Bayesian Optimisation for Bayesian Evidence (BOBE)

Nathan Cohen

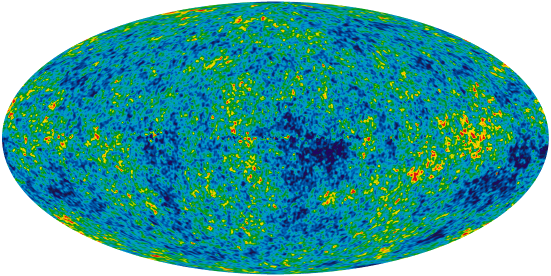
Based on work in progress with Jan Hamann and Ameek Malhotra

UNSW SYDNEY

University of New South Wales, Sydney, Australia

DSU Corfu, 10 September 2024

Broad Strokes: What is the problem?

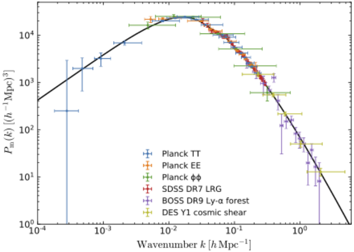


Parameter	Plik best fit	Plik [1]	CamSpec [2]	([2] − [1])/σ_1	Combined
$\Omega_b h^2$	0.022383	0.02237 ± 0.00015	0.02229 ± 0.00015	−0.5	0.02233 ± 0.00015
$\Omega_c h^2$	0.12011	0.1200 ± 0.0012	0.1197 ± 0.0012	−0.3	0.1198 ± 0.0012
$100\theta_{MC}$	1.040909	1.04092 ± 0.00031	1.04087 ± 0.00031	−0.2	1.04089 ± 0.00031
τ	0.0543	0.0544 ± 0.0073	$0.0536^{+0.0069}_{-0.0077}$	−0.1	0.0540 ± 0.0074
$\ln(10^{10} A_s)$	3.0448	3.044 ± 0.014	3.041 ± 0.015	−0.3	3.043 ± 0.014
n_s	0.96605	0.9649 ± 0.0042	0.9656 ± 0.0042	+0.2	0.9652 ± 0.0042
$\Omega_m h^2$	0.14314	0.1430 ± 0.0011	0.1426 ± 0.0011	−0.3	0.1428 ± 0.0011
H_0 [km s^{-1}Mpc^{-1}]	67.32	67.36 ± 0.54	67.39 ± 0.54	+0.1	67.37 ± 0.54
Ω_m	0.3158	0.3153 ± 0.0073	0.3142 ± 0.0074	−0.2	0.3147 ± 0.0074
Age [Gyr]	13.7971	13.797 ± 0.023	13.805 ± 0.023	+0.4	13.801 ± 0.024
σ_8	0.8120	0.8111 ± 0.0060	0.8091 ± 0.0060	−0.3	0.8101 ± 0.0061
$S_8 \equiv \sigma_8(\Omega_m/0.3)^{0.5}$	0.8331	0.832 ± 0.013	0.828 ± 0.013	−0.3	0.830 ± 0.013
z_{re}	7.68	7.67 ± 0.73	7.61 ± 0.75	−0.1	7.64 ± 0.74
$100\theta_*$	1.041085	1.04110 ± 0.00031	1.04106 ± 0.00031	−0.1	1.04108 ± 0.00031
r_{drag} [Mpc]	147.049	147.09 ± 0.26	147.26 ± 0.28	+0.6	147.18 ± 0.29

[NASA / WMAP Science Team, Planck2018 results: VI. Cosmological parameters]

How to Infer a Parameter

- Given a Cosmological Model $\mathcal{M}$ with parameters θ
  - Standard ΛCDM (n_s, t_0, τ, H_0, $\Omega_b h^2$, $\Omega_c h^2$, $\Omega_m h^2$)
  - ΛCDM with primordial features (ΛCDM + Amplitude, Frequency, Phase)
- Use Boltzmann Code (such as CLASS or CAMB)
- Get Theoretical Prediction



[ESA and the Planck Collaboration]

How to Infer a Parameter

- Given some observational data $\mathcal{D}$ we can calculate a likelihood $\mathcal{L}(\mathcal{D}|\theta, \mathcal{M})$
- This is the probability of the data given our model $\mathcal{M}$ and specific values of parameters θ

But Bayes! There's more

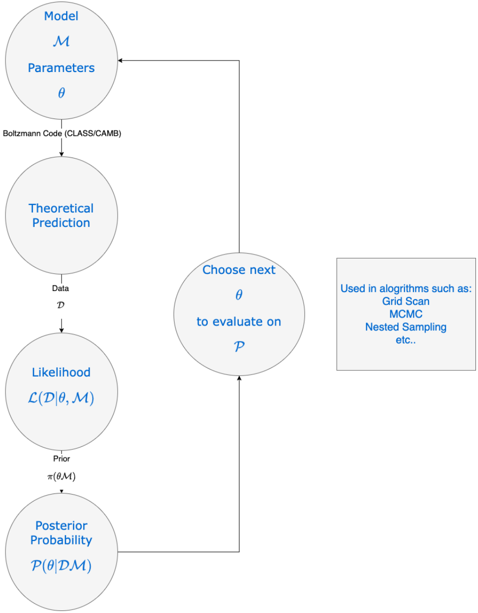
- Bayes Theorem:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$
$$Posterior = \frac{Prior \times Likelihood}{Evidence}$$

- With our Likelihood $\mathcal{L}$ and a prior $\pi(\theta|\mathcal{M})$ we can calculate a posterior probability $\mathcal{P}(\theta|\mathcal{D},\mathcal{M})$

- This allows us to update our likelihood and posterior probability based on new data

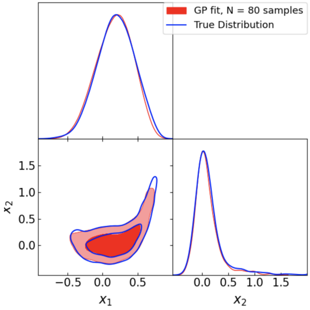
The circle of parameter inference



What do we get out of this?

- We will eventually reach a ‘termination criterion‘

- This should indicate we are confident in our predictions (to some level of accuracy)



The Good the Bad and the Ugly of MCMC

Good:

- Easy to implement
- Easy to parallelise
- Very litte extra computation required
- Scales mildly with number of dimensions
- Works great for most of cosmology (near-Gaussian posteriors)

Bad:

- Not very good at finding the maximum
- Requires a lot of function evaluations ($\mathcal{O}(10^4)$ for $N = \mathcal{O}(10)$)
- Ignores most of the information collected

Ugly:

- Struggles with not-nice posteriors (multi-modal, non-Gaussian, etc..)

Why is this an issue?

- If the likelihood is not easy to evaluate or not a "nice" shape MCMC doesn't work as well
- In these cases, MCMC's random sampling methodology can be problematic
- Can we do better by not ignoring the information from previous samples?

Is there a solution?

- Our goal is to design a more efficient method that learns the shape of the posterior
- We want to take advantage of what we already know by deterministically selecting our next "sample"
- Bayesian Optimisation may present a possible solution

Bayesian Optimisation

Fundamentally consists of two steps:

1. Regression - Guess the shape of the function based on the data

2. Next Step Selection - Decide at which point to evaluate the next function value

1) Gaussian Process Regression (GPR)

- A Gaussian Process is specified by a Kernel:

$$\mathcal{N}(\mu(x), K(x, x^{'}))$$

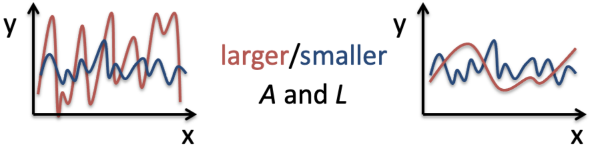
Where $K(x, x^{'})$ is the covariance function and $\mu(x)$ is the mean of the $\mathcal{N}$.

- A simple example covariance function:

$$K(x, x^{'}) = Ae^{\left[\frac{-(x-x^{'})^2}{L}\right]}$$

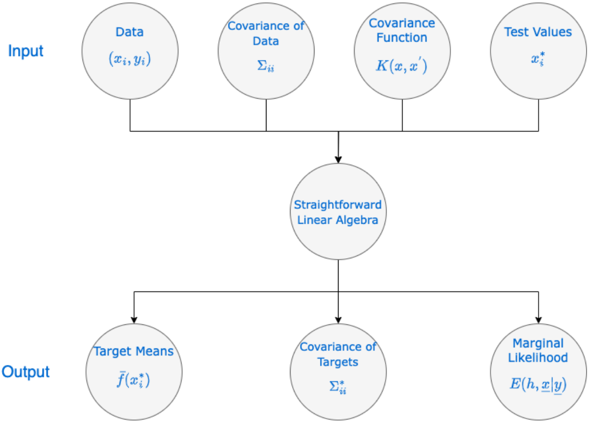
- Hyperparameters (h):
  - Covariance of the data: $K(x, x^{'})$ (relationship between parameters)
  - Prior Width: A (Certainty in the prediction)
  - Correlation Width: L (How much structure we expect in a given distance)

1) GPR Hyperparameters



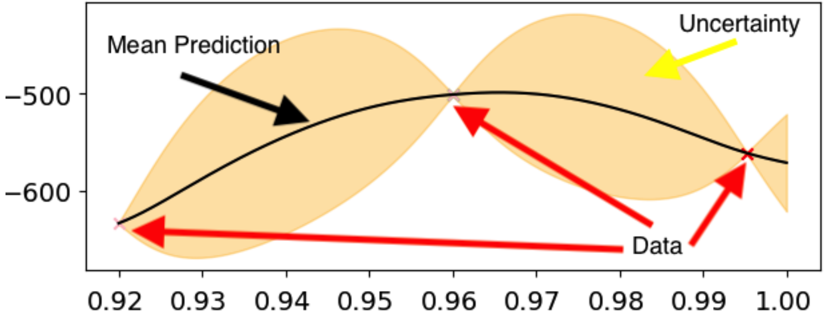
- Hyperparameters:
  - Prior Width: A (Certainty in the prediction)
  - Correlation Width: L (How much structure we expect in a given distance)

1) GPR Linear Algebra Machine



- $E(h, \underline{x}|\underline{y})$ is the probability of the model given the data
- Maximising E as a function of hyperparameters allows us to let the data decide the most appropriate GP!

1) GPR What we get out



- We are replacing an expensive/complex posterior with a cheap and easy GPR interpolation

Bayesian Optimisation

1. Regression - Guess the shape of the function based on the data
2. Next Step Selection - Decide at which point to evaluate the next function value

2) Acquisition Function

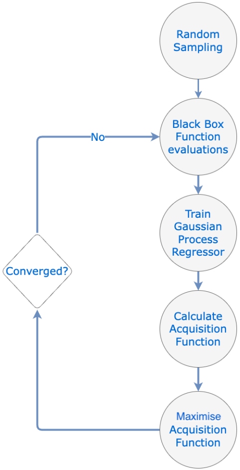
- GPR gives us a guess based on data
  - This helps quantify uncertainty, but we want to calculate the next sample in a smart way not randomly

- We want to recreate how an informed human agent would fit a function

- Define some function dependant on GPR mean and uncertainty that optimises a metric
  - i.e Expected Improvement (Largest Value)

What is Bayesian Optimisation?

- Bayesian Optimisation is a decision making framework
- The range of possible acquisition functions makes Bayesian Optimisation very adaptable



Bayesian Optimisation - Upsides and Downsides

Advantages:

- Efficiency
- Good at finding global maxima
- Good at determing shape of posterior
- Works for not-nice functions
- Does not require fine tuning of settings by user

Disadvantages:

- Falls into the trap of dimensionality
- Significant extra computation

Bayesian Optimisation vs MCMC

- This computational overhead means there is a threshold in terms of complexity and cost of posterior evaluations

- Above this threshold, Bayesian Optimisation will peform better than MCMC (Limit: cost $\to \infty$)

Model Selection for Parameter Inference

- To do parameter inference, we must choose a model
- How do we choose this model?
- How can we compare different models?
- Bayesian Optimisation may also provide a solution to this!

Model Selection with Bayesian Evidence

- Bayes Theorem:

$$\mathcal{P}(\mathcal{M}|\mathcal{D}) = \frac{\mathcal{L}(\mathcal{D}|\mathcal{M})\pi(\theta|\mathcal{M})}{P(\mathcal{D})}$$

- Probability of $\mathcal{M}$ given $\mathcal{D}$:

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D})}$$

- Bayesian Evidence (Bayes Factor):

$$B = P(\mathcal{D}|\mathcal{M}) = \int d\theta \mathcal{L}(\mathcal{D}|\theta, \mathcal{M})\pi(\theta|\mathcal{M})$$

Model Selection with Bayesian Evidence

- To compare two different models $\mathcal{M}_1$ and $\mathcal{M}_2$:

$$B_{12} = \frac{P(\mathcal{D}|\mathcal{M}_1)}{P(\mathcal{D}|\mathcal{M}_2)}$$

- $\mathcal{M}_1$ is B_{12} times more probable than $\mathcal{M}_2$

Model Selection with Bayesian Evidence

- Bayesian Evidence:

$$P(\mathcal{D}|\mathcal{M}) = \int d\theta \mathcal{L}(\mathcal{D}|\theta, \mathcal{M})\pi(\theta|\mathcal{M})$$

- This is the integral over our posterior prediction $\mathcal{P}(\mathcal{M}|\mathcal{D})$

- Rewards models with accurate ‘risky‘ predictions over generic ones - Occam’s Razor

What do we actually want to optimise?

- The aim of this optimisation is to find the Bayesian Evidence to some level of precision.
- The evidence is a relative measure of how well a model fits the data.
- That is to say, we can use the evidence to compare how well two models fit the data.

Bayesian Optimisation for Bayesian Evidence

- Integration over mutli-dimensional posteriors is still hard
- Typical methods require $\mathcal{O}(10^5 - 10^6)$ evaluations for ΛCDM with primordial features
  - Nested Sampling [Skilling 2004, Feroz et al. 2013, Handley et al. 2015]
- We can use the same methodology as parameter inference - replace the actual posterior with the GPR interpolation

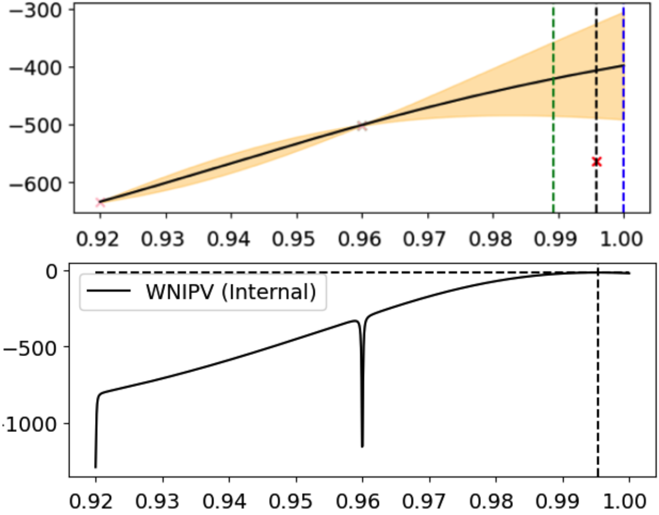
Optimising for the Evidence

- We want our acquisition function 'metric' to be reducing the uncertainty in the integral over parameter space
- We use the Weighted Negative Integrated Posterior Variance

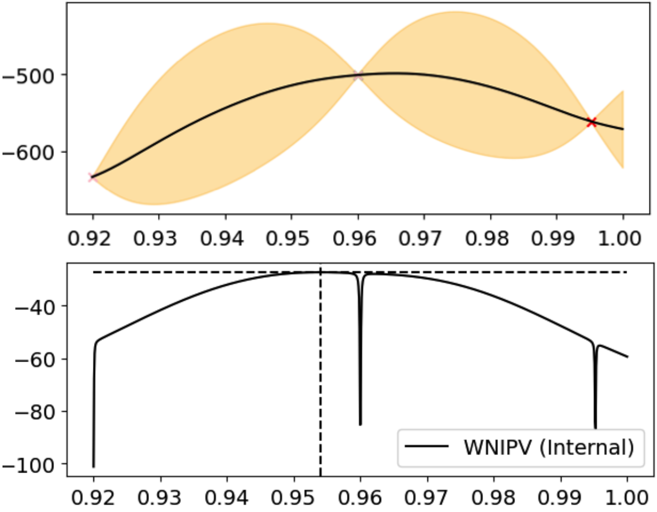
$$WNIPV(\theta) = \int d\theta' \sigma \widehat{GP(\theta)}(\theta')$$

- $\widehat{GP(\theta)}$ - posterior if we pretend to take a sample at a new point θ
- σ - GP Uncertainty

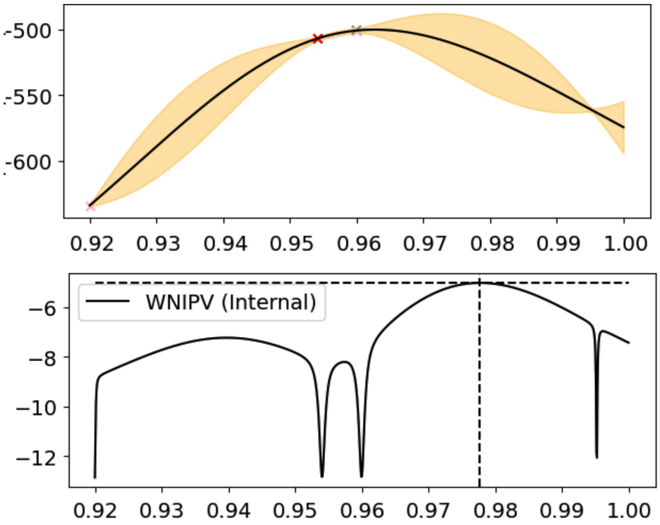
Bayesian Optimisation (iteration 1)



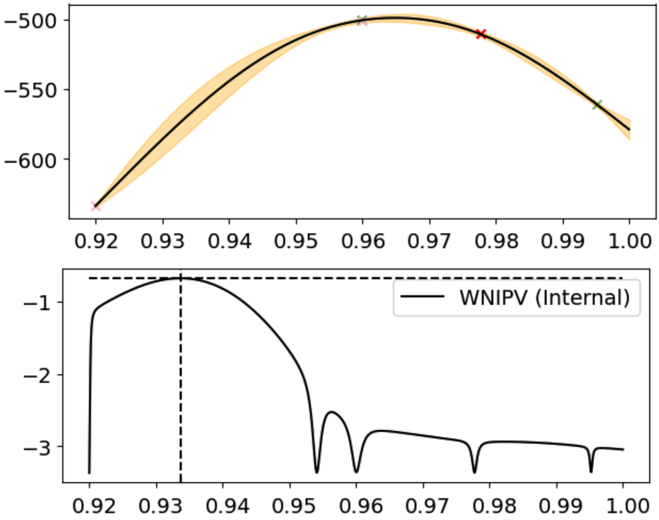
Bayesian Optimisation (iteration 2)



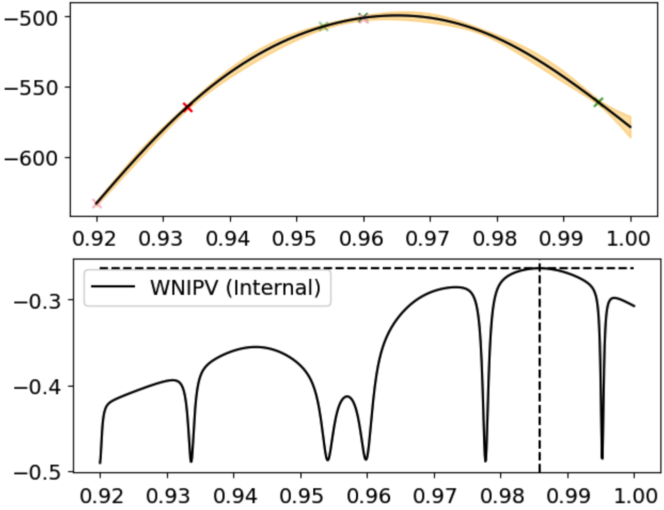
Bayesian Optimisation (iteration 3)



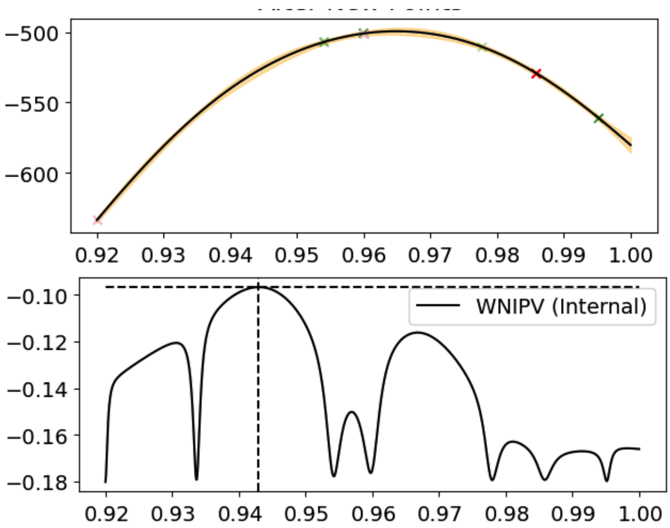
Bayesian Optimisation (iteration 4)



Bayesian Optimisation (iteration 5)



Bayesian Optimisation (iteration 6)



Bayesian Optimistian - Precision Threshold

- Through each iteration the overal uncertainty of the GP goes down
- The maximum value of the acquisition function also reduces
- This allows us to define a threshold or termination criteria on the precision of the evidence

The Finishing Touches

- We use some smart priors to help deal with higher dimensional parameter spaces (Sparse Axis-Aligned Subspaces) [Eriksson & Jankowiak (2021)]
  - All dimensions are innocent until proven guilty
- We use Nested Sampling to get both a direct numerical estimate of the uncertainty on the evidence as well as uncorrelated samples of the posterior
- We sample from hyperparameter space PDF with NUTS (modified HMC) instead of optimising over it

Preliminary Results - Banana

Preliminary Results - Annular Function

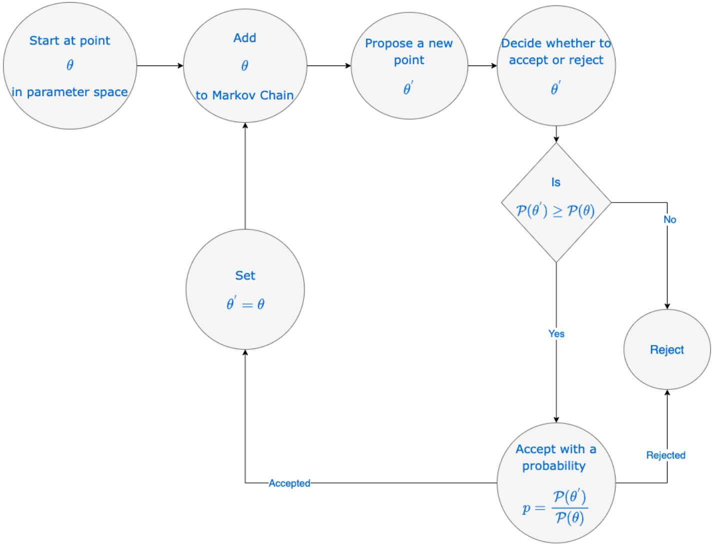
Summary

- Efficient ML algorithm for model selection and parameter inference
  - Preliminary improvement of 100x fewer samples
- Best for difficult to obtain, expensive to calculate and complicated likelihoods
- Benchmark is to take fewer samples than other methods with the same precision on evidence
- Paper and code coming soon!

Extra Slides

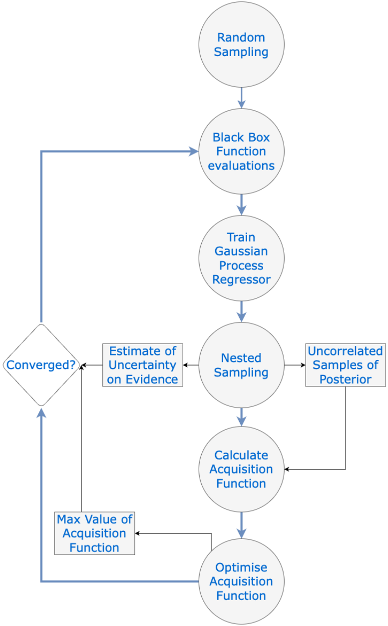
Extra! Extra! Read all about it!

Metropolis-Hastings MCMC

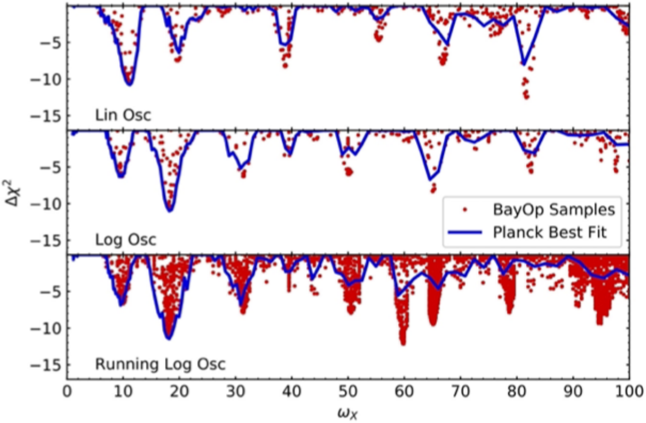


[Metropolis et al. (1953)]

BOBE Algorithm

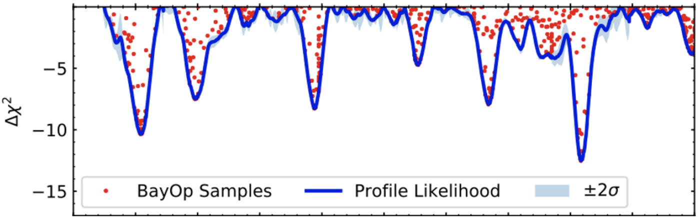


Example Application: ΛCDM with modulated Primordial Fluctuations [Jan Hamann & Julius Wons, 2021]



- Using Nested Sampling : $\mathcal{O}(10^5)$ samples
- Our results with BO: 2 orders of magnitude improvement in # evaluations

Example Application: ΛCDM with modulated Primordial Fluctuations [Jan Hamann & Julius Wons, 2021]



- Also learns the global shape of the function!
- 1700 samples, 8 frequency bins