Python Fully Bayesian Unfolding: PyFBU

Clement Helsens, Oliver Majersky
CERN-EP

Fully Bayesian Unfolding python implementation

- Fully based on python tools (pymc3, theano, matplotlib…), ROOT is not involved
- Developed as a standard python package
  - GitHub https://github.com/pyFBU/
  - Pypi https://pypi.python.org/pypi/fbu
- This implementation
  - Is very low level (users have to write their own interface from their ntuples and provide arrays)
  - Allows to marginalise systematic uncertainties and mitigate their effect and combine channels
  - Needs a detailed documentation with notebooks
  - The ttbar AC analysis has made a nice wrapper that we could hopefully generalize
- Disclaimer :
  - Sampling a very high dimensional phase space is time consuming
  - Pymc4 (TensorFlow Probability as backend) and other bayesian tools were tried out as it could be a nice a alternative to speed up sampling

Likelihood, output (example from ttbar Ac analysis)

$$p\left(\boldsymbol{T}|\{\boldsymbol{D}_1\cdots\boldsymbol{D}_{N_{ch}}\}\right)=\int\prod_{i=1}^{N_{ch}}\mathcal{L}\left(\boldsymbol{D}_i|\boldsymbol{R}_i(\boldsymbol{T},K_{\text{boosted}};\boldsymbol{\theta}_{\text{s}}),\boldsymbol{B}_i(\boldsymbol{\theta}_{\text{s}},\boldsymbol{\theta}_{\text{b}})\right)$$

$$\mathcal{N}(\boldsymbol{\theta}_{\text{s}})\,\mathcal{N}(\boldsymbol{\theta}_{\text{b}})\,\pi(\boldsymbol{T})\,\pi(K_{\text{boosted}})\,\mathrm{d}\boldsymbol{\theta}_{\text{s}}\,\mathrm{d}\boldsymbol{\theta}_{\text{b}},$$

- TCA example -- combined lepton+jets & dilepton channels
  - 8 regions in total, O(100) nuisance parameters, up to 20 bins per region (depending on unfolded observable)
  - 2D unfolding achieved
  - Sampling ~16 hours on 4 CPUs
- Output: full trace of all samples for
  - Nuisance parameters
  - Unfolded bins
- Stored as numpy arrays