Розгортання (unfolding)

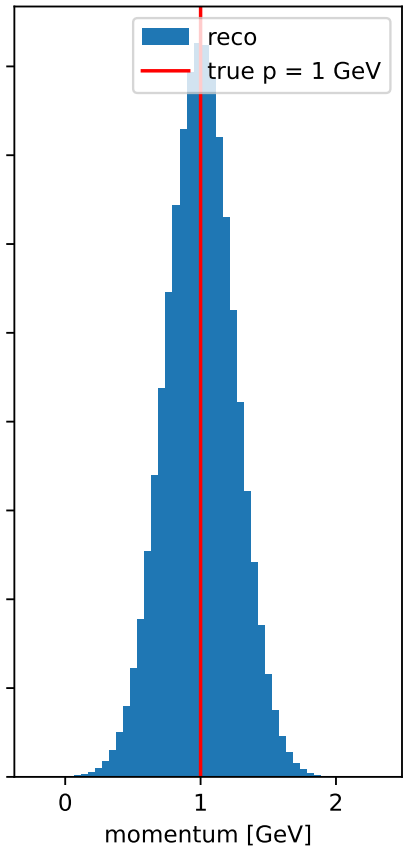
Олександр Зенаєв

Розгортання (unfolding): формулювання

- Unfolding (розгортання, відновлення, деконволюція) в експериментальній фізиці – математичний метод відновлення справжнього (істинного, true) розподілу фізичної величини на основі спостережуваних (виміряних, reconstructed, reco) 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

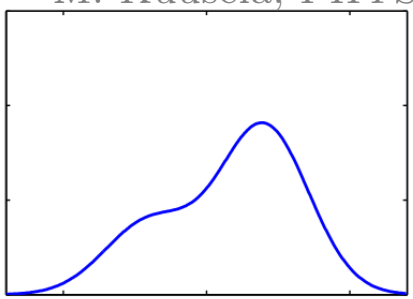
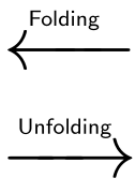


Figure: Smeared spectrum



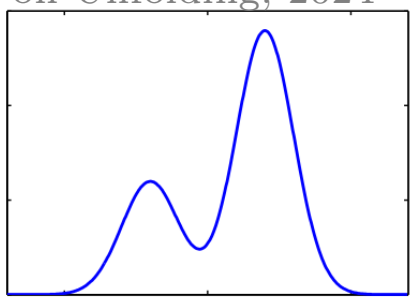


Figure: True spectrum

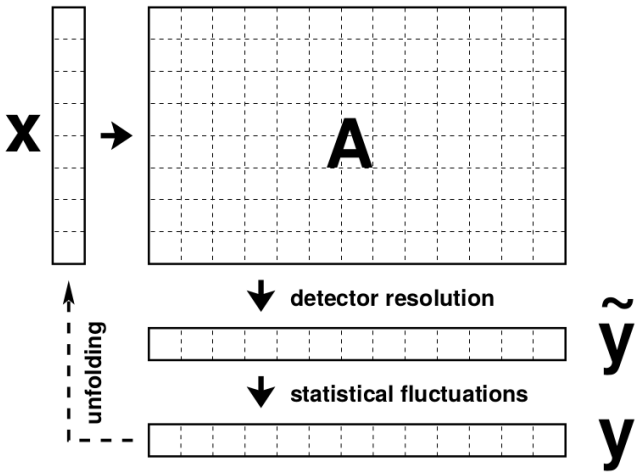
Розгортання (unfolding): розв'язок

- 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. Він дозволяє використовувати більшу кількість бінів на спостережуваному рівні, ніж на істинному рівні, що допомагає стабілізувати розв'язок.

TUnfold



- Unfolding problem: $\boldsymbol{\mu} = \boldsymbol{A}\boldsymbol{x}$
- $\boldsymbol{\mu}$: detector expectation (M_y bins), $\boldsymbol{V_{yy}}$ is its covariance
- $\boldsymbol{A}$: response matrix (taken from MC)
- $\boldsymbol{x}$: unknown truth, M_x bins ($M_x \leq M_y$)
- $\mathcal{L} = (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})^T \boldsymbol{V_{yy}}^{-1} (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) + \tau^2 (\boldsymbol{x} - \boldsymbol{x_0})^T (\boldsymbol{L}^T \boldsymbol{L})(\boldsymbol{x} - \boldsymbol{x_0})$
- $\frac{\partial \mathcal{L}}{\partial x} = 0 \quad \Rightarrow \quad \boldsymbol{x} = \boldsymbol{x}(\boldsymbol{y}, \boldsymbol{V_{yy}}, \boldsymbol{x_0}),\ \boldsymbol{V_{xx}} = \boldsymbol{V_{xx}}(\boldsymbol{y}, \boldsymbol{V_{yy}}, \boldsymbol{x_0})$
- $\boldsymbol{x_0}$: bias vector for regularization (taken from MC)

- Регуляризація: накладання обмеження (шляхом додавання χ^2 penalty) на відмінність розв'язку x від x_0 (зазвичай МК симуляції), або (краще) його другої похідної (кривини)
- TUnfold підтримує багатовимірні розподіли
- Є декілька методів для підбору ступені регуляризації (regularisation strength)
- $len(x) \leq len(y)$

2.2 Algorithm

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

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

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

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

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

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

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

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 while determining the stationary point of $\mathcal{L}$. The matrix $\mathbf{L}$ has m columns and n_R rows, corresponding to n_R regularisation conditions. The bias vector $f_b \boldsymbol{x_0}$ is composed of a normalisation factor f_b and a vector $\boldsymbol{x_0}$. In the simplest case, one has $f_b = 0$, $n_R = m$ and $\mathbf{L}$ is the unity matrix. In that case, $\mathcal{L}_2$ simplifies to $\tau^2 ||\boldsymbol{x}||^2$, effectively suppressing large deviations of $\boldsymbol{x}$ from zero. If $f_b = 1$, deviations of $\boldsymbol{x}$ from $\boldsymbol{x_0}$ are suppressed. Choices of the matrix $\mathbf{L}$ different from the unity matrix are discussed in section 7.

Регуляризація

- Регуляризація призначена для зменшення похибок на розв'язок (variance) ціною зміщення результату (викривлення, 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

[Python] https://github.com/zenaiev/hep/blob/main/unfold/unfold.py
[C++] https://github.com/zenaiev/hep/blob/main/unfold/unfold.cpp

Google Colab:

https://colab.research.google.com/github/zenaiev/hep/blob/main/unfold/unfold_cpp.ipynb