Розгортання (unfolding)

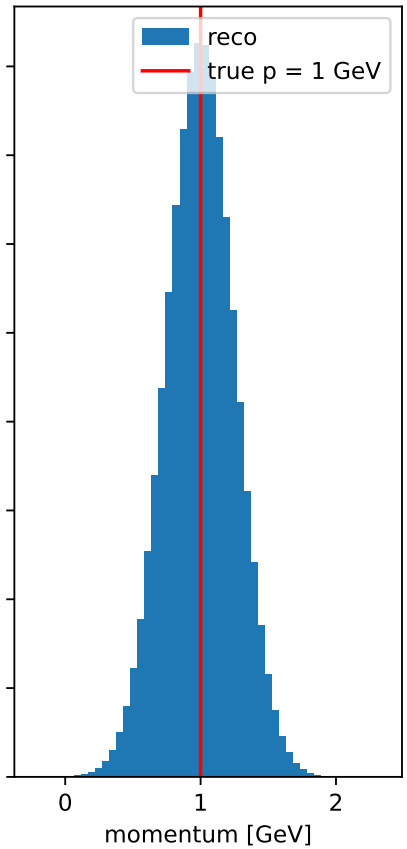
Олександр Зенаєв

Розгортання (unfolding): формулювання

- Unfolding (розгортання, відновлення, деконволюція) в експериментальній фізиці – математичний метод відновлення справжнього (істинного) розподілу фізичної величини на основі спостережуваних (виміряних) даних
- Unfolding є ключовою проблемою при вимірюванні розподілів (наприклад, диференційних перерізів народження частинок, $\frac{\mathrm{d}\sigma}{\mathrm{d}X}$), що зазнають суттєвого впливу через обмежену роздільну здатність детектора
- Експериментально вимірюють диференційні перерізи, що усереднені за бінами

$$\frac{\mathrm{d}\sigma}{\mathrm{d}X} \sim \frac{\Delta\sigma}{\Delta X} \sim \frac{\Delta N}{\Delta X}$$

- Через обмежену роздільну здатність, події мігрують між бінами: події у справжньому біні ΔX_1 можуть бути реконструйовані в іншому біні ΔX_2



M. Kuusela, PHYSTAT Conference on Unfolding, 2024

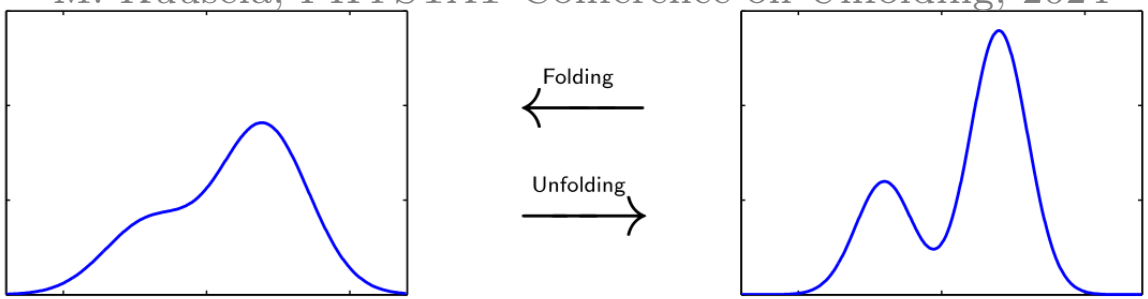


Figure: Smeared spectrum

Figure: True spectrum

Розгортання (unfolding): розв'язок

- Задача не має однозначного розв'язку (некоректна постановка задачі, ill-posed problem): нескінчена кількість параметрів
- Необхідна регуляризація:
  - обмеження кількості невідомих параметрів (біни), але через статистичні похибки все одне виникають швидко осцилюючі компоненти (high frequency components)
  - додавання додаткових обмежень на розв'язок (гладкість) для придушення швидко осцилюючих компонент
- Існує декілька бібліотек для розгортання, доступних через інтерфейс RooUnfold. Інструкція: https://gitlab.cern.ch/RooUnfold/RooUnfold/-/blob/master/README.md

```
git clone https://gitlab.cern.ch/RooUnfold/RooUnfold.git
cd RooUnfold
mkdir build
cd build
cmake ..
make -j4
cd ..
source build/setup.sh
```

- Ми розглянемо метод матричного розгортання, реалізований в TUnfold

2.2 Algorithm

The unfolding algorithm, as implemented in TUnfold, determines the stationary point of the "Lagrangian"

$$\mathcal{L}(x,\lambda) = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 \qquad \text{where} \tag{3}$$

$$\mathcal{L}_1 = (\boldsymbol{y} - \mathbf{A}\boldsymbol{x})^{\mathsf{T}} \mathbf{V_{yy}}^{-1} (\boldsymbol{y} - \mathbf{A}\boldsymbol{x}), \tag{4}$$

$$\mathcal{L}_2 = \tau^2 (\boldsymbol{x} - f_b \boldsymbol{x_0})^{\mathsf{T}} (\mathbf{L}^{\mathsf{T}}\mathbf{L})(\boldsymbol{x} - f_b \boldsymbol{x_0}), \tag{5}$$

$$\mathcal{L}_3 = \lambda (Y - \boldsymbol{e}^{\mathsf{T}}\boldsymbol{x}) \qquad \text{and} \tag{6}$$

$$Y = \sum_i y_i, \tag{7}$$

$$e_j = \sum_i A_{ij}, \tag{8}$$

The term $\mathcal{L}_1$ is what one expects from a least square minimisation. The vector $\boldsymbol{y}$ has n rows. The covariance matrix $\mathbf{V_{yy}}$ of $\boldsymbol{y}$ is diagonal in many cases, such that the diagonals hold the squares of the uncertainties. TUnfold also supports the use of non-diagonal $\mathbf{V_{yy}}$. The vector $\boldsymbol{x}$ corresponds to the unfolding result and has m rows. The elements A_{ij} of $\mathbf{A}$ describe for each row j of $\boldsymbol{x}$ the probabilities to migrate to bin i of $\boldsymbol{y}$. The matrix $\mathbf{A}$ often is determined using Monte Carlo simulations.

The term $\mathcal{L}_2$ describes the regularisation, which damps fluctuations in $\boldsymbol{x}$. Such fluctuations originate from the statistical fluctuations of $\boldsymbol{y}$, which are amplified when determining the stationary point of equation 3. The parameter τ^2 gives the strength of the regularisation. It is considered as a constant

Регуляризація

- Регуляризація призначена для зменшення похибок на розв'язок (variance) ціною зміщення (викривлення) результату (bias)
- Ключовою проблемою є знаходження балансу між похибками і зміщенням (bias-variance tradeoff). Це можна розуміти як баланс між статистичною і систематичною похибкою.
- Необхідною умовою є виконання покриття (coverage): наскільки добре розв'язок описує істинний розподіл в межах похибок
- Концепція покриття перевіряється через тести покриття (closure tests, cover tests): використовуючи МК симуляцію (коли відомий істинний розподіл), на штучних прикладах тестують розв'язок, наскільки добре він відтворює істинний розподіл. На основі цих тестів оптимізують розгортання (обирають метод та регуляризацію)
- Якщо можна, краще уникати регуляризації

Практичне заняття

- Згенерувати експоненційний розподіл (імпульси частинок)
- Викривити згенерований розподіл, накладаючи розмиття за нормальним розподілом із додатковим зсувом
- Виконати розгортання за допомогою бібліотеки TUnfold (через інтерфейс RooUnfold)
- Порівняти результати розгортання без регуляризації та з регуляризацією

Github:
https://github.com/zenaiev/hep/blob/main/unfold/unfold.py

Google Colab:

[under development...]