



Science & Technology
Facilities Council

UK Research
and Innovation

PHYSTAT-nu 2019

A biased summary

Lukas Koch

STFC Rutherford Appleton Laboratory

PHYSTAT-nu @ CERN 2019



“The workshops deals with the statistical issues in the broad range of modern neutrino physics, in particular in view of the increased precision of current and future experiments. Topics to be included are

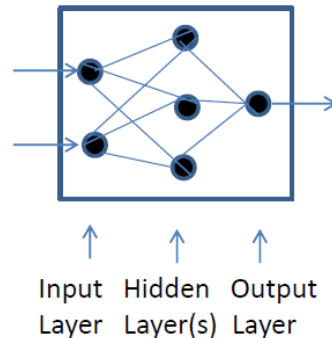
- Measuring model parameters, choosing between models
- Setting limits, discovery
- Determination of systematic uncertainties
- Unfolding
- Machine learning for event reconstruction and classification

Bias ahead!

- Personal selection of “interesting” presentations, discussions, comments
 - Also just stuff that I have learned
- Try to reflect “general opinion” (where even applicable), but no guarantee
- If you have any further questions, ask ~~your doctor or pharmacist or nurse~~ a proper statistician.

New insights in neural networks

- “Introduction to Statistical Issues for Phystat-nu” by Louis Lyons



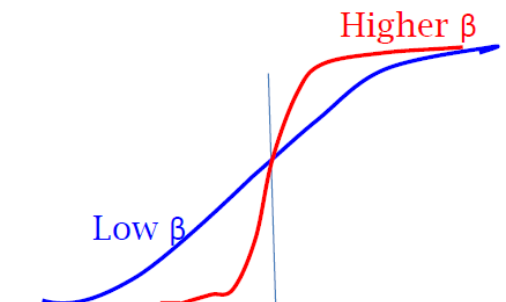
For each hidden or output node j
$$\text{Output}_j = F \left[\sum \text{Input}_i * W_{ij} + T_j \right]$$

(W and T = network params)

Typical $F(x) = 1/(1 + e^{-\beta x})$ Sigmoid

For large β , output of node j is ‘ON’ if
$$\sum I_i w_{ij} + T_j > 0$$

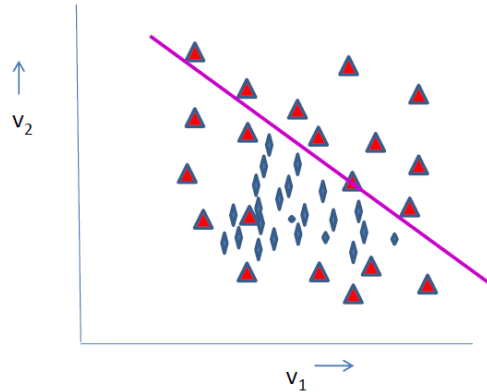
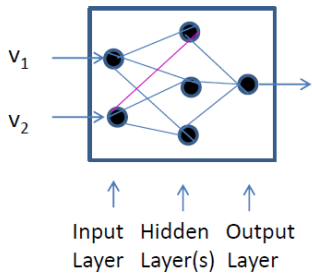
This is ‘hyper-plane’ in I space ●



40

https://indico.cern.ch/event/735431/contributions/3175658/attachments/1777100/2889616/Introductory_Statistics_for_PHYSTAT-nu2019.pdf

HOW DOES IT WORK?



For First hidden node

Straight line is

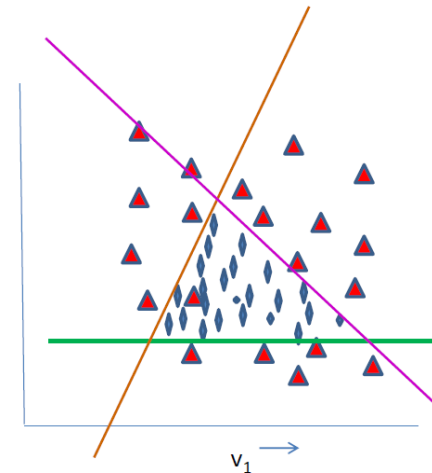
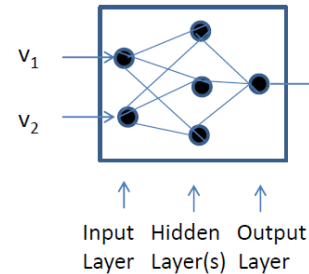
$$w_{11} * v_1 + w_{21} * v_2 + T_{10} = 0$$

where

w_{ij} is weight from i^{th} input node to j^{th} hidden node

T_{k0} is threshold for k^{th} hidden node

HOW DOES IT WORK?



Output = Sigmoid{ $0.4H_1 + 0.4H_2 + 0.4H_3 - 1.0$ }
 Output is 'On' only if $H_1 H_2 H_3$ all are 'On'

N.B.

- * Complexity of final region depends on number of hidden nodes.
- * Finite $\beta \rightarrow$ rounded edges for selected region; and contours of constant output in (v_1, v_2) plane.

41

44

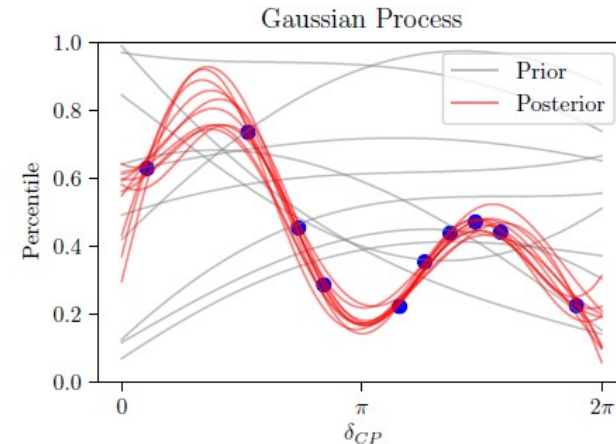
Faster FC

- “Efficient Neutrino Oscillation Parameter Inference with Gaussian Process” by Nitish Nayak

- ▶ Enormously flexible! Can basically approximate any well behaved function with an appropriate choice of the kernel.
- ▶ Predictions at new data points are computationally tractable with basic linear algebra, i.e for $\mathcal{GP}(\mathbf{0}, k(\cdot, \cdot))$:

$$f(x')|f(x) \sim \mathcal{N}\left(\frac{k(x, x')}{k(x, x)}f(x), k(x', x') - \frac{k(x, x')^2}{k(x, x)}\right)$$

- ▶ Kernel hyperparameters can be learned via maximising the likelihood of current set of observations marginalised over the function distribution, f

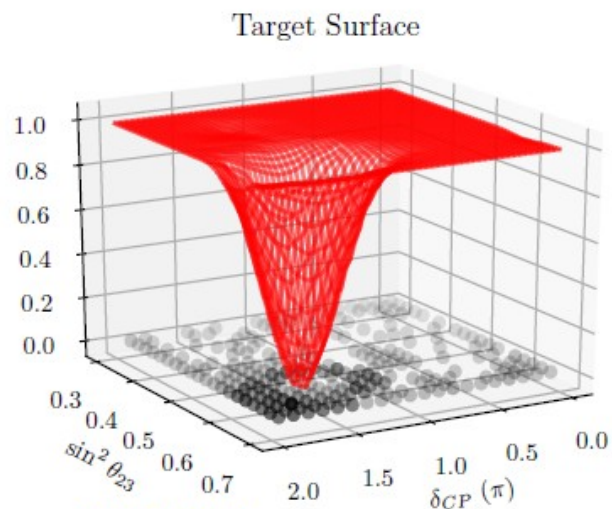


Optimised Confidence Interval Search

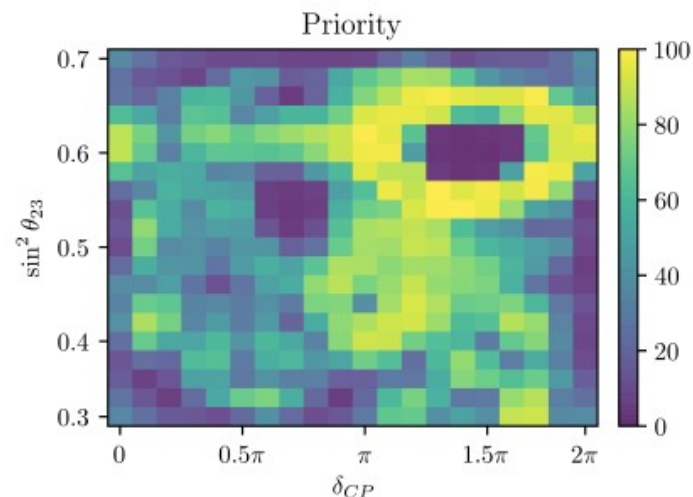
- ▶ Use an acquisition function that proposes new points in θ -space to explore based on \mathcal{GP} approximated percentile surface.

$$a(\theta) = \sum_{\alpha_i} \left| \frac{\hat{q}(\theta) - \alpha_i}{\sigma_{\hat{q}(\theta)}} \right|^{-1}$$

- ▶ Here, $\hat{q}(\theta)$ is \mathcal{GP} mean, $\sigma_{\hat{q}(\theta)}$ is \mathcal{GP} std-dev, α_i is chosen to be (0.68, 0.90)
- ▶ $a(\theta)$ balances between exploration, i.e MC experiments at new points and exploitation, i.e reducing \mathcal{GP} error



PhyStatNu - 2019

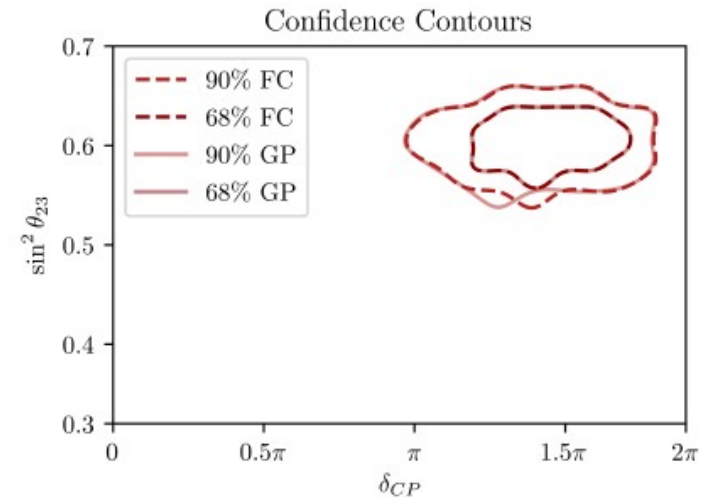
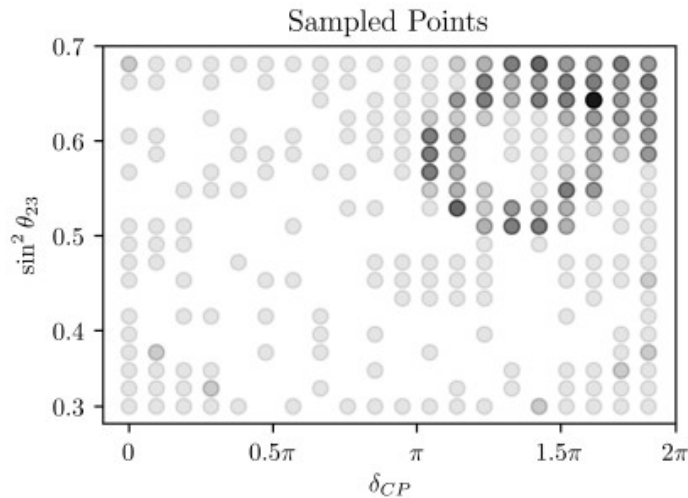


16 / 32

Lingge Li, Nitish Nayak, Jianming Bian, Pierre Baldi

Results

- ▶ "Real" data similar to latest best-fit estimate from NOvA. ($\sin^2\theta_{23} = 0.56$, $\Delta m_{32}^2 = 2.44 \times 10^{-3} \text{ eV}^2$, $\delta_{CP} = 1.5\pi$)
- ▶ $\sin^2\theta_{23} - \delta_{CP}$ 68% and 90% CI for IH after 5 iterations

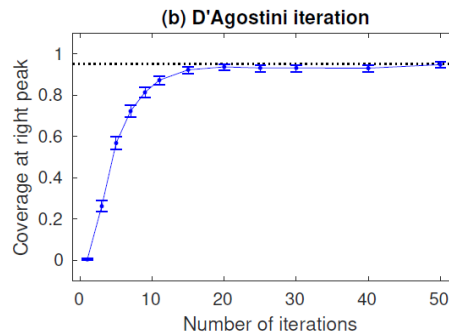
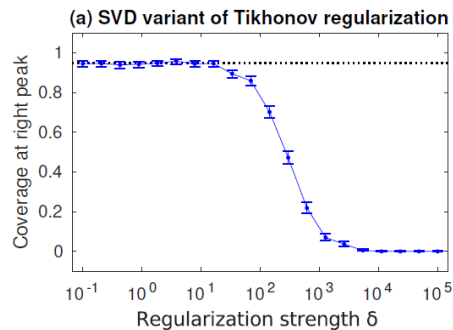


- ▶ Grayscale denotes number of experiments thrown in relation to FC (2000)
- ▶ Algorithm does a good job of finding the FC contour edge!

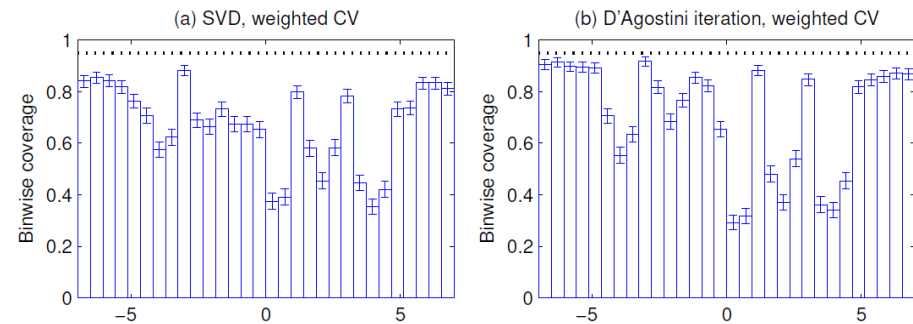
Better regularisation of unfolded results

- “Introduction to Unfolding: A Statistician’s Perspective” by Mikael Kuusela

Coverage as a function of regularization strength



Undercoverage of existing methods



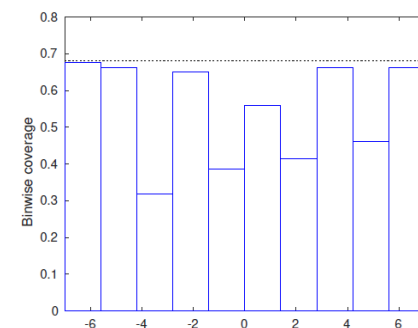
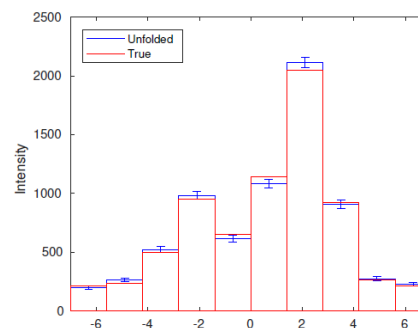
There is major undercoverage if regularization strength chosen using (weighted) cross-validation; same is true for L-curve and MMLE

- The simplest form of debiasing is to reduce δ from the cross-validation/L-curve/MMLE value until the intervals have close-to-nominal coverage
- The challenge is to come up with a data-driven rule for deciding *how much to undersmooth*
- I have been working with a student Lyle Kim to implement the data-driven methods from Kuusela (2016) as an extension of TUnfold
- The code is available at:

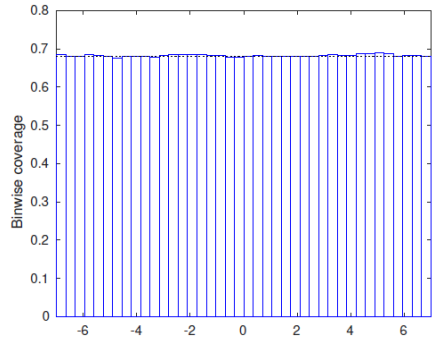
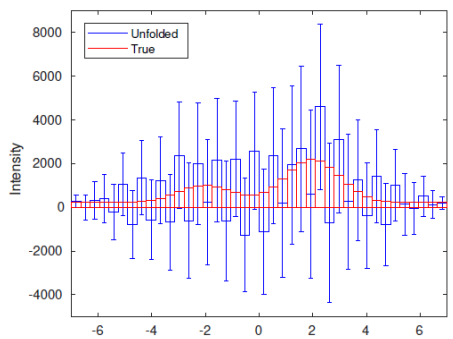
<https://github.com/lylejkim/UndersmoothedUnfolding>

- If you're already working with TUnfold, then trying this approach requires adding only one extra line of code to your analysis
 - See the Github repository for example usage

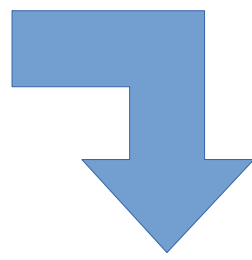
Wide bins, standard approach, perturbed MC



The response matrix $K_{i,j} = \frac{\int_{S_i} \int_{T_j} k(s,t) f^{MC}(t) dt ds}{\int_{T_j} f^{MC}(t) dt}$ depends on f^{MC}
 \Rightarrow Undercoverage if $f^{MC} \neq f$



Re-bin *after* unfolding

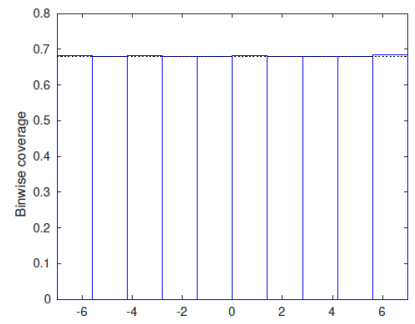
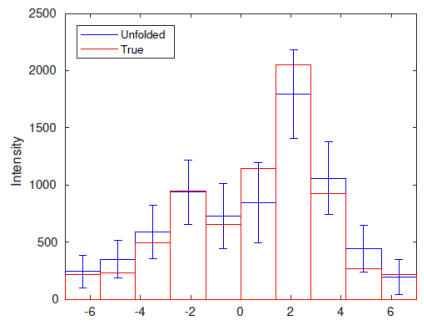


With narrow bins, less dependence on f^{MC} so coverage is correct, but the intervals are very wide¹

⇒ Let's aggregate these into wide bins, keeping track of the correlations

¹More unfolded realizations given in the [backup](#).

Wide bins via fine bins, perturbed MC



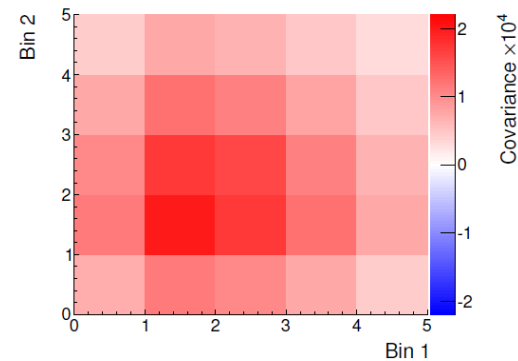
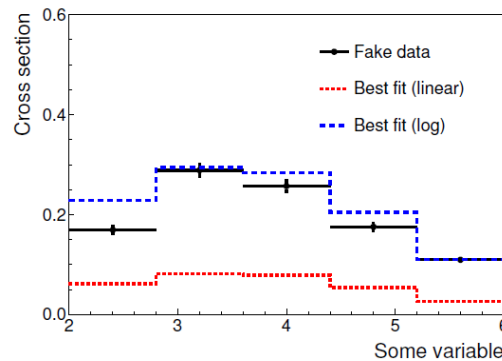
Wide bins via fine bins gives both correct coverage and intervals with reasonable length²

²More unfolded realizations given in the [backup](#).

Beware the correlations

- “Unfolding: the neutrino experiment experience” by Philip Rodrigues

Fits to strongly-correlated data 2



$$\chi^2 = (\mathbf{D} - \mathbf{M})^T \mathbf{C}^{-1} (\mathbf{D} - \mathbf{M})$$

$$C_{ij} = \sum_{\text{universe } k} (y_i^{(k)} - y_i^*) (y_j^{(k)} - y_j^*)$$

- ▶ “Multi-universe”: throw random systematic universes, re-extract result
- ▶ Empirically, $y \rightarrow \log(y)$, ameliorates the issue, \Rightarrow log-normal uncertainties on y (?)

“Box-Cox transformation for resolving the Peelle’s Pertinent Puzzle in curve fitting”, Oh and Seo 2004

- ▶ Is this the best way to communicate our systematics?

●●●●

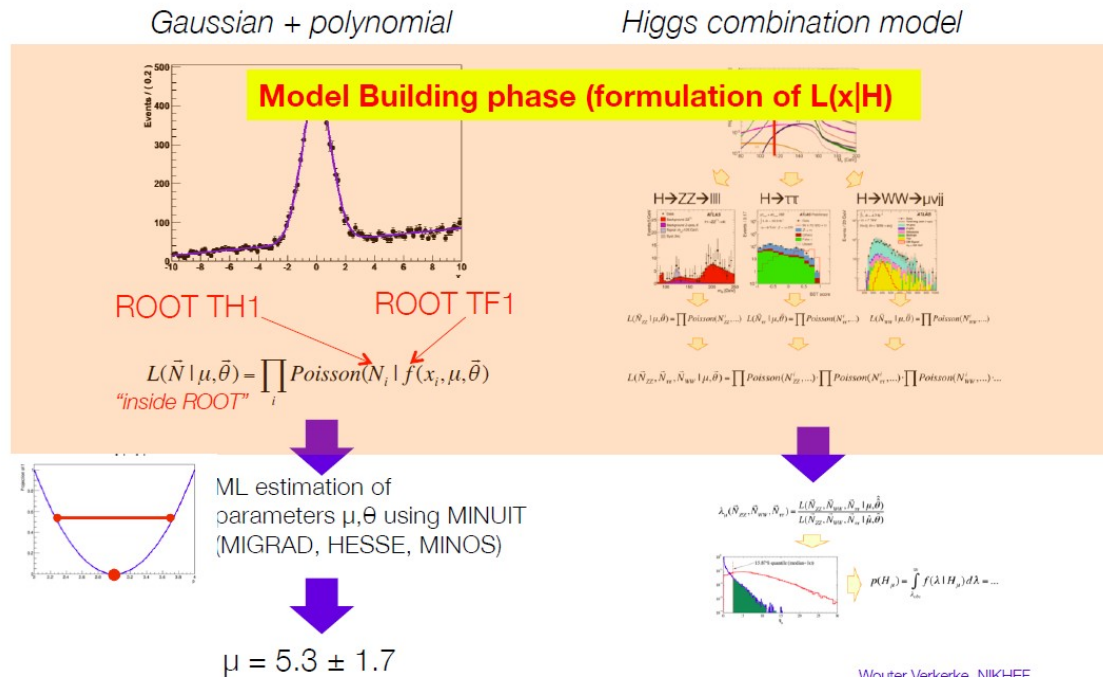
18

<https://indico.cern.ch/event/735431/contributions/3137826/attachments/1781943/2904065/physta-t-nu-2019-neutrino-unfolding.pdf>

Sharing data the LHC way

- “Statistical model building at the LHC” by Wouter Verkerke

How is Higgs discovery different from a simple fit?



https://indico.cern.ch/event/735431/contributions/3137802/attachments/1783665/2902994/phystatnu_modeling.pdf

How is Higgs discovery different from a simple fit?

Gaussian + polynomial

Higgs combination model

Design goal:

Separate **building of Likelihood model** as much as possible from statistical analysis **using the Likelihood model**

- More modular software design
- 'Plug-and-play with statistical techniques
- Factorizes work in collaborative effort

ML estimation of parameters μ, θ using MINUIT (MIGRAD, HESSE, MINOS)

$$\mu = 5.3 \pm 1.7$$

$$\lambda_i(\theta_1, \theta_2, \theta_3) = \frac{L(\theta_1, \theta_2, \theta_3, \mu, \theta)}{L(\theta_1, \theta_2, \theta_3, 1, \theta)}$$

$$p(H_1) = \int_{\lambda_{min}}^{\lambda_{max}} f(\lambda | H_1) d\lambda = \dots$$

Wouter Verkerke, NIKHEF

The idea behind the design of RooFit/RooStats/HistFactory

- Modularity, Generality and flexibility
- Step 1 – Construct the likelihood function $L(x|p)$

RooFit, or RooFit+HistFactory

- Step 2 – Statistical tests on parameter of interest p

Procedure can be Bayesian, Frequentist, or Hybrid, but always based on $L(x|p)$

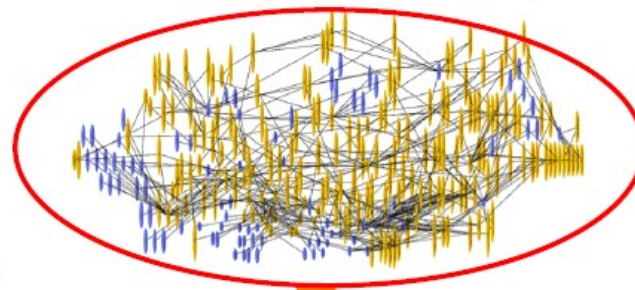
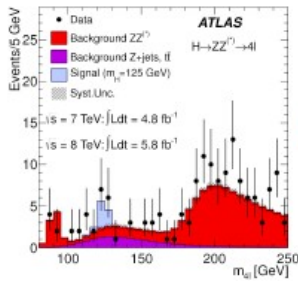
RooStats

- Steps 1 and 2 are conceptually separated, and in Roo* suit also implemented separately.

Wouter Verkerke, NIKHEF

Analysis chain identical for highly complex (Higgs) models

- **Step 1** – Construct the likelihood function $L(x|p)$



Rooworkspace

*Complete description
of likelihood model,
persistable in ROOT file
(RooFit pdf function)
Allows full introspection
and a-posteriori editing*

- **Step 2** – Statistical tests on parameter of interest p

```
Rooworkspace* w=TFile::Open("L.root")->Get("w") ;  
RooAbsPdf* model = w->pdf("model") ;  
pdf->fitTo(data,  
           GlobalObservables(w->set("MC_G1Obs"),  
           Constrain(*w->st("MC_NuisParams")) ;
```

er Verkerke, NIKHEF

Frequentist v Bayesian

- “PhyStat-v 2019, A Statistical Summary” by David A. van Dyk

Frequentist or Bayesian?

Do you have to choose??

- Bayes prescribes method — Frequency evaluates method.
- Frequency evaluation of Bayesian methods.
*I like intervals to include true values... at least once in a while.
Rob, Glen over lunch: marginalize/profile nuisance parameters*
- Model fitting: often little difference in fits and errors.
... but not always, see below.
- Why not control detection error
and assess probability of new physics?
- Why throw away half of your tool box?

I'm impressed with the openness of neutrino researchers to both Bayesian and Frequency based methods.

- Several Bayesian procedures (Alex H, Stefano G, Matteo A, Glen C) Imperial College London
- My experience with cosmologists & particle physicists.

David A. van Dyk

PhyStat-v 2019

<https://indico.cern.ch/event/735431/contribution/s/3137832/attachments/1785472/2906708/phystatnu-summary.pdf>

Maybe avoid unfolding

Unfolding and Deconvolution

Nice overview by Mikael K, Stefan S, Phillip R, Stephen D, Xin Q

- X is a smeared / blurred version of “ideal” data Y .
- Suppose $Y \sim f$ so that $X \sim f \circ K$
- Unfolding ignores f and models Y as Multinomial(p), fits p by ML or Penalized ML.
... stopping EM early is an outdated strategy.

Better Strategy ... *but what if there are complex errors in K ?*

- Compare $f \circ K$ directly with X .
- No need for regularization (f will provide it automatically).
- Or compare $f_1 \circ K$ with $f_2 \circ K$.
If smeared data can't distinguish models... like mass hierarchy!
- Avoid background subtraction. (STEPHEN D CAN DO!)
- LIRA: Bayesian, estimate regularization on the fly!
Weak structure in f – Esch, DvD et al (2004), Astrophysical J, 610, 1213

Imperial College
London

Beware the priors

- “Fits to large and combined data sets” by Stefano Gariazzo

■ Can current data tell us the neutrino mass ordering?

- 1 [Hannestad, Schwetz, 2016]: **extremely weak** (2:1, 3:2) preference for NO (cosmology + [Bergstrom et al., 2015] neutrino oscillation fit) Bayesian approach;
- 2 [Gerbino et al, 2016]: **extremely weak** (up to 3:2) preference for NO (cosmology only), Bayesian approach;
- 3 [Simpson et al., 2017]: **strong** preference for NO (cosmological limits on $\sum m_\nu$ + constraints on Δm_{21}^2 and $|\Delta m_{31}^2|$) Bayesian approach;
- 4 [Schwetz et al., 2017], “Comment on ...” [Simpson et al., 2017]: effect of prior?
- 5 [Capozzi et al., 2017]: **2 σ** preference for NO (cosmology + [Capozzi et al., 2016, updated 2017] neutrino oscillation fit) frequentist approach;
- 6 [Caldwell et al., 2017] **very mild** indication for NO (cosmology + neutrinoless double-beta decay + [Esteban et al., 2016] readapted oscillation results) Bayesian approach;
- 7 [Wang, Xia, 2017]: Bayes factor NO vs IO is **not informative** (cosmology only).

[Simpson et al, 2017]

[Caldwell et al, 2017]

using m_1, m_2, m_3 (A)

using $m_{\text{lightest}}, \Delta m_{21}^2, |\Delta m_{31}^2|$ (B)

intuition says: (B) is closer to observable quantities! Better than (A)?

Should we use linear or logarithmic priors on m_k (m_{lightest})?

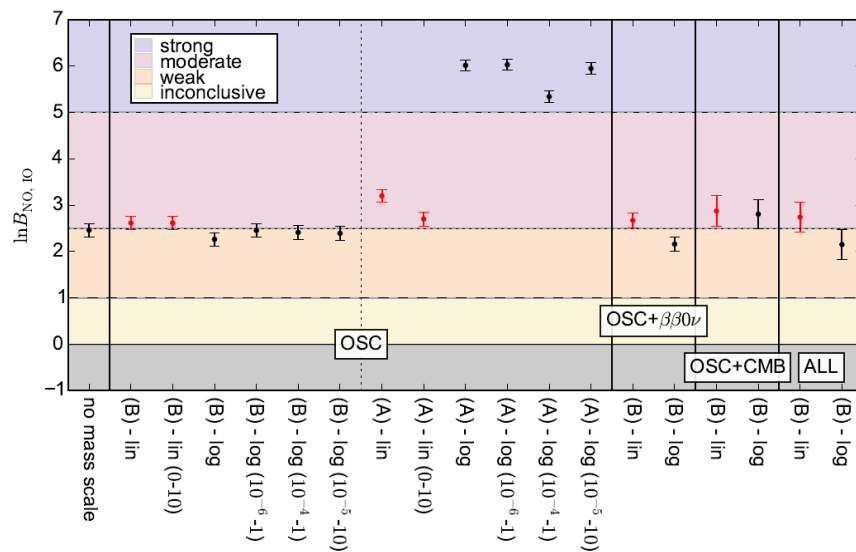
Can data help to select (A) or (B), linear or log?

| Case A | | | Case B | | |
|-----------|---------------|------------------------|--------------------------|---------------|---|
| Parameter | Prior | Range | Parameter | Prior | Range |
| m_1/eV | linear log | 0 - 1 $10^{-5} - 1$ | m_{lightest}/eV | linear log | 0 - 1 $10^{-5} - 1$ |
| m_2/eV | linear log | 0 - 1 $10^{-5} - 1$ | $\Delta m_{21}^2/eV^2$ | linear | $5 \times 10^{-5} - 10^{-4}$ |
| m_3/eV | linear log | 0 - 1 $10^{-5} - 1$ | $ \Delta m_{31}^2 /eV^2$ | linear | $1.5 \times 10^{-3} - 3.5 \times 10^{-3}$ |

S. Gariazzo "Fits to large and combined data sets" PHYSTAT-nu, 23/01/2019

Comparing the mass orderings

[SG+, JCAP 03 (2018) 11]



Note: only oscillation data until the end of 2017 are included!

S. Gariazzo "Fits to large and combined data sets" PHYSTAT-nu, 23/01/2019 19/25

Disclose everything!

- “Statistical Issues & Methods in Neutrinoless Double-beta Decay Experiments” by Matteo Agostini

Statistics & Double- β Decay Community

[Phys. Rev. Lett. 120, 132502 (2018)]

- most experiments quote results from multiple methods and give enough info to reproduce the analysis
- sensitivity always reported (sometimes also for Bayesian methods)
- blind analysis is almost the standard
- frequentist intervals still used as Bayesian intervals (even when Bayesian interval is available)
- sensitivity computed for the no signal hypothesis, more interesting to quote discovery power

| Statistical Method in the last PRL of the MAJORANA DEMONSTRATOR | $T_{1/2}$ lower limit 90% prob [10^{25} yr] | $T_{1/2}$ lower limit sensitivity [10^{25} yr] |
|---|--|---|
| Counting (FC) | 1.6 | |
| Unbinned likelihood fit (FC) | 1.9 | 2.1 |
| Unbinned likelihood fit & CLs | 1.5 | 1.4 |
| Bayesian flat prior | 1.6 | |
| Bayesian Jeffreys prior | 2.6 | |

M. Agostini (TU Munich)

29

<https://indico.cern.ch/event/735431/contributions/3137822/attachments/1784175/2904069/agostini-0vbb-phystat.pdf>

More cooperation

- Big LHC experiments have dedicated “statistics committees”
 - Quite useful in making sure “correct” methods are used
- Many neutrino experiments are too small for own committee
- Possible solution:
Inter-collaboration statistics committee?

Folding forward

- “ReMU – Response Matrix Utilities” poster by Lukas Koch (me)

ReMU - Response Matrix Utilities
A Python framework for likelihood calculations and hypothesis testing using binned events and response matrices.

Lukas Koch, STFC Rutherford Appleton Laboratory, lukas.koch@stfc.ac.uk

Forward Folding
Conducting an experiment:
• Sample from true distribution
• Detector adds efficiency and smearing effects

To compare with models:
• Sample from model distribution
• Simulate detector effects
• High statistics limit yields expectation values to be compared with data

Only feasible for limited number of models because:
• Pull detector simulation takes lots of computing time
• Requires expert knowledge about detector
• Detector simulation software often only available within a collaboration

Find optimal linear relation between true (μ_j) and reconstructed (ν_i) distributions:
$$\nu_i = \sum_j P(\text{reco bin } i | \text{truth bin } j) \mu_j$$

Replace full simulation with response matrix R_{ij}
$$R_{ij} = P(\text{reco bin } i | \text{truth bin } j)$$

Matrix multiplication is computationally simple task that does not require expert knowledge and allows fast tests of many theoretical models:

Linear relationship between truth and reco must be universal, i.e. independent of tested models, so truth binning will be finer (and in more variables) than reco binning.

Under-constrained truth not an issue when only folding forward, from truth to reco.

Implementation in ReMU
ReMU implements classes for n-dimensional binning:
`binning.Binning(n_dims, n_bins, n_centers, n_widths, n_weights)`

Truth and reco binning are combined into a response matrix object, which is filled with simulated data:
`binning.ResponseMatrix(n_dims, n_bins, n_centers, n_widths, n_weights)`

“LikelihoodMatrix” class implements likelihood calculations for arbitrary models, i.e. truth expectations:
$$L(\mu) = \prod_{i=1}^N \frac{e^{-R_{ij}\mu_j}}{n_i!} \prod_{j=1}^M \frac{(R_{ij}\mu_j)^{n_i}}{n_i!} e^{-R_{ij}\mu_j}$$

Uncertain detector properties are included as prior $P(\mu_j)$
$$L(\mu) = \int P(\mu_j) \prod_{i=1}^N \frac{e^{-R_{ij}\mu_j}}{n_i!} \prod_{j=1}^M \frac{(R_{ij}\mu_j)^{n_i}}{n_i!} e^{-R_{ij}\mu_j}$$

ReMU replaces integral with sum over sample from $P(\mu_j)$
$$L(\mu) = \sum_{k=1}^N \prod_{i=1}^N \frac{e^{-R_{ij}\mu_j^k}}{n_i!} \prod_{j=1}^M \frac{(R_{ij}\mu_j^k)^{n_i}}{n_i!} e^{-R_{ij}\mu_j^k}$$

Convenience functions likelihood maximisation:
`binning.LikelihoodMatrix.maximize()`
and for numerically calculating different p-values:
`binning.LikelihoodMatrix.pvalue()`

Allows very simple construction of confidence intervals:
`binning.LikelihoodMatrix.confidence_intervals()`

Support for PyMC
PyMC is a Markov Chain Monte Carlo (MCMC) sampling toolkit:
<http://pymc-devs.github.io/pymc/>

ReMU can create PyMC MCMC objects, allowing very easy sampling from the posterior distribution:
`binning.LikelihoodMatrix.pyMC()`

This is useful especially for models with many free (nuisance) parameters, as the MCMC sampling scales much better than the numerical frequentist methods.

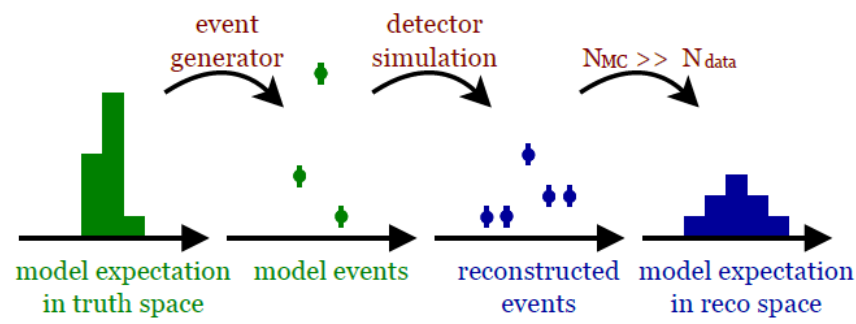
With the fit trees available, it is also easy possible to investigate correlations of parameters or the distribution of functions of parameters:
`binning.LikelihoodMatrix.correlations()`
`binning.LikelihoodMatrix.function_distributions()`

A Note on Backgrounds
Subtracting background events from the data vector would break the Poisson statistics, so backgrounds must be added to the expectation values:

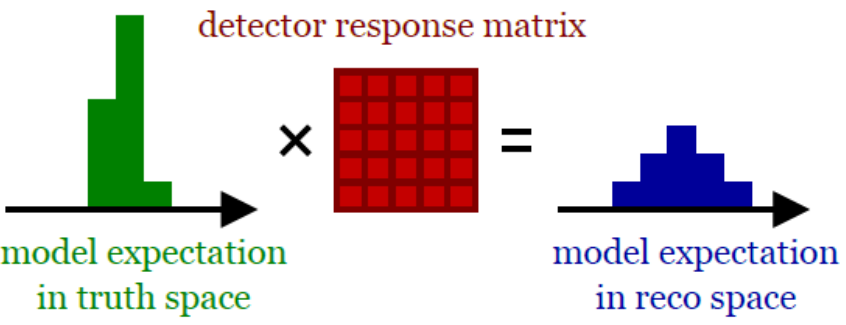
If the data is not able to constrain the backgrounds, e.g. via control regions, detector experts can provide templates to be used as priors in distributions fits.

If the background is very detector specific, its shape can also be provided as a column in the response matrix:

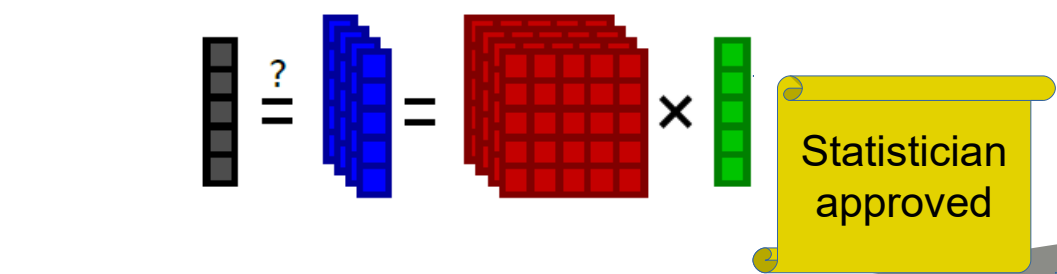
https://indico.cern.ch/event/735431/contributions/3273211/attachments/1784381/2904492/ReMU_Posters.pdf



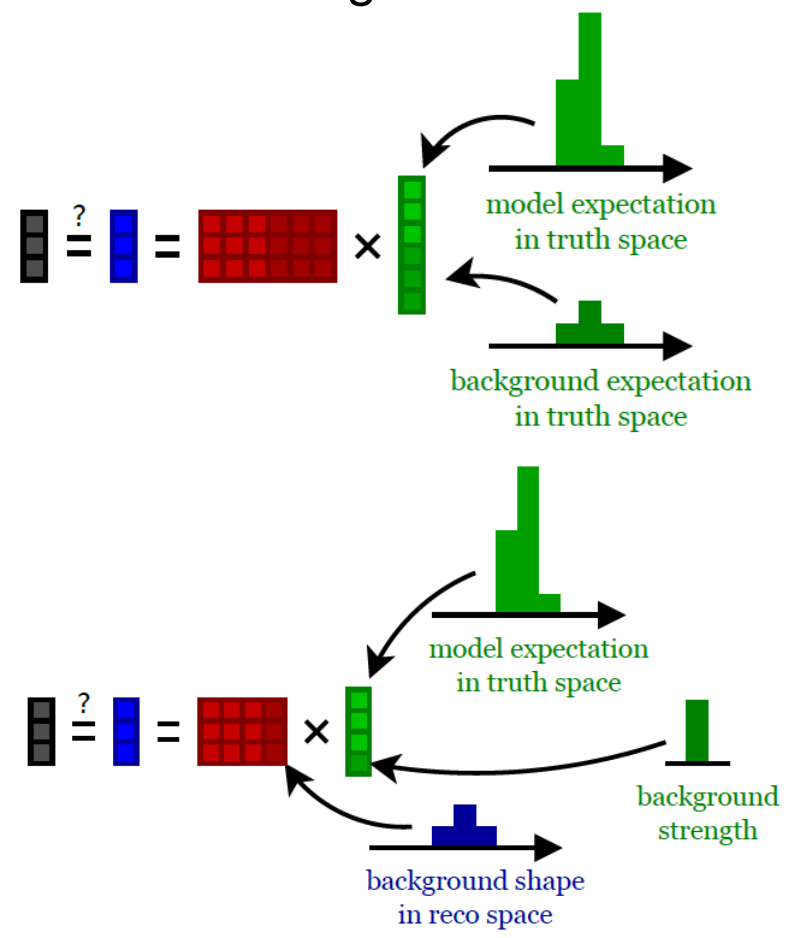
Replace full detector MC with matrix multiplication



Include detector systematics as set of matrices



Multiple possible ways to handle backgrounds



Thank you!





Science & Technology
Facilities Council

UK Research and Innovation