Developments in MC tuning methods: Professor 2 and all that

Andy Buckley
University of Glasgow

MPI@LHC 2015, ICTP, Trieste, Nov 2015

Introduction

MC tuning is a necessary evil – data description is needed, and models aren't as predictive as we'd like. Data and models also aren't as perfect as we'd like: we need estimates of systematics. *Hope that somewhere along the way we also gain better physical understanding…*

Professor is numerical machinery frequently used these days to aid MC generator tuning. Since 2009. Many widespread tunes used it…though not all!

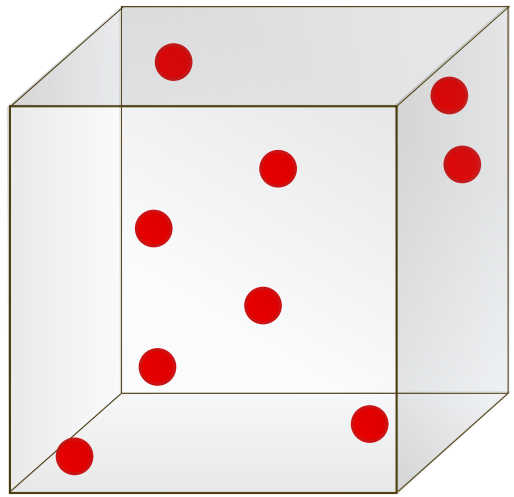
Basic idea is to build fast parameterisations of observable response to parameters, from parallel sampling of the param space. *Then optimise.*

Systematics should be derivable from the shape of the goodness-of-fit around the optimum. *Right?*

In this talk: Professor v2, handling systematics, other applications

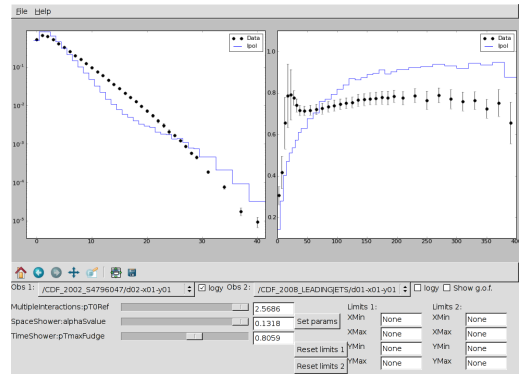
The Professor method

- **MC is slow: $\sim$ 1 day per run** $\Rightarrow$ can't use in serial optimisation.
- Solution is very simple: trivially parallelise MC runs throughout "reasonable" range of parameter space, and use sampled "anchor" points to interpolate each bin's param dependence. Up to $\mathcal{O}(15)$ params.
- We use SVD polynomial fits for robustness – requires that values vary in a polynomial fashion, *or are transformed to do so.*
- Analytic interpolations $\Rightarrow$ fast serial minimisation of an objective function.
- Much strength is in the "system" features to support the core machinery.



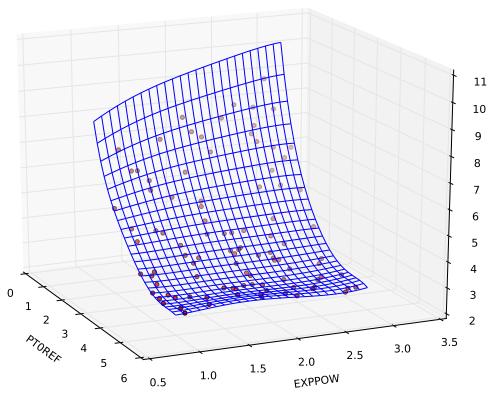
Professor 1

- Python code, using scipy and pyminuit
- Heavy internal code framework, hangover from early development work
- Parameterisation up to 5th order – manually encoded
- Many "magic" behaviours, coded in > 20 scripts. . . not all well maintained!
- Lots of "advanced" features, like uncertainty correlations, sensitivities, interactive GUI, etc.
- **Hit "maximum entropy" point of development!**



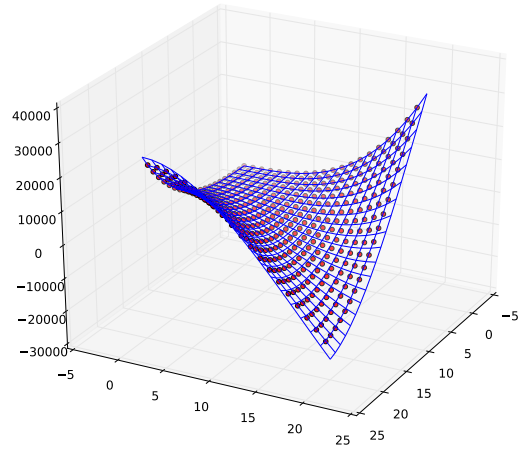
Professor 2

- Start again from scratch, to make Professor lean, mean & flexible again. **Thanks to Holger Schulz**
- Part motivated by more generic applications, part by pure frustration!
- Core code now in C++; independent of concepts like “bins”. Very generic
- Wrapped into Python, and used as core of a library for data I/O ⇒ lessons learned from v1
- Works best integrated with the YODA data toolkit, but not tied to Rivet
- Now only a few scripts – and most work is done by the library, so scripts easily customised



More on Professor 2

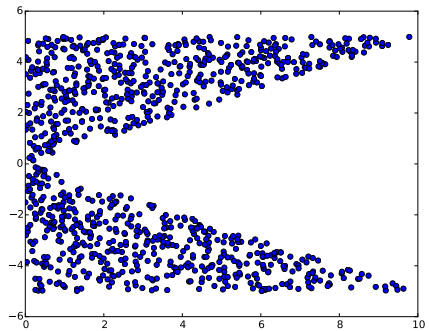
- All-orders parameterisations now possible, including. zero (i.e. constant)
- SVD stability improvements through param mapping
- Not all old features: we still "need" envelopes, sensitivities, correlations, eigentunes, runcombns, kebabs, ...
- More general weighting system: matching patterns, bin ranges, etc.
- Strengths are speed and programmability – less emphasis on pre-built magic, more on power & flexibility
- Interactive Web interface in development



Param sampling

Prof 2's more powerful sampling script **`prof2-sample`** allows biased sampling using SymPy expressions.

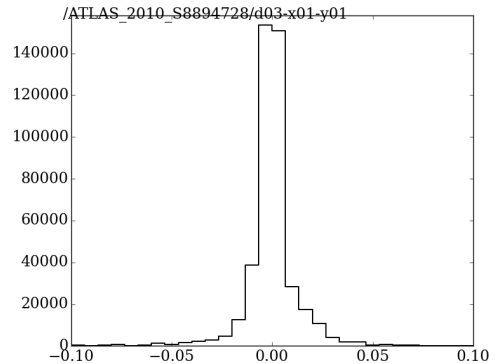
- Programmable vetos also supported, e.g. to ban regions where param A greater than param B. Used in Sherpa 2.2 tuning.
- Sampling can now be used to generate run scripts and any other file as well as the standard params list $\Rightarrow$ easy to use "meta-params" to control multiple parameters.
- Still very up to the user to ensure that sampling is dense enough and concentrated appropriately. *To calculate where best to sample, you'd need to already know the answer!*

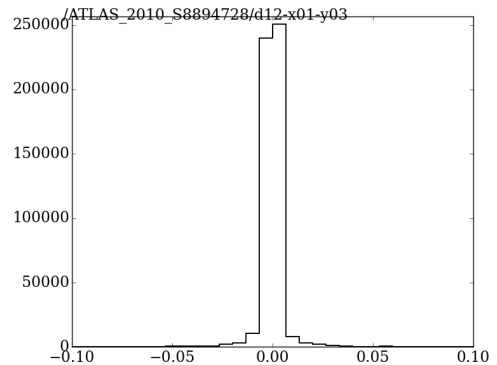


Testing the parameterisation

Can't just assume that parameterisation is working. . . but this is often done / inferred much later. New **`prof2-residuals`** tests explicitly.

- Loop over runs and histogram absolute & relative residuals between ipol and MC, e.g. $(f(p_i) - \mathrm{MC}_i)/\mathrm{MC}_i$
- Breakdown by observable, and value / error. Easily extended.
- For better testing, train interpolation on one run subset and test on the remainder



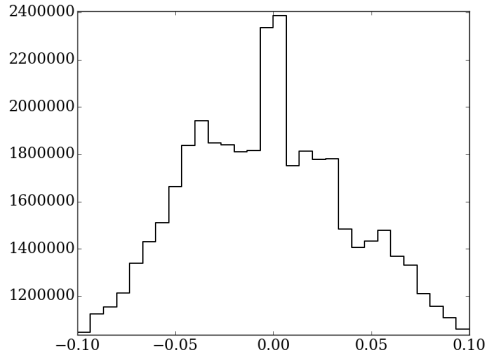


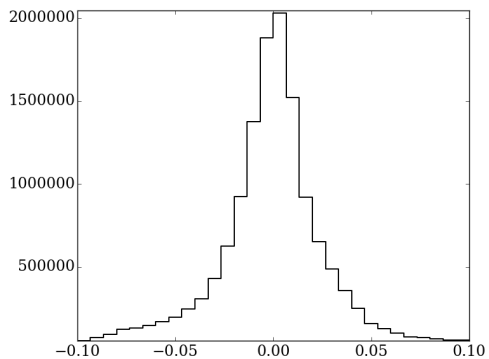
Measuring goodness-of-fit

- Historically used a simple pseudo-χ^2:

$$\chi^2(\vec{p}) = \sum_b w_b \frac{(f_b(\vec{p}) - \mathrm{ref}_b)^2}{\Delta\mathrm{ref}_b^2 + \Delta f_b^2 + \epsilon^2} \quad (1)$$

- Several limitations: no stat/syst separation, weight has $\sqrt{}$ of intuitive effect, $\Delta f_b^2 = \mathrm{median}(\Delta\mathrm{MC})$ i.e. const!
- Expt correlations available in Prof 1.4; coming soon in v2
- Linearised weights also imminent – depending on feedback. Affects correlations via cov_{ij}
- Prof 2 allows error parameterisation: greatly improves residuals. Denom is of equal importance in χ^2! *Needs to be regularised in fit.*

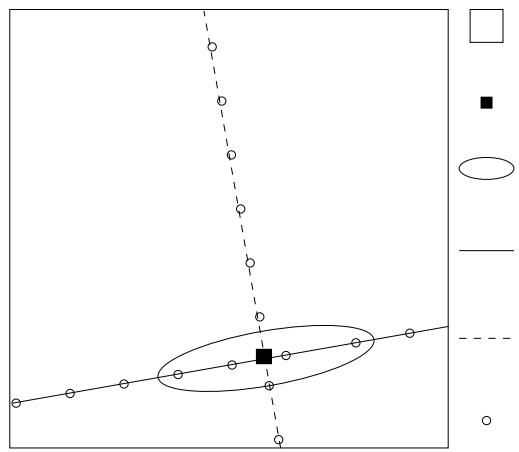




Eigentunes and coverage

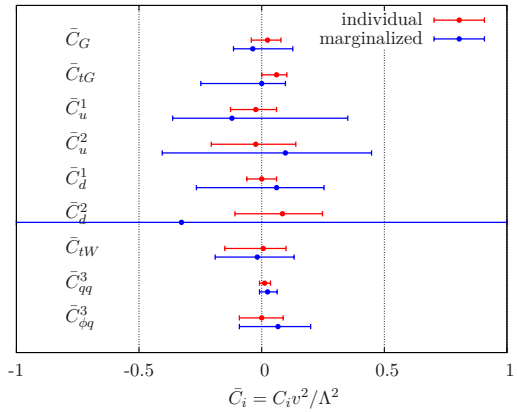
Eigentunes have gained acceptance as a "robust" way to create systematic variation tunes.

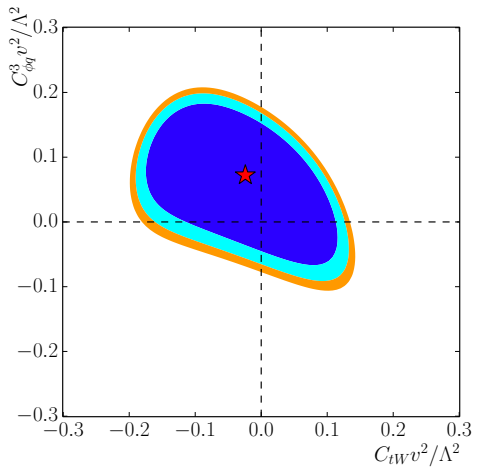
- Motivation cf. PDF Hessian errors.
- *Directions* are robust: physics in the components of principle directions
- But distance along vectors not well-defined. If true χ^2, expect $\Delta \sim$ numparams; actually more like num bins for coverage
- Effect of large correlated systematics? Not experimental. . . but in model??
- Can we define a statistically robust $\Delta\chi^2$ for tunes? Perhaps instead aim for iterated minimal data coverage.
- More robust dimensional reduction wanted / needed? cf. ATLAS A14 procedure



Prof4BSM – life beyond tuning

- Much recent development/use hasn't been for tuning at all…
- Fast parameterisation also finds use in BSM physics, e.g. **arXiv:1506.08845**, arXiv:1511.05170
- Use parameterisation of observables in Wilson coefficient space to build confidence limit contours
- Speed important for marginalising limits in many dimensions for projections





Summary & outlook

- Professor is a well-established tool to aid in many-parameter MC tuning. *Not a replacement for physics awareness.*
- Used by majority of experiment tunes, also some MC author tunes.
- Prof 1 series had become unwieldy for developers, flaky for users: Prof 2 is a leaner, faster reboot.
- Not all functionality yet replaced – taking time to think about better-motivated heuristics.
- But already some advantages like speed, format support, and uncertainty parameterisation.
- New version has so far been used more for BSM than QCD; but not for long, I hope!