Estimating tuning uncertainties with Professor

Holger Schulz, Heiko Lacker, Jan Eike von Seggern (HU Berlin),
Andy Buckley (Edinburgh), Hendrik Hoeth (Durham)

CERN, January 13, 2010

Oversampling and interpolation quality

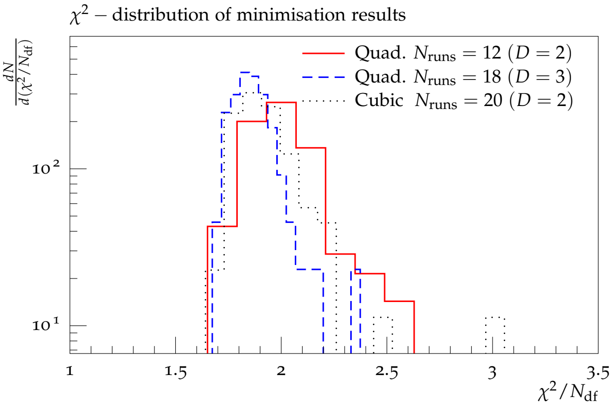
- In n dimensions: Singular Value Decomposition (SVD) requires $N_{\text{min}}^{(n)}$ generator runs:

$$N_{\text{min}}^{(n)} = 1 + n + n(n+1)/2 + \underbrace{(n+1)(n+2)/6}_{\text{cubic only}}$$

- SVD allows for oversampling.
- Degree of oversampling: $D = N_{\text{runs}} / N_{\text{min}}^{(n)}$
- What is a sensible D?
- $\rightarrow$ use $\mathcal{O}(1000)$ different interpolations with different N_{runs}
- Perform minimisations, investigate g.o.f. measures
- Examples shown are from a two-dimensional Tuning of Jimmy

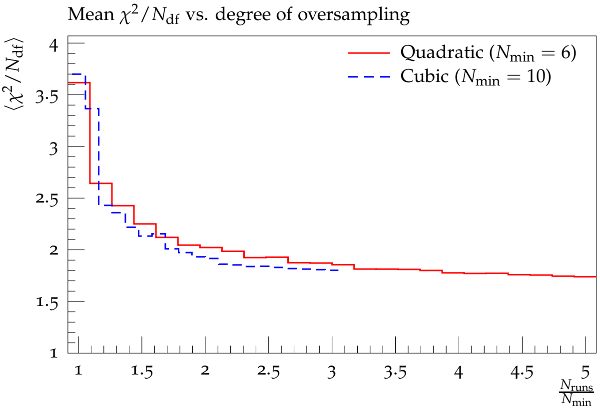
Distribution of Goodness of Fit values

- Spread of results decreases with increasing D, polynomial degree
- Observe lower χ^2 / N_{df}-boundary



Goodness of Fit vs. degree of oversampling

- Oversampling is neccesary, $D > 2 \ldots 3$ seems sensible
- Hower, g.o.f. improves slowly for $D > 4$, almost saturates

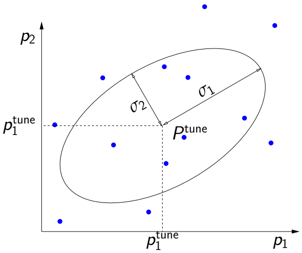


Tuning uncertainties (work in progress)

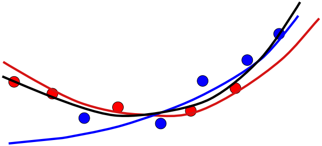
Goal: establish a robust estimate of tuning uncertainties (confidence-belt)

We currently study two different sources of tuning uncertainties:

- Statistical uncertainties → exploit covariance matrix returned by minimiser (inspired by NNPDF approach)

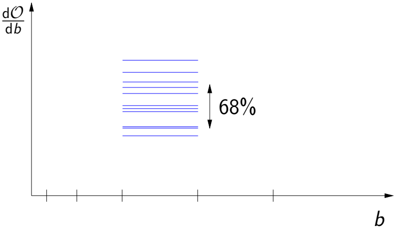


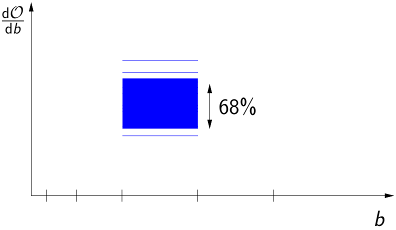
- Intrinsic systematics of the Professor method: freedom when parameterising generator response → many minimisation results



Confidence belt construction

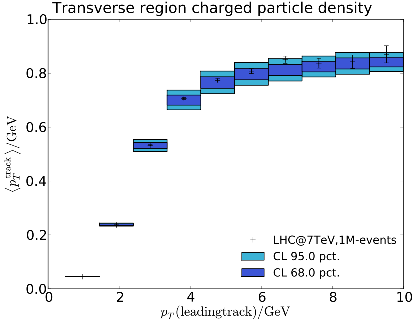
1. Use points sampled from ellipsis or different minimisation results
2. Use parameterisation to get bin-content predictions
3. For each bin b and each observable $\mathcal{O}$: determine central 68, 95 pct.



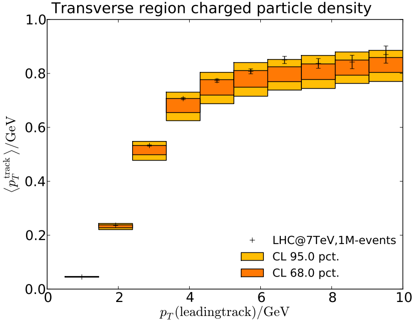


Confidence belt - without pseudodata

statistical uncertainties

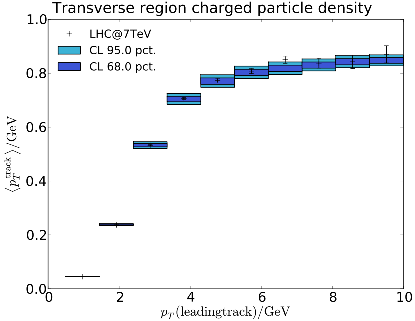


"systematic" uncertainties

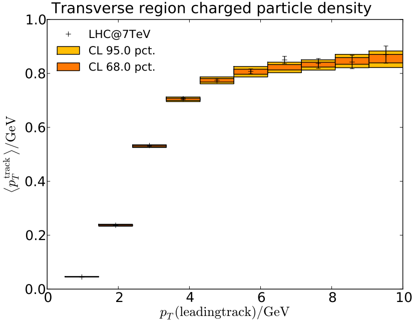


Confidence belt - adding pseudodata

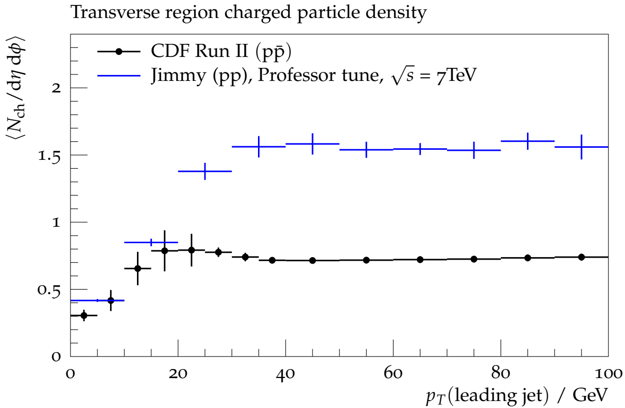
statistical uncertainties



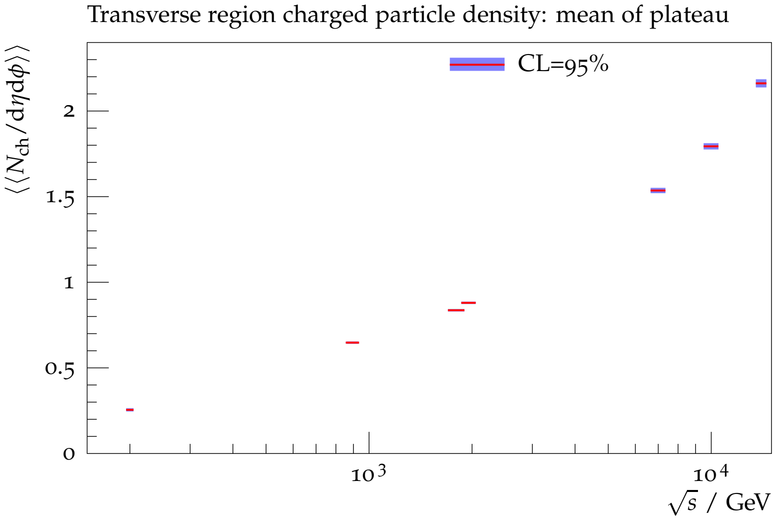
"systematic" uncertainties



Underlying event plateau. . .



... ITS MEAN EVOLVING WITH $\sqrt{s}$



Summary

- We studied how the interpolation benefits from oversampling
- $N_{\text{runs}} / N_{\text{min}}^{(n)} > 2 \ldots 3$ is advisable
- Working on quantification of tuning uncertainties
- Statistical uncertainty estimate shows expected behaviour
- More work, especially on systematic uncertainty estimate needed

Thank you!

Backup

2nd order polynomial includes lowest-order correlations between parameters

$$MC_b(\vec{p}\,) \approx f^{(b)}(\vec{p}\,) = \alpha_0^{(b)} + \sum_i \beta_i^{(b)}\, p'_i + \sum_{i \le j} \gamma_{ij}^{(b)}\, p'_i\, p'_j$$

Now use N generator runs, i.e. N different parameter sets x,y:

$$\underbrace{\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}}_{\vec{v}\text{ (N values, i.e. N bin contents)}} = \underbrace{\begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 \\ & & & \vdots & & \\ 1 & x_N & y_N & x_N^2 & x_N y_N & y_N^2 \end{pmatrix}}_{\tilde{\mathbf{P}}\text{ (N sampled parameter sets)}} \underbrace{\begin{pmatrix} \alpha_0 \\ \beta_x \\ \beta_y \\ \gamma_{xx} \\ \gamma_{xy} \\ \gamma_{yy} \end{pmatrix}}_{\vec{c}\text{ (coeffs)}}$$

Therefore: $\boxed{\vec{c}_b = \tilde{\mathcal{I}}[\tilde{\mathbf{P}}]\vec{v}}$ where $\tilde{\mathcal{I}}$ is the pseudoinverse operator.

$$\boxed{\vec{c}_b = \tilde{\mathcal{I}}[\tilde{\mathbf{P}}]\vec{v}}$$

- Use Singular Value Decomposition (SVD), a general diagonalisation for all normal matrices M: $M = U\Sigma V^*$
- Method available in SciPy.linalg
- Minimal number of runs = number of coefficients in $\vec{c}_b$:
  $$N_{\text{min}}^{(n)} = 1 + n + n(n+1)/2 + \underbrace{(n+1)(n+2)/6}_{\text{cubic only}}$$
- Oversampling by a factor of three has proven to be much better

Num params, P	$N_2^{(P)}$ (2nd order)	$N_3^{(P)}$ (3rd order)
1	3	4
2	6	10
4	15	35
6	28	84
8	45	165
9	55	220

$$\vec{c}_b = \tilde{\mathcal{I}}[\tilde{\mathbf{P}}]\vec{v}$$

- Use Singular Value Decomposition (SVD), a general diagonalisation for all normal matrices M:$M = U\Sigma V^*$
- Method available in SciPy.linalg
- Minimal number of runs = number of coefficients in $\vec{c}_b$:
  $N_{\text{min}}^{(n)} = 1 + n + n(n+1)/2 + \underbrace{(n+1)(n+2)/6}_{\text{cubic only}}$
- Oversampling by a factor of three has proven to be much better

Num params, P	$N_2^{(P)}$ (2nd order)	$N_3^{(P)}$ (3rd order)
1	3	4
2	6	10
4	15	35
6	28	84
8	45	165
9	55	220

$$\boxed{\vec{c}_b = \tilde{\mathcal{I}}[\tilde{\mathbf{P}}]\vec{v}}$$

- Use Singular Value Decomposition (SVD), a general diagonalisation for all normal matrices M:$M = U\Sigma V^*$
- Method available in SciPy.linalg
- Minimal number of runs = number of coefficients in $\vec{c}_b$:
  $N_{\mathrm{min}}^{(n)} = 1 + n + n(n+1)/2 + \underbrace{(n+1)(n+2)/6}_{\text{cubic only}}$
- Oversampling by a factor of three has proven to be much better

Num params, P	$N_2^{(P)}$ (2nd order)	$N_3^{(P)}$ (3rd order)
1	3	4
2	6	10
4	15	35
6	28	84
8	45	165
9	55	220