First steps towards an improved tuning method for Monte Carlo generators

Fabian Klimpel, Stefan Kluth, Andrea Knue
DPG annual meeting 2017
Münster, 27.03.2017
1. Introduction to parameter tuning
2. Reproduction of a specific tuning
3. Introducing new approaches for parameter tuning
4. Outlook
Why parameter tuning?

- Before the particle collision:
  - Initial state is (approximately) known
- After the collision:
  - Final state (hadrons etc.) is measured
- In MC generators, models describe the transition
- The model parameters need values in order to reproduce measurements

**But**: How to get their values?
• Model parameters can be varied in certain intervals without creating strange results - parameters need to be set to values that describe the data best = tuning

27.03.2017
1. Sample random parameter sets in predefined range
2. Use the sets as input for the MC generator
3. Extract observables from each MC configuration
   -> binned histograms for each observable
4. Interpolate every bin independently by fixed order polynomial function
5. Perform a simultaneously $X^2$ minimisation by comparing the interpolation with the reference data
Recommendation from the Professor authors:

1. create subsamples of all performed MC runs
2. Perform the interpolation & tuning for every subset independently
3. Extract the parameters based on their frequency
1. Introduction to parameter tuning
2. Reproduction of a specific tuning
3. Introducing new approaches for parameter tuning
4. Outlook
What to tune?

• For comparison, tune was performed on *existing e+e- tune*

• Hard process:

• Events simulated with Pythia 8 at leading order

• $\sqrt{s} = m(Z^0)$

• Analyses from LEP (Aleph, Opal, Delphi), DESY (Jade) & PDG data

  -> 103 histograms / 1115 bins

  • Unfolded data to particle level
  • Published in Rivet framework [1] [2] [3] [4] [5]
Parameters to tune?

1. From parton shower:
   - High energy partons radiate with the coupling strength $\alpha_s$ -> shower
   - Shower cutoff at $p_T,\text{min}$
   - Then: confinement -> Hadronisation, hadron decays

2. From hadronisation (Lund-String model):
   - $f(z) \propto \frac{1}{z} (1 - z)^{a_{\text{Lund}}} \exp \left( -b_{\text{Lund}} \frac{m_T^2}{z} \right)$
   - Baryons: $a = a_{\text{Lund}} + a_{\text{ExtraDiquark}}$
   - $P(p_T) \propto \exp \left( -\frac{1}{2} \frac{p_T^2}{\sigma^2} \right)$

-> overall 6 parameters to tune

- Run-combinations & interpolation:
  - 650 parameter sets
  - 300 subsets with 500 parameter sets each
  - 6D Interpolation with polynomial of fourth order

Bild: 3D Plot
Tuning result with parameter ranges

- Parameter runs into limit of given range
  - need to extend range
- The interpolation functions should have similar behaviour
  - based on similar parameter sets
  - extrapolation should give similar parameter estimations
• Double peaks should not exist
- result is sensitive to input set
• Is there a problem while tuning? Did they converge in a local minimum?
- re-tune by using another method
1. Introduction to parameter tuning
2. Reproduction of a specific tuning
3. Introducing new approaches for parameter tuning
4. Outlook
Bayesian Analysis Toolkit (BAT)

• Use BAT as control tune for Professor

• Working principle:
  • Based on self adapting Markov chain
  • Steps determined by Metropolis-Hastings algorithm

• Benefits:
  • This delivers topological information about the likelihood
  • The algorithm can find the maximum likelihood

• BAT delivers very similar result

-> It seems to be not a problem of the tune itself
Interpolation

• Professor uses a fixed order polynomial function for interpolation
  -> possible over-/underfitting?

• A convergence should not be judged upon a simple $X^2$ only
  • Small $X^2$ $\not\implies$ good fit

-> another approach could avoid these
Interpolation

• Goals:
  • Iteratively increasing of the number of monomials
  • Checking if it is better / worse
  • Search for the best iteration

• Quality judgment:
  • Calculate:
    • \( n_{Simulation\text{data},i} = \frac{1}{\|\nabla f_{Simulation\text{data}}\|} \nabla f_{Simulation\text{data}} \mid_{x_i} \)
    • \( n_{Interpolation,i} = \frac{1}{\|\nabla f_{Interpolation}\|} \nabla f_{Interpolation} \mid_{x_i} \)
  • New parameter: \( D_{Smooth} = \frac{1}{N} \sum_{i=1}^{N} n_{Simulation\text{data},i} \cdot n_{Interpolation,i} \)

-> search for minimum of \( f = X^2 \frac{(1 - D_{Smooth})}{(1 + D_{Smooth})} \)

27.03.2017
Tuning result with parameter ranges

- Tuned parameter are also at the upper limit of the parameter range

-> again need to extend the range
Tuning result without parameter ranges

- Extending the range creates only one maximum

-> a reasonable value can be extracted
1. Introduction to parameter tuning
2. Reproduction of a specific tuning
3. Introducing new approaches for parameter tuning
4. Outlook
Outlook

- Exchange uncertainty calculation
  - Usage of covariance matrices of the fit parameters

- Reject bins if
  - MC generator cannot reproduce the data for all parameter sets
  - The theory is too inaccurate (especially LO)

- Redefine parameter ranges to avoid running into limit and repeat tune
• Tuning approach performed by Professor 2.1.4 (potentially) unstable

• Final tuning seems to work properly

• Problem caused by the interpolation algorithm

• Using new approach to increase stability with
  1. An iterative construction that increases the number of monomials
  2. A new break-off / quality parameter $f = X^2 \frac{(1 - DS_{mo_{th}})}{(1 + DS_{mo_{oth}})}$