aim at predicting masses in NS $25 \leq Z \leq 50$.
We use a Multi Layer Perceptron (easy to use... simple test) [weka]



Parameters (only for real *aficionados*)

Hidden layers = 2, with 45 nodes in the first and 84 nodes in the second layer.
Learning rate = 0.29
Momentum = 0.47
Training time = 6000
Percentage split = 66

[R. Utama and J. Piekarewicz; Phys. Rev. C 96 (2017): 044308.]

(Dis)Advantages

What do we conclude?

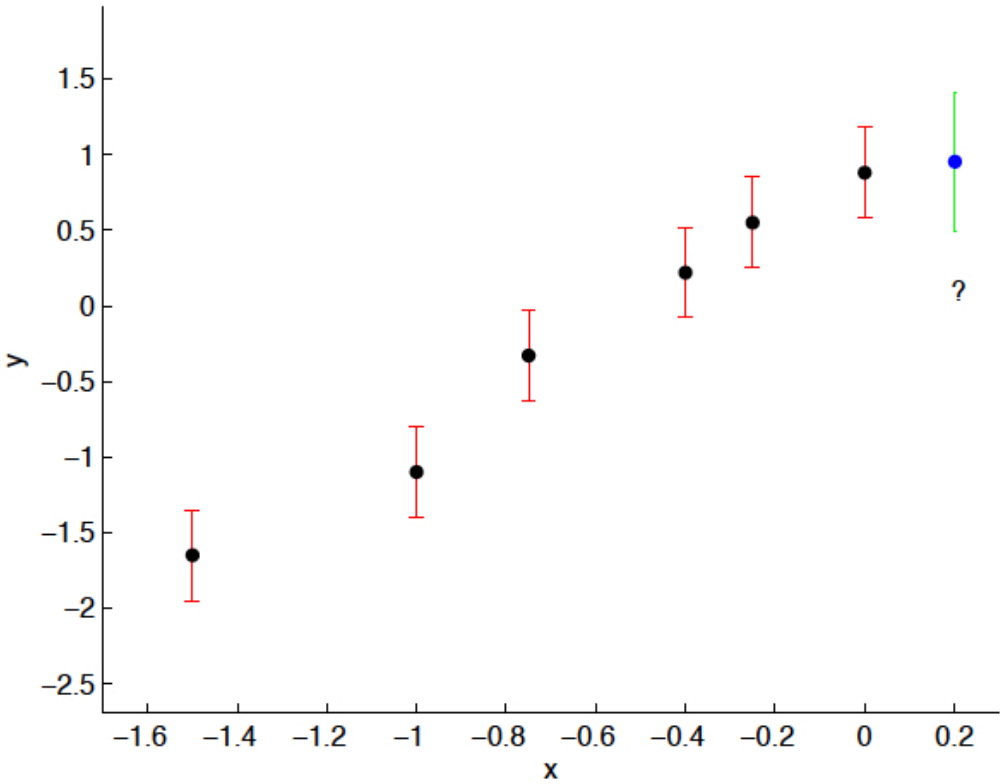
- NN is created to *learn* patterns in data (residuals)
- NN works nicely in *interpolations*.
- Residual are *more* similar to white noise

A word of caution

- Overfitting is a real danger (so many parameters in NN... no real rule!)
- NN can not predict new physics (*i.e* a new shell closure outside training region)
- Can we model *physically* what NN has found?
- At large extrapolations the NN goes to zero (we fit residuals)

Gaussian Process Emulator

Give a set of point (red). How to predict (blue), using no (little) assumptions on the data? (*i.e.* $f(x) = ax + b$)



$$y(x) = f(x) + \mathcal{N}(0, \sigma^2)$$

Definitions

A stochastic process is a collection of random variables indexed by some variable $x \in \mathcal{X}$

$$f = \{f(x) : x \in \mathcal{X}\}$$

$f(x) \in \mathcal{R}$ and $\mathcal{X} = \mathcal{R}^n$ [extension to multi-layers exists]
A Gaussian process is a stochastic process with Gaussian distribution

$$(f(x_1), \ldots f(x_n)) \approx \mathcal{N}(\mu(x), k(x, x'))$$

We can rescale the data so that $\mu = 0$ and we assume

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right] + \sigma_n^2 \delta(x, x')$$

l is correlation length. Obtained via Maximum Likelihood Estimator (MLE)

What's the value y^* in x^*?

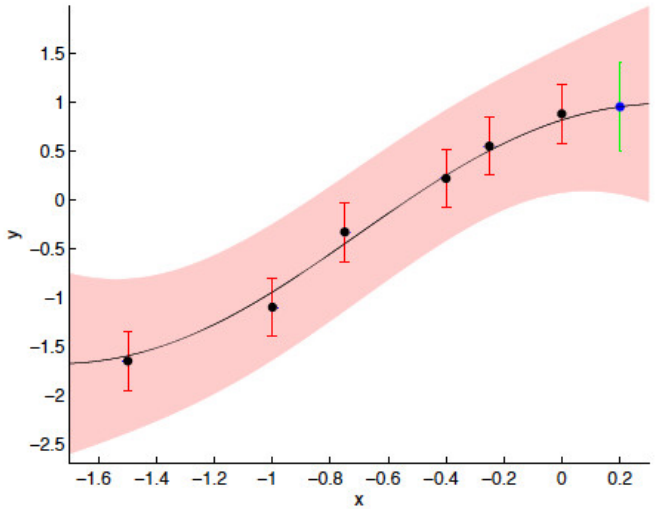
The conditional probability reads

$$y^*|\mathbf{y} \approx \mathcal{N}(K_* K^{-1}\mathbf{y}, K_{**} - K_* K^{-1} K_*^T)$$

where

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ \dots & \dots & \dots & \dots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = [k(x_*, x_1), k(x_*, x_2), \dots, k(x_*, x_n)] \quad K_{**} = k(x_*, x_*)$$



Application: learning a χ^2 surface

We aim at estimating the parameters of a model

Simplified Liquid Drop

$$B/A = a_v - a_s A^{-1/3}$$

- N=Z only (from ^{2}H to ^{100}Sn)
- No Coulomb/No pairing

$\rightarrow$ 2 D model... easy to make plots!

Least square fitting

$$\chi^2 = \sum_{nuclei} \left(\mathcal{O}^{exp} - \mathcal{O}^{th} \right)^2$$

No error assumed (for simplicity) on masses .... toy model!!!

$$a_v = 11.16\text{MeV} \quad a_s = 9.60\text{MeV}$$

GPE for χ^2

Main steps...

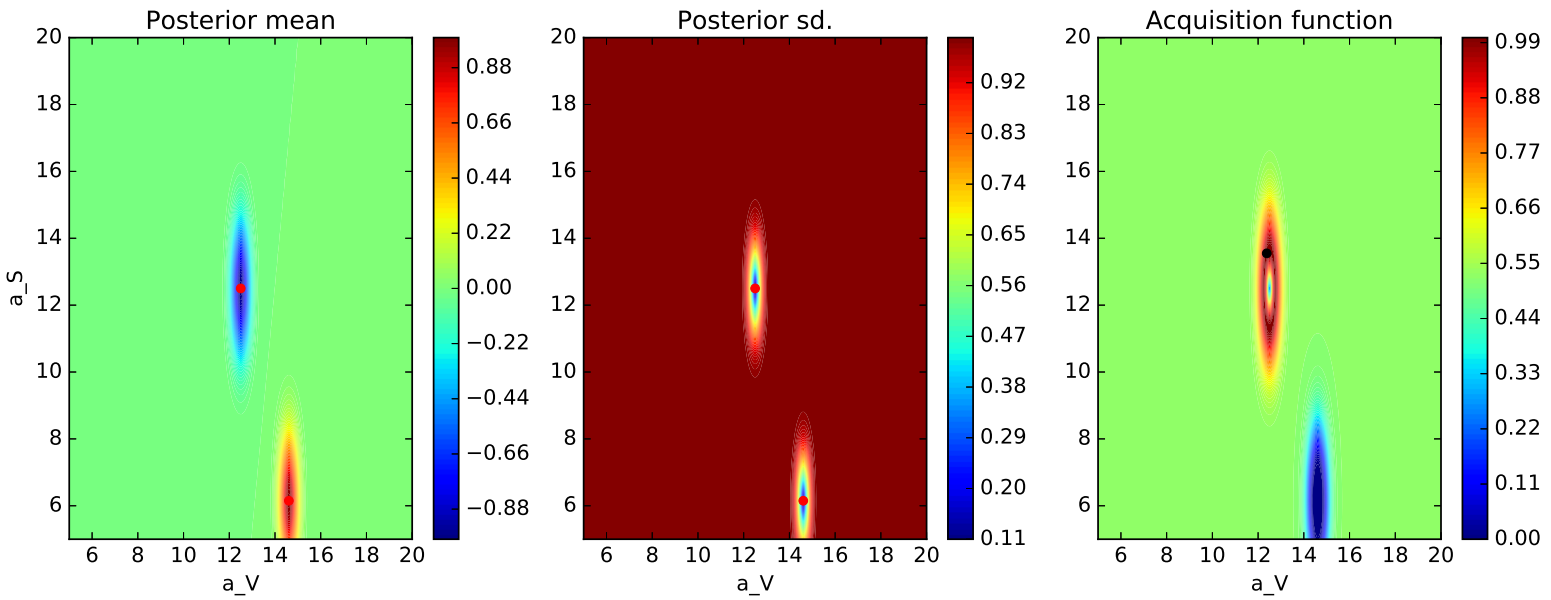
- Run GPE to emulate 2D surface of χ^2
- Iterative procedure guided by *acquisition* function
- Use the *real* simulation for a set of point selected by GPE
- Accumulate GPE iterations around minimum (not known a priori!)
- Refine the minimum using gradient method

Why?

- GPE scans the *whole* surface (contrary to a gradient
- GPE *should* detect more minima at once (our expectation)
- GPE *should* require a lower number of iterations compared to *standard* minimisation routines

[A. Gration and M. I Wilkinson, (2019). *Dynamical modelling of dwarf spheroidal galaxies using Gaussian-process emulation*. MNRAS 485(4), 4878-4892.]

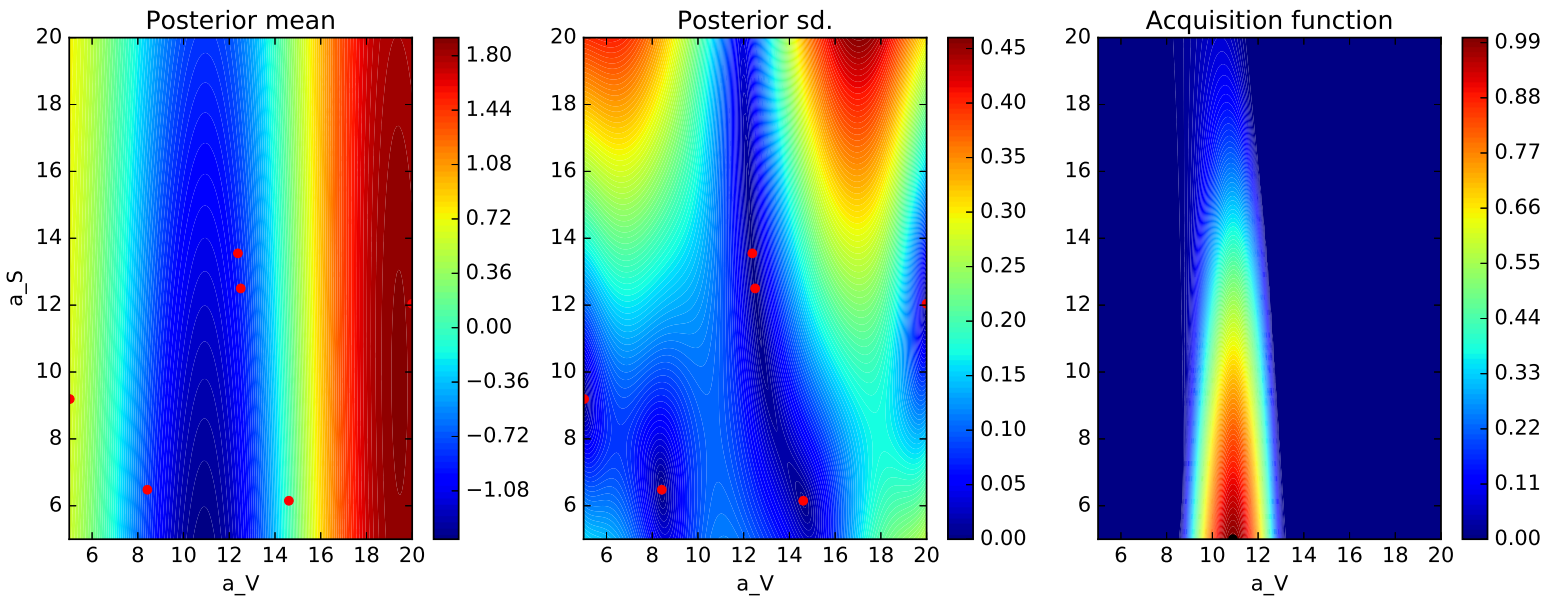
Initial point+1 point



Vocabulary

- Posterior mean $\rightarrow \chi^2$ surface produced by GPE
- Posterior sd. $\rightarrow$ predicted variance of the surface
- Acquisition function $\rightarrow$ next point required by GPE

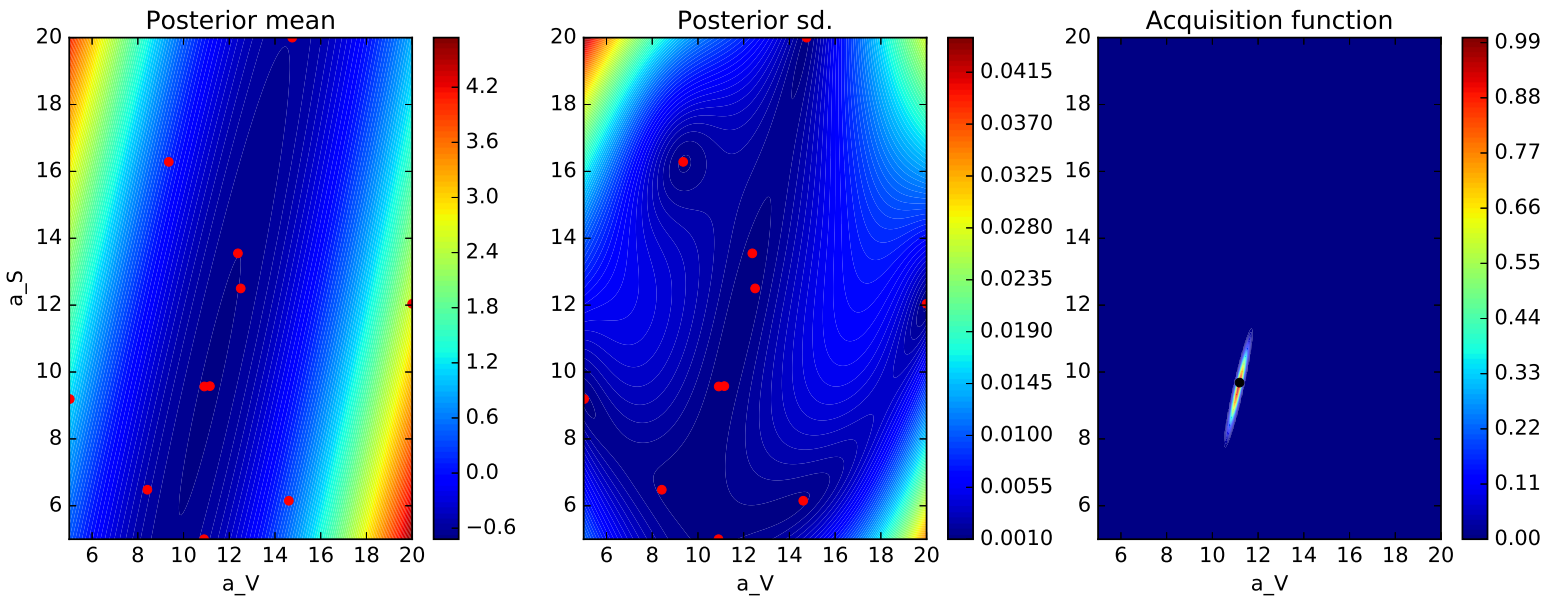
+5 points



Vocabulary

- Posterior mean $\rightarrow \chi^2$ surface produced by GPE
- Posterior sd. $\rightarrow$ predicted variance of the surface
- Acquisition function $\rightarrow$ next point required by GPE

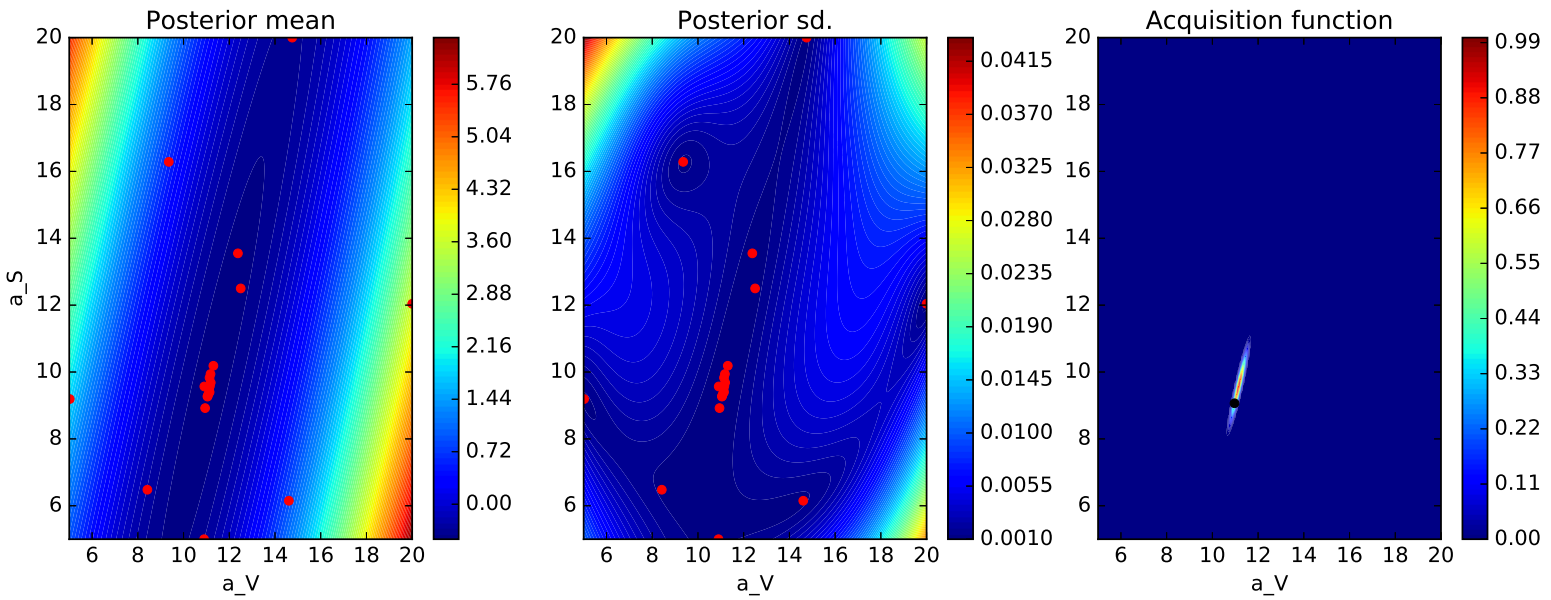
+10 points



Vocabulary

- Posterior mean $\rightarrow \chi^2$ surface produced by GPE
- Posterior sd. $\rightarrow$ predicted variance of the surface
- Acquisition function $\rightarrow$ next point required by GPE

+20 points

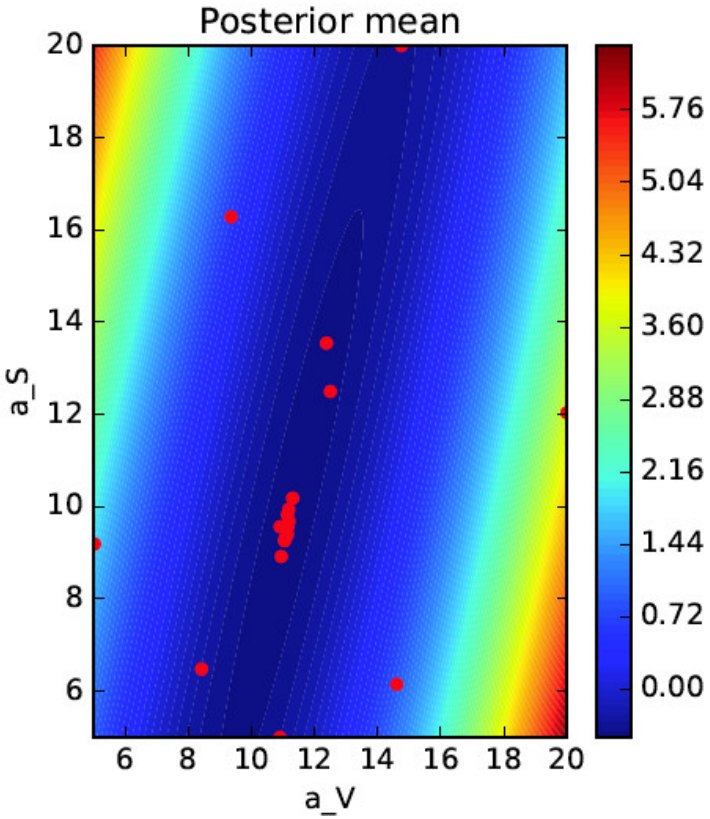


Vocabulary

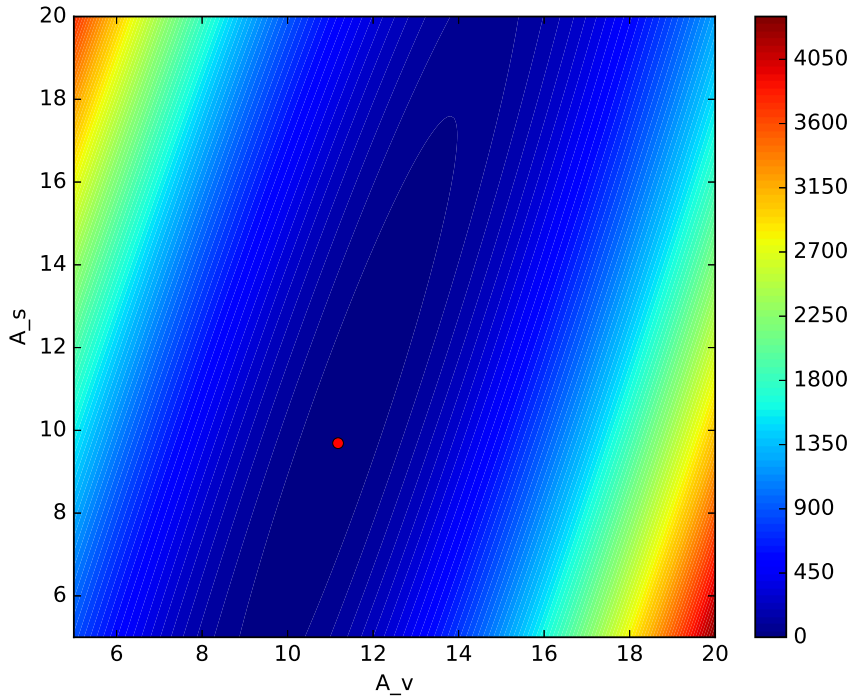
- Posterior mean $\rightarrow \chi^2$ surface produced by GPE
- Posterior sd. $\rightarrow$ predicted variance of the surface
- Acquisition function $\rightarrow$ next point required by GPE

GPE vs Exact

GPE



Exact



Conclusions

GPE can be a *real* advantage to learn a χ^2 surface $\rightarrow$ pre-optimisation process avoiding getting trapped in local minima (great expectations!)

Conclusions & Ideas

Several advanced statistical methods on the market

There is no free lunch!

- All methods rely on approximations/hypothesis. Do not use them as *black-boxes*
- NN/GPE are very powerful $\rightarrow$ need supervision of a physicist!
- There is no *intelligence*, but a sophisticated fitting (parameter estimate)

York team: shopping list

We aim at *learning* new methods and apply them to nuclear problems

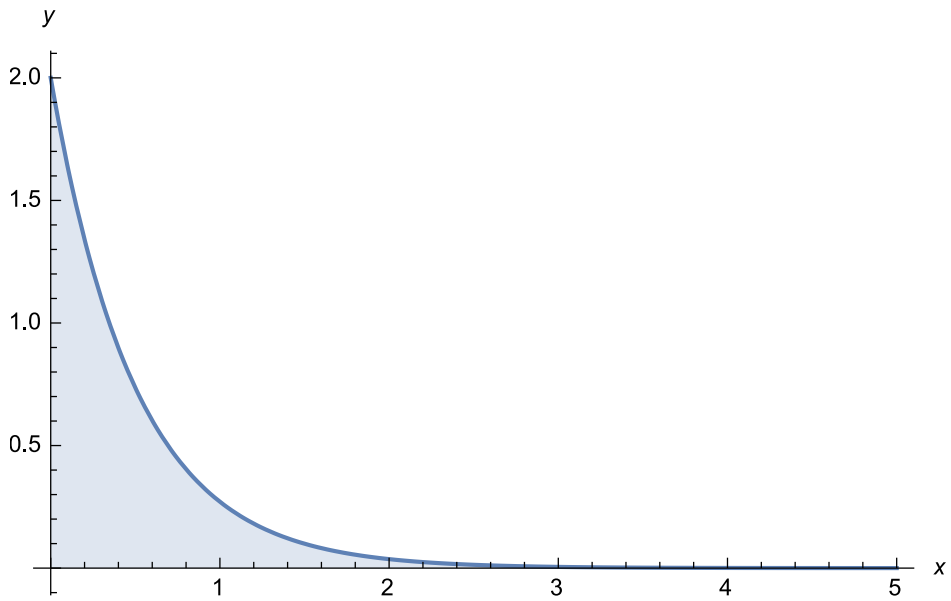
- (Dream) detector calibration
- (Plausible) apply GPE to fit functionals
- (Realistic) build simple NN/GPE to complete models and improve local extrapolations

Happy to share knowledge/ideas and desperately seeking for manpower (students)


THANK
YOU

Let's do an experiment!

Let's assume we have a population following an exponential distribution



$$PDF(x) = \lambda e^{-\lambda x}$$

Let's assume $\lambda = 2$

We run the experiment to obtain the data

value	0.068	1.649	0.058	0.165	0.522	0.040	1.078	0.512	0.354	0.449
position	1	2	3	4	5	6	7	8	9	10

Table: Random values extracted from exponential distribution with mean $\frac{1}{\lambda} = \frac{1}{2}$.

To calculate the mean of the parent distribution, I use the estimator

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} X_i = 0.489 \tag{1}$$

In this case the error on the man is know

$$\sigma_M = \frac{\sigma}{\sqrt{N}} = 0.154 \tag{2}$$

Not always so lucky....

Let's use Bootstrap to calculate the errors with no *prior* knowledge!

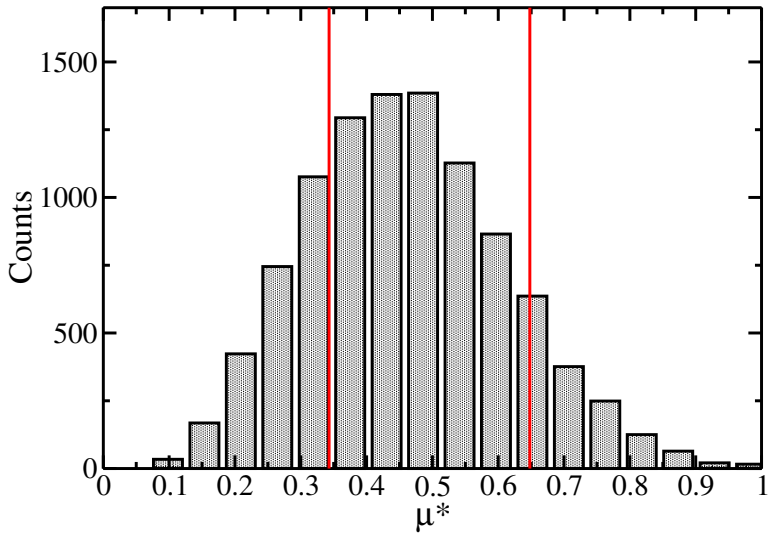
Bootstrap in action

1. Use a Monte Carlo to re-sample your data-set
   $X = \{0.068, 1.649, 0.058, 0.165, 0.522, 0.040, 1.078, 0.512, 0.354, 0.449\}$

$$
\begin{aligned}
X_1^* &= (0.068, 1.649, 1.078, 0.165, 0.522, 1.649, 0.058, 0.512, 0.354, 0.449) \ , \\
X_2^* &= (0.449, 1.649, 0.354, 0.165, 0.522, 1.649, 0.058, 0.512, 0.354, 0.068) \ , \\
X_3^* &= (0.068, 1.649, 1.078, 0.165, 0.522, 0.068, 0.058, 0.512, 0.354, 0.449) \ , \\
&\dots
\end{aligned}
$$

2. Apply the estimator to each of the sets X_n^*
3. Make an histogram and admire the *empirical* distribution of the estimator
4. Assume the *empirical* is equal to the *real* distribution of the estimator
5. Use 68% quantile to calculate error bars

Results



Use the empirical PDF!

We extract the mean of the histogram and 68% quantile $\bar{\mu^*} = 0.489^{0.159}_{-0.146}$. This is called Non-parametric Bootstrap (we made no assumption on the shape of the PDF)

Some warning

Big samples are always better. $N \geq 10 - 15$.
Re-sampling means to perform combinations.

$$\begin{pmatrix} 2n-1 \\ n \end{pmatrix} = \frac{(2n-1)!}{n!(n-1)!} \, . \tag{3}$$

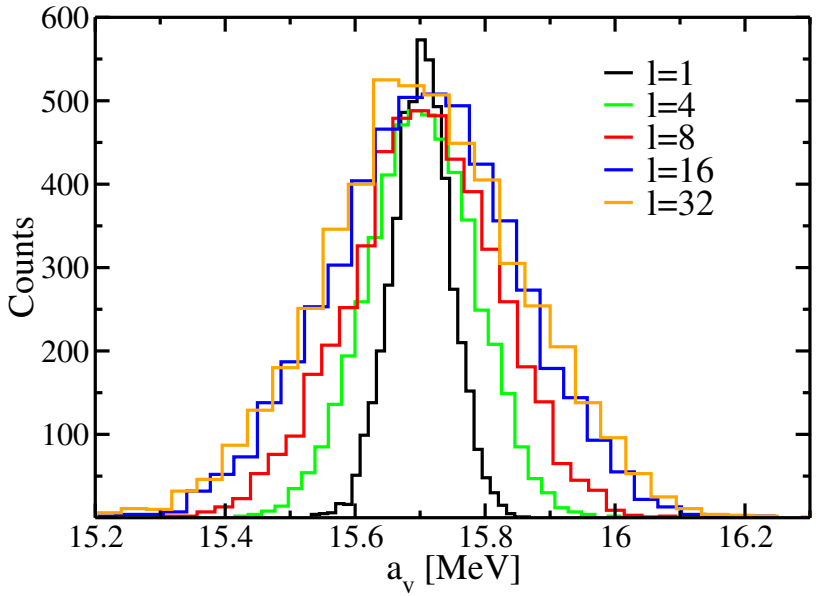
Repeated combinations add no info to the problem!

Some values

For n=5 we have 126 combinations.
For n=10 we have 92378 combinations.
For n=15 we have 77558760 combinations

How many MC you need? At least $10^3/10^4$ to avoid adding extra bias!
Very simple!

Results



We observe saturation... l should have same size as correlation length of the data.

Errors

Parameter	[MeV]	Error (uncorrelated) [MeV]	Error (correlated) [MeV]
a_v	15.69	± 0.025	± 0.14
a_s	17.75	± 0.08	± 0.44
a_c	0.713	± 0.002	± 0.009
a_a	23.16	± 0.06	± 0.35
δ	11.8	± 0.9	± 0.80

Errors are larger (1 order of magnitude) $\rightarrow$ it impacts error propagation on observables. If the model is wrong... it is still wrong, but with better error bars

Is there *any* effect?

The answer is on the next slide!