

V. FORMATO - 5/12/18

UNFOLDING

THE UNFOLDING PROBLEM

Consideriamo una variabile aleatoria y .

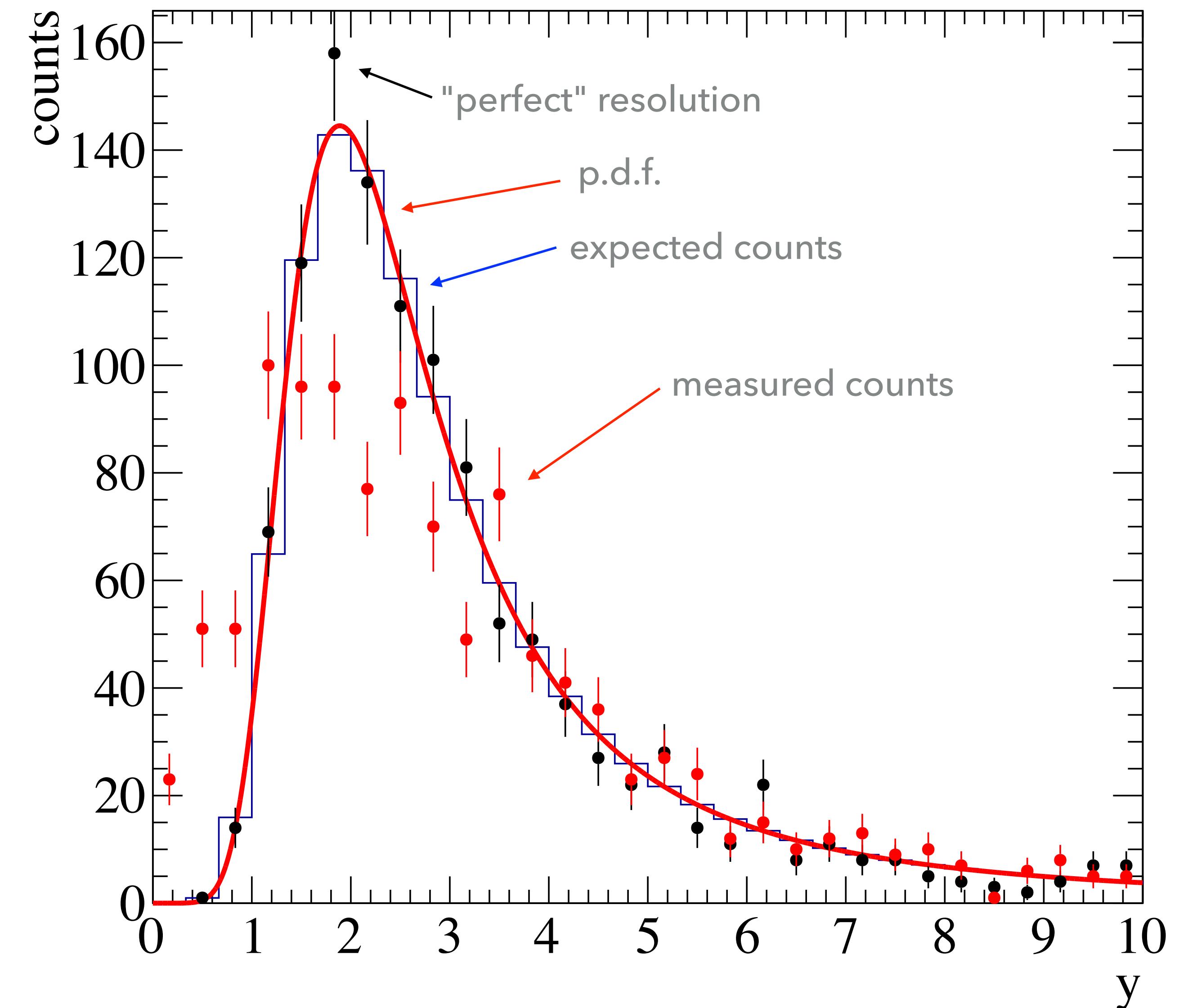
Potremmo volerne misurare la distribuzione (p.d.f.) $f(y)$, oppure, se la p.d.f. e' nota, potremmo voler trovare i parametri...

La presenza della risoluzione sperimentale complica le cose:

In generale la $f(y)$ viene "sbrodolata" dalla risoluzione: i picchi si allargano, le code si popolano, etc...

$$\tilde{f}(x) = \int K(x|y)f(y) dy$$

measured distribution response function true distribution



Migration

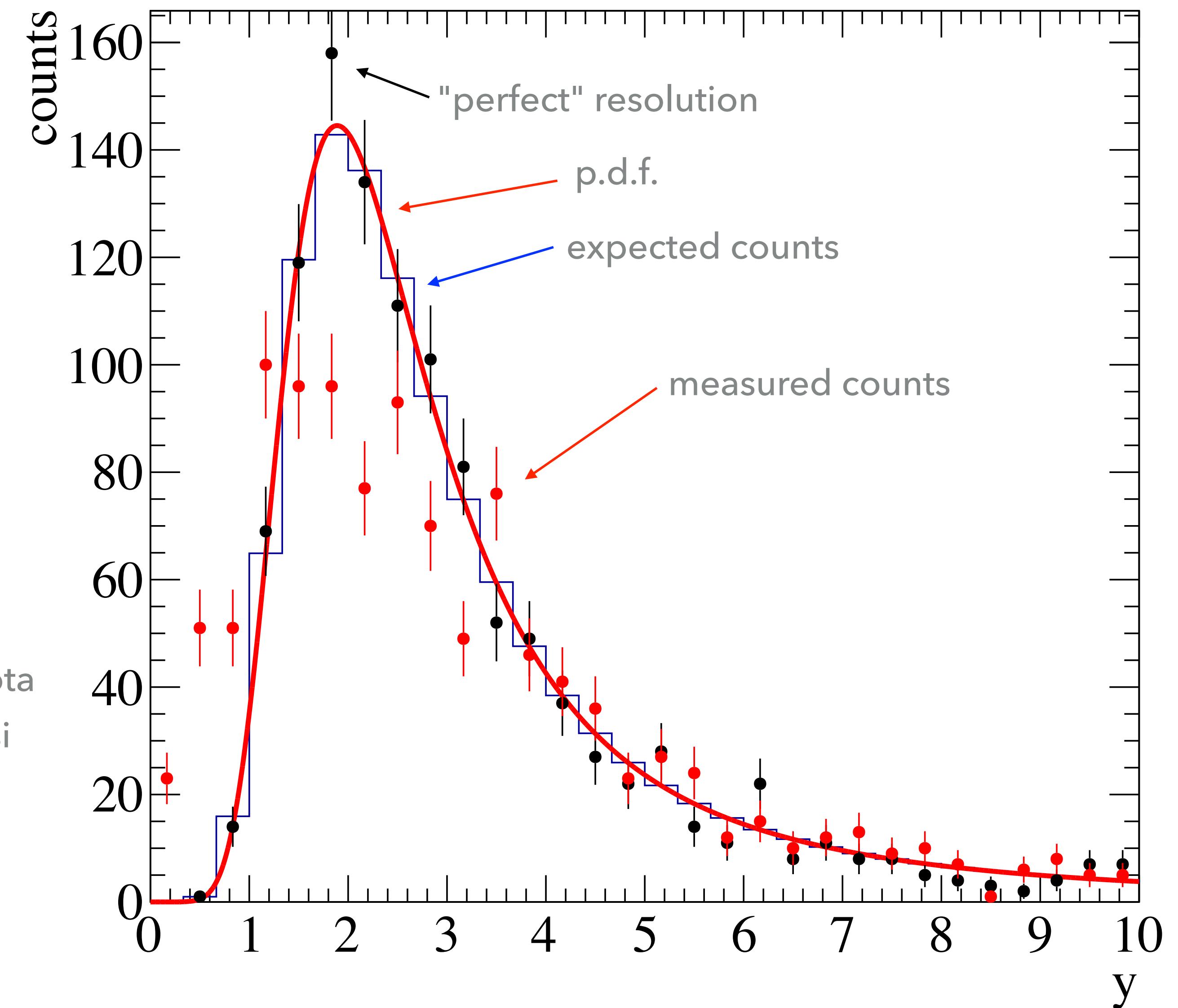
$$\tilde{f}(x) = \int K(x|y)f(y) dy$$

Equazione di Fredholm di tipo 1

L'obiettivo e' trovare $f(y)$: abbiamo bisogno di uno stimatore (possibilmente unbiased, efficiente e consistente)

Se c'e' un input dalla teoria con una determinata parametrizzazione $f(y; \theta)$ possiamo "foldarla" con K e comparare il risultato alla misura \rightarrow stimare i parametri.

Se la forma di $f(y)$ non e' nota il problema dell'unfolding si traduce in "Trovare $f(y)$ a partire da $f(x)$ "



Migration

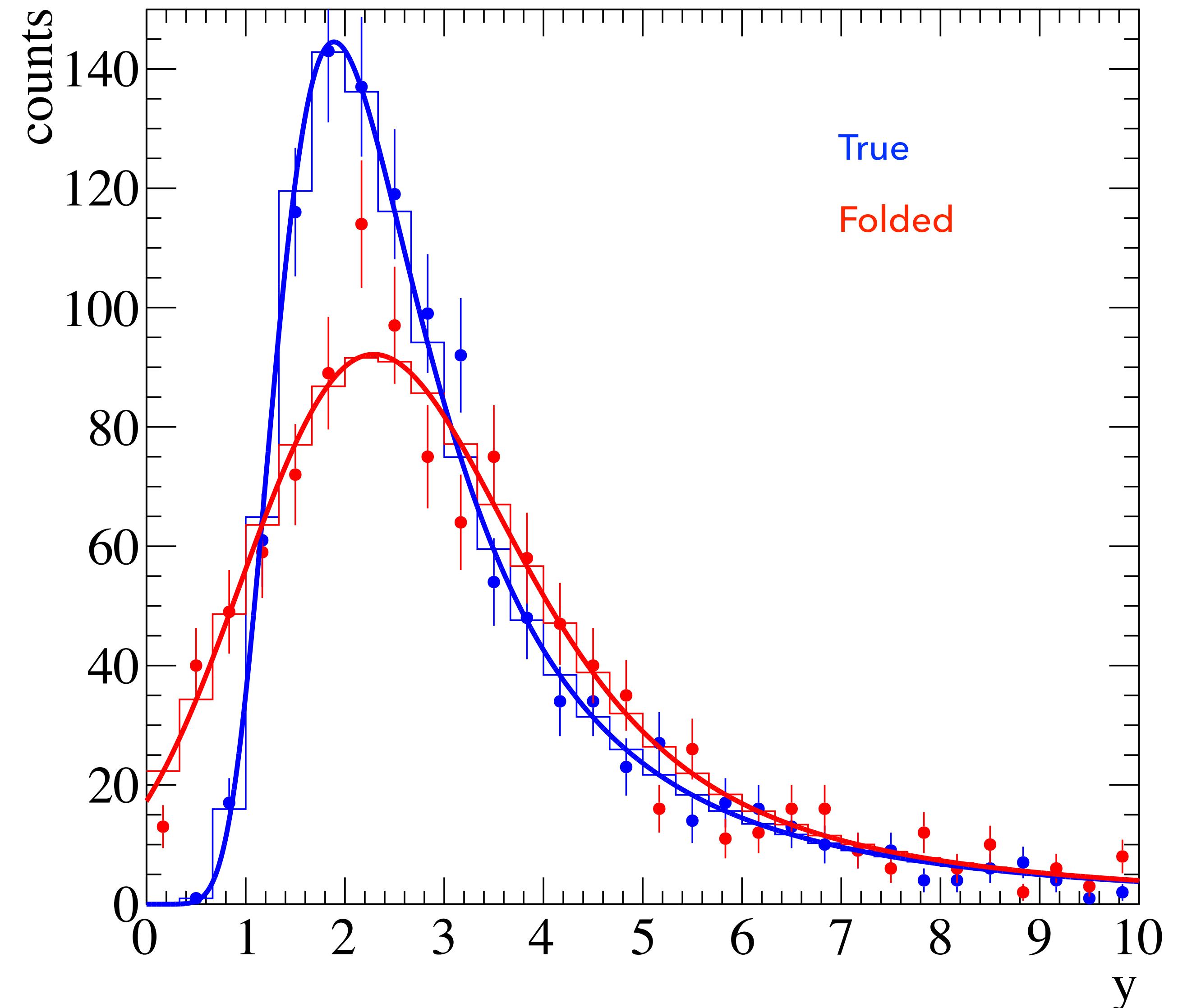
Operativamente un esperimento e' limitato dalla risoluzione e dalla statistica → passare ad istogrammi: discretizzazione.

$$f(y) \rightarrow \mu_i = \int_{y_i}^{y_{i+1}} f(y) dy$$

$$\tilde{f}(x) \rightarrow \nu_i = \int_{x_i}^{x_{i+1}} \tilde{f}(x) dx$$

$$K(x|y) \rightarrow R_{ij} = \frac{\mu_j}{\nu_i}$$

$$\nu_i = \mathbb{E}(n_i) = \sum_{j=1}^M R_{ij} \mu_j$$



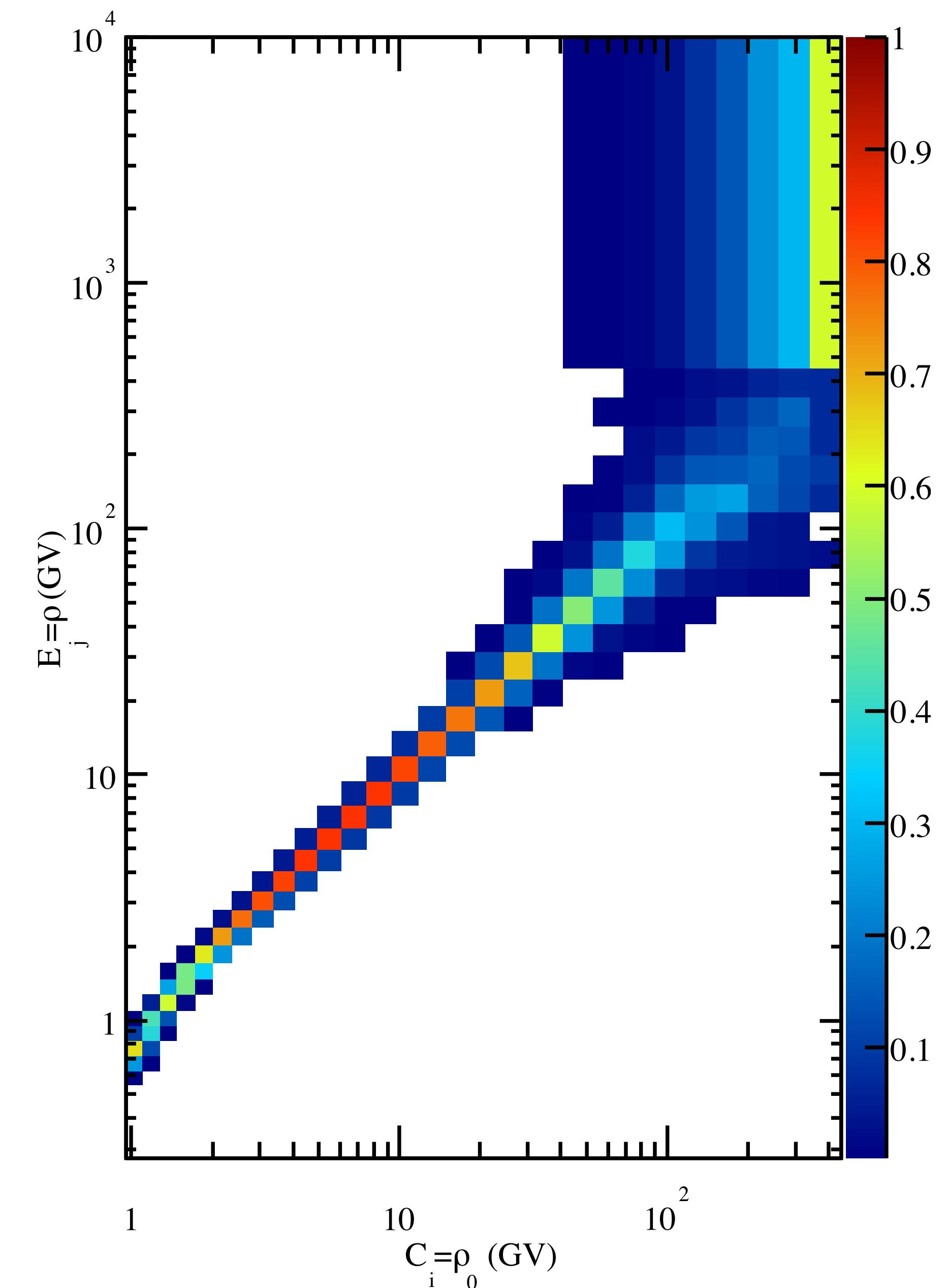
THE MATRIX

La definizione generale e'

$$R_{ij} = \frac{\int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} K(x|y) dx dy}{\int_{x_j}^{x_{j+1}} f(y) dy}$$

Se il kernel K e' noto puo' essere ricavata analiticamente, altrimenti la procedura standard e' quella di ricavare la matrice di risoluzione da una simulazione Montecarlo

Gli eventi sono generati secondo una distribuzione f_{Sim} , la migliore stima possibile di f , mappati su g_{Sim} che diventa la nostra stima migliore di g .



EFFICIENCY

Alcuni eventi di interesse potrebbero essere persi e non misurati per via di inefficienza del detector o per via di zone morte nell'accettanza. Queste possono essere incluse nella matrice di risoluzione.

$$\sum_{j=1}^M R_{ij} = \mathbb{P}(\text{osservato in un bin qualunque} \mid \text{true value nel bin } i) = \varepsilon_i$$

In generale nel nostro sample possono esserci eventi che provengono da background a modificare la distribuzione di x

$$\tilde{f}(x) = \int K(x|y) f(y) + b(y) dy$$

$$\beta_i = \int_{y_i}^{y_{i+1}} b(y) dy$$

$$\mathbb{E}(n_i) = \nu_i = \sum_{j=1}^M R_{ij} \mu_j + \beta_i$$

FIRST SOLUTION: MAXIMUM LIKELIHOOD

Ci troviamo di fronte al problema $\nu = R\mu + \beta$. Sembra naturale cercare la soluzione come

$$\mu = R^{-1}(\nu - \beta)$$

Assumendo che i dati siano distribuiti poissonianamente

$$\mathbb{P}(n_i|\nu_i) = \nu_i^{n_i} \frac{e^{-\nu_i}}{n_i!}$$

Possiamo calcolare la log-likelihood: $\log \mathcal{L}(\mu) = \sum_{i=1}^N n_i \log \nu_i - \nu_i \log n_i!$ il cui massimo e' (senza troppa sorpresa)

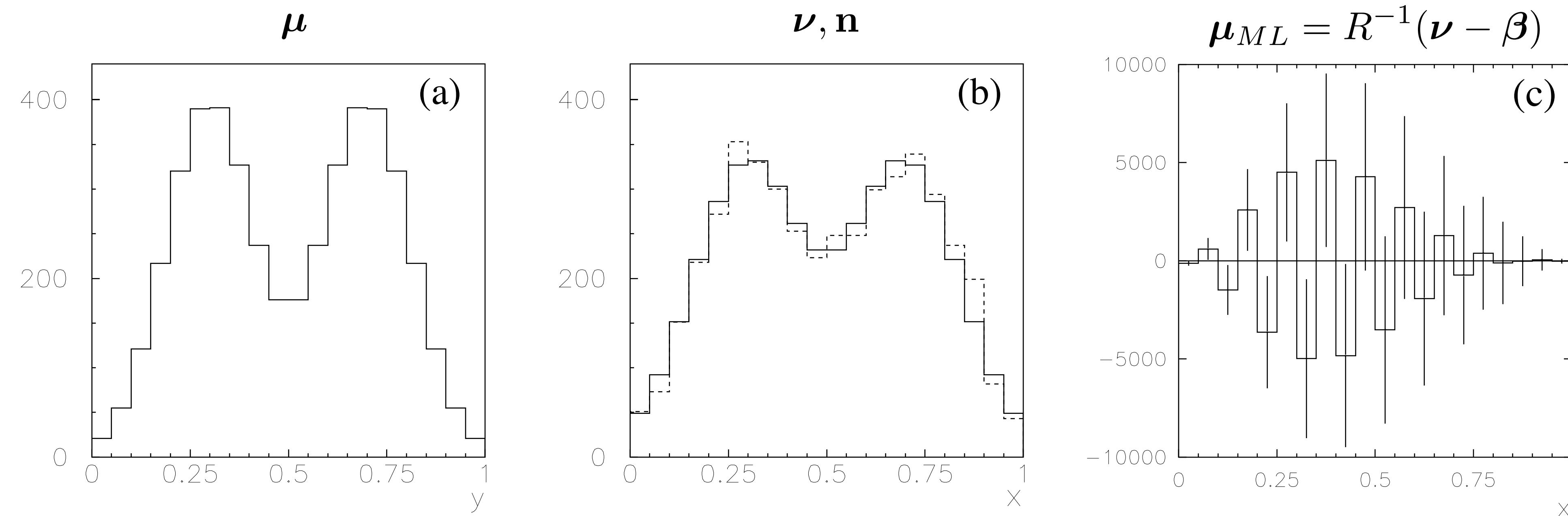
$$\nu_{ML} = \mathbf{n} \quad \Rightarrow \quad \mu_{ML} = R^{-1}(\nu - \beta)$$

...funzionerà?



FIRST SOLUTION: MAXIMUM LIKELIHOOD

(Esempio preso da <http://www.ippp.dur.ac.uk/Workshops/02/statistics/proceedings/cowan.ps>)



- 1) Fluttuazioni statistiche non sono contemplate in R , applicare R^{-1} semplicemente le amplifica, pensando siano "fisiche"
- 2) In genere il problema, cosi' formulato, non e' ben posto. R potrebbe anche non essere invertibile...

INTERMISSION: SINGULAR VALUE DECOMPOSITION

Prendete una matrice $m \times n$. Chiamatela A .

Fattorizzatela come $A = USV^T$

Dove U è una matrice $m \times m$, V una matrice $n \times n$ e S una matrice $m \times n$ con valori positivi non nulli sulla diagonale (s_i).

I valori s_i sono chiamati "singular values" e le colonne di U e V sono chiamati "singular vectors" sinistri e destri.

A cosa serve? 🤔

È una generalizzazione del concetto di "diagonalizzazione" di una matrice. (<http://people.csail.mit.edu/harinoff/320/SingularValueDecomposition.pdf>)

Esempio: $A = \frac{1}{2} \begin{pmatrix} 1 + \varepsilon & 1 - \varepsilon \\ 1 - \varepsilon & 1 + \varepsilon \end{pmatrix}$  $U = V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $S = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}$

$$A^{-1} = VS^{-1}U^T = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2\varepsilon} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

INTERMISSION: SINGULAR VALUE DECOMPOSITION

Esempio:

$$\mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, \quad \beta = \mathbf{0}, \quad V = \begin{pmatrix} n_1 & 0 \\ 0 & n_2 \end{pmatrix}$$

$$R\boldsymbol{\mu} = \mathbf{n} \quad \Rightarrow \quad S(V^T \boldsymbol{\mu}) = U^T \quad \Rightarrow \quad S\mathbf{z} = \mathbf{d}$$

$$\mathbf{z} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mu_1 + \mu_2 \\ \mu_1 - \mu_2 \end{pmatrix}, \quad \mathbf{d} = \frac{1}{\sqrt{2}} \begin{pmatrix} n_1 + n_2 \\ n_1 - n_2 \end{pmatrix}$$

Ma il sistema ora è diagonale e si risolve facilmente, e possiamo ruotare "all'indietro"

$$\boldsymbol{\mu} = V\mathbf{z} = \frac{b_1 - b_2}{2\varepsilon} \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \frac{b_1 + b_2}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

La soluzione è una combinazione lineare pesata di due autovettori della matrice di risoluzione. Ora supponete che $(b_1 - b_2)^2 \leq b_1 + b_2$ ossia che i due conteggi siano "statisticamente vicini". Il primo pezzo di soluzione è a tutti gli effetti un numero random soggetto a forti fluttuazioni. Ma se ε è sufficientemente piccolo, esso può dominare rispetto al termine stabile! (Comprensibile, per $\varepsilon \rightarrow 0$ l'apparato è a tutti gli effetti incapace di distinguere i due bin)

ALCUNE CONSIDERAZIONI PRIMA DI CESTINARE IL METODO!

$$\mu_{ML} = R^{-1}(\nu - \beta) \text{ e' unbiased}$$

E la covarianza vale

$$V_{ij} = \text{cov}[\mu_{ML,i}, \mu_{ML,j}] = \sum_{k,l=1}^N (R^{-1})_{ik} (R^{-1})_{jl} \text{cov}[n_k, n_l]$$

$$= \sum_{k,l=1}^N (R^{-1})_{ik} (R^{-1})_{jl} \delta_{kl} \nu_k = \sum_{k=1}^N (R^{-1})_{ik} (R^{-1})_{jk} \nu_k$$

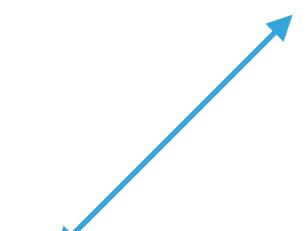
(se ricordate il plot, gli errori erano molto grandi)

ALCUNE CONSIDERAZIONI PRIMA DI CESTINARE IL METODO!

Cramér-Rao: stimatore unbiased → varianza ha un limite inferiore. Il nostro stimatore è unbiased, qual è il suo limite di Cramér-Rao?

$$-\mathbb{E} \left[\frac{\partial^2 \log \mathcal{L}}{\partial \mu_k \partial \mu_l} \right] = \sum_{i=1}^N \frac{R_{ik} R_{il}}{\nu_k}$$

Riguardiamo la varianza dello stimatore...

$$V_{ij} = \sum_{k=1}^N (R^{-1})_{ik} (R^{-1})_{jk} \nu_k$$


Questo stimatore è effettivamente quello di varianza minima (tra tutti gli estimatori unbiased).

Per avere degli errori "sensati" dovremo accettare un piccolo bias...

...e trovare il compromesso tra bias e varianza è la parte più importante di ogni procedura di unfolding.

CORRECTION FACTOR

Un primo passo avanti partendo dal metodo ML: definire $\mu_i = C_i(n_i - \beta_i)$ dove $C_i = \mu_i^{\text{MC}}/\nu_i^{\text{MC}}$

il fattore correttivo viene quindi dalla simulazione Montecarlo (senza considerare il background). Per questo stimatore la varianza è

$$V_{ij} = \text{cov}[\mu_i, \mu_j] = C_i C_j \text{cov}[n_i, n_j]$$

spesso C_i e' $\sim O(1)$, la varianza è quindi di molto ridotta rispetto al caso ML.

D'altro canto il bias è $b = \left(\frac{\mu_i^{\text{MC}}}{\nu_i^{\text{MC}}} - \frac{\mu_i}{\nu_i - \beta_i} \right) (\nu_i - \beta_i)$ ed è legato a quanto il MC è in grado di riprodurre la realtà

Consideriamo $C_i = 0.1; \beta_i = 0; n_i = 100; \rightarrow \mu_i = C_i n_i = 10; \sigma_{\mu_i} = C_i \sqrt{n_i} = 1$

e otteniamo un'incertezza del 10%. Com'e' possibile se solo 10 eventi portano veramente informazione?

CORRECTION FACTOR

Features:

- C dipende implicitamente dalla distribuzione che stiamo cercando di ricavare
- Le correlazioni tra i diversi bin sono completamente ignorate
- Il numero totale di eventi dopo la correzione potrebbe essere diverso dal numero di eventi totale prima della correzione

La riduzione in varianza è a scapito di un bias difficile da quantificare.

Una soluzione per ridurre il bias è quella di scegliere la larghezza dei bin di modo che sia più grande di un fattore 2-3 rispetto alla risoluzione attesa.

Utilizzato (nella variante iterativa) dall'esperimento AMS

REGULARISED UNFOLDING

Prendiamo $-2\log L$ come quantificatore della distanza tra i dati n e l'aspettazione ν

Consideriamo la soluzione ML, e muoviamoci un in intorno di essa, definito da un certo $\Delta \log L$. Di tutti gli stimatori nell'intorno scegliamo quello più "liscio", purché soddisfi $\log \mathcal{L}(\hat{\mu}) \geq \log \mathcal{L}_{\max} - \Delta \log \mathcal{L}$

Cerchiamo uno stimatore che massimizzi $\phi(\hat{\mu}) = \log \mathcal{L}(\hat{\mu}) + \tau S(\hat{\mu})$

Parametro di regolarizzazione Funzione di regolarizzazione

Se vogliamo anche che lo stimatore conservi il numero di eventi dobbiamo aggiungere un termine

$$\phi(\hat{\mu}) = \log \mathcal{L}(\hat{\mu}) + \tau S(\hat{\mu}) + \lambda \left(\sum_{i=1}^N n_i - \nu_i \right)$$

Cosa scegliamo per S ? Che valore di τ ?

REGULARISED UNFOLDING

Consideriamo

$$S[f(y)] = \int \left(\frac{d^k f}{dy^k} \right)^2 dy$$

che, per $k=2$, dà

$$S[\boldsymbol{\mu}] = \sum_{i=2}^{M-1} [(\mu_{i+1} - \mu_i) - (\mu_i - \mu_{i-1})]^2$$

(Scriviamo i vettori nella base della matrice di covarianza)

$$\log \mathcal{L} = -\frac{1}{2}\chi^2 = -\frac{1}{2}(R\boldsymbol{\mu} - \mathbf{y})^T V_y^{-1} (R\boldsymbol{\mu} - \mathbf{y}) = -\frac{1}{2}(R'\boldsymbol{\mu}' - \mathbf{y}')^T (R'\boldsymbol{\mu}' - \mathbf{y}')$$

Per cui l'oggetto che vogliamo minimizzare è

$$\phi(\hat{\boldsymbol{\mu}}) = (R'\hat{\boldsymbol{\mu}} - \mathbf{y}')^T (R'\hat{\boldsymbol{\mu}} - \mathbf{y}') + \tau(C\hat{\boldsymbol{\mu}})^T (C\hat{\boldsymbol{\mu}})$$

Il sistema diventa quindi

$$\begin{bmatrix} R' \\ \sqrt{\tau} \cdot C \end{bmatrix} \hat{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{y}' \\ \mathbf{0} \end{bmatrix}$$

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 1 & -2 & 1 \\ \cdots & \cdots & 1 & -1 & -1 \end{pmatrix}$$

REGULARISED UNFOLDING

$$\begin{bmatrix} R' \\ \sqrt{\tau} \cdot C \end{bmatrix} \hat{\mu} = \begin{bmatrix} \mathbf{y}' \\ \mathbf{0} \end{bmatrix} \Rightarrow \begin{bmatrix} R'C^{-1} \\ \sqrt{\tau} \cdot \mathbf{1} \end{bmatrix} \hat{\mu} = \begin{bmatrix} \mathbf{y}' \\ \mathbf{0} \end{bmatrix}$$

(migliora la stabilità del sistema)

Si parte dal caso tau=0, usiamo la SVD sulla matrice principale e risolviamo come abbiamo fatto prima

$$R'C^{-1} = USV^T$$

$$s_i z_i = d_i \Rightarrow \hat{\mu} = C^{-1} V \mathbf{z}$$

e riotteniamo la soluzione "standard", con tutti i suoi problemi. Rimettiamoci l'accendino...

$$d_i \rightarrow d_i \frac{s_i^2}{s_i^2 + \tau}, \quad z_i = \frac{d_i s_i}{s_i^2 + \tau}$$

La matrice di covarianza per z_i è facile da calcolare

$$Z_{ik} = \frac{s_i^2}{(s_i^2 + \tau)} \delta_{ik} \Rightarrow V_\mu = C^{-1} V Z V^T (C^{-1})^T$$

effettivamente i vari contributi alla soluzione provenienti dagli autovettori di R sono pesati da un termine che preferisce una "bassa curvatura"



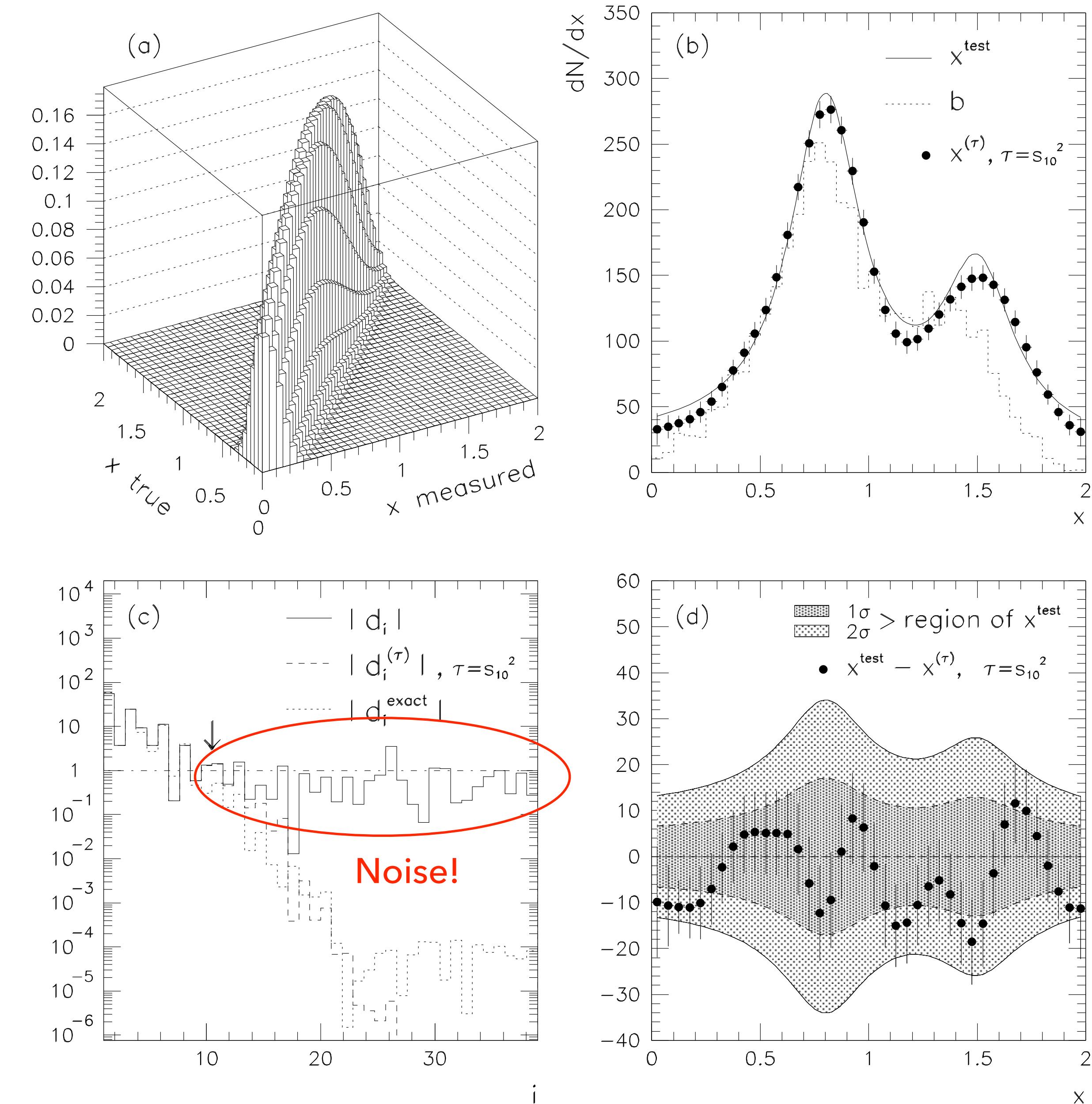
SVD REGULARISED

(tratto da [http://people.csail.mit.edu/hasinoff/320/
SingularValueDecomposition.pdf](http://people.csail.mit.edu/hasinoff/320/SingularValueDecomposition.pdf))

Figure 1: **a).** The probability matrix \hat{A} corresponding to the response function (58). **b).** The true distribution (59) (solid curve) compared to the measured histogram b and the unfolded distribution $x^{(\tau)}$ for $\tau = s_{10}^2$. **c).** The absolute values of d_i (solid line) compared to the regularized r.h.s. (dashed line) and the one unaffected by the statistical fluctuations (dotted line). The horizontal line shows statistical errors in d_i , while the arrow indicates the boundary between the significant and non-significant equations. **d).** The deviation of the unfolded distribution from the true exact one (see text for details).

Dal plot dei d_i è visibile a occhio quali autovettori di R sono quelli responsabili delle fluttuazioni e del rumore nella soluzione "standard"!

La regolarizzazione sopprime naturalmente il contributo di queste componenti "rumorose"



BAYESIAN UNFOLDING

(<https://xkcd.com/2059/>)

MODIFIED BAYES' THEOREM:

$$P(H|x) = P(H) \times \left(1 + P(C) \times \left(\frac{P(x|H)}{P(x)} - 1 \right) \right)$$

H: HYPOTHESIS

x: OBSERVATION

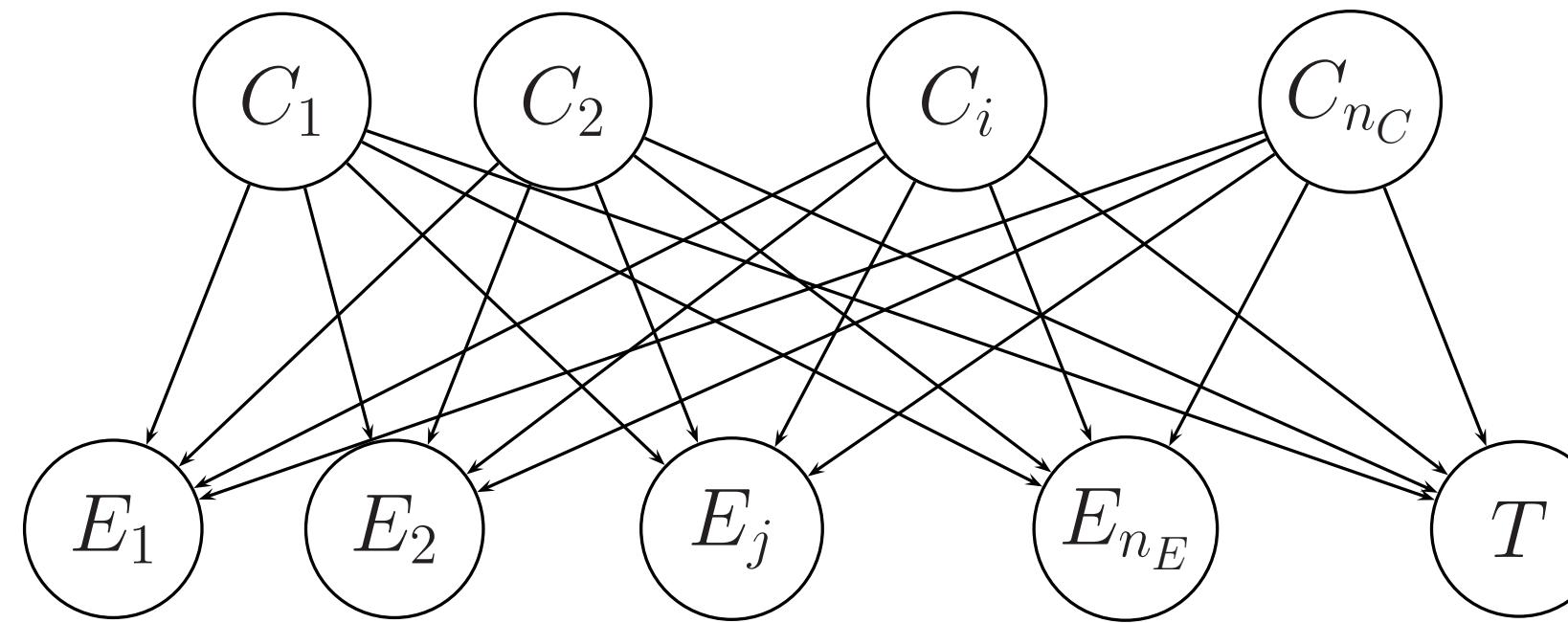
P(H): PRIOR PROBABILITY THAT H IS TRUE

P(x): PRIOR PROBABILITY OF OBSERVING x

P(C): PROBABILITY THAT YOU'RE USING
BAYESIAN STATISTICS CORRECTLY

BAYESIAN UNFOLDING

Un altro modo di visualizzare la matrice di risposta è come una mappa tra una serie di "cause" e un insieme di "effetti".



In questo caso possiamo sfruttare il teorema di Bayes

$$\text{posterior} \quad \text{matrice di risposta} \quad \text{prior}$$
$$P(C_i | E_j) = \frac{P(E_j | C_i) \cdot P(C_i)}{\sum_k P(E_j | C_k) \cdot P(C_k)}$$

BAYESIAN UNFOLDING

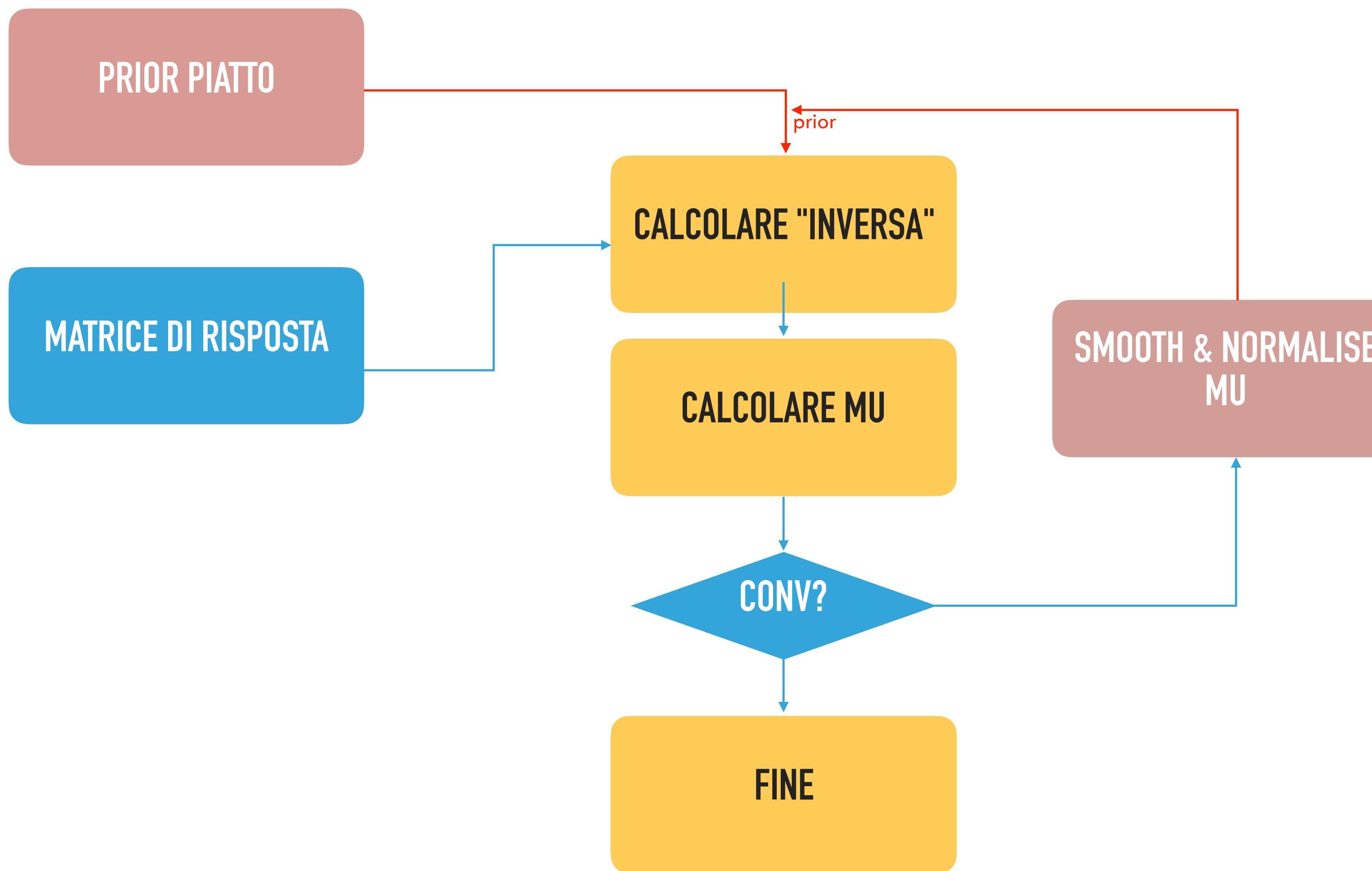
Tornando alla nostra notazione...

$$\mu_i = \frac{1}{\varepsilon_i} \sum_{j=1}^N \mathbb{P}(C_i|E_j)(n_j - \beta_j) = \frac{1}{\varepsilon_i} \sum_{j=1}^N \theta_{ij}(n_j - \beta_j)$$

$$\varepsilon_i = \sum_{j=1}^N \mathbb{P}(E_j|C_i) = \sum_{j=1}^N \lambda_{ij}$$

$$\lambda_{ij} = \left. \frac{\nu_j}{\mu_i} \right|_{\text{MC}} \quad \xrightarrow{\hspace{10cm}} \quad \theta_{ij} = \frac{\lambda_{ji} \mathbb{P}(C_i)}{\sum_k \lambda_{jk} \mathbb{P}(C_k)}$$

BAYESIAN UNFOLDING



BAYESIAN UNFOLDING: ERROR ESTIMATION

La versione originale del metodo basava la stima degli errori su molte approssimazioni + propagazione della varianza.

Salvo poi essere rivisitato più tardi. Partiamo dalla matrice di risposta

$$\boldsymbol{\lambda}_i = (\lambda_{i1}, \dots, \lambda_{iN}) \sim \text{Dir} [\boldsymbol{\alpha}_{\text{prior}} + \boldsymbol{\nu}|_{\mu_i}^{\text{MC}}]$$

dal momento che è ottenuta da un MC, ha la sua incertezza statistica e le sue fluttuazioni. Simuliamone diverse realizzazioni tramite un toy MC basato sulla distribuzione duale (Dirichlet).

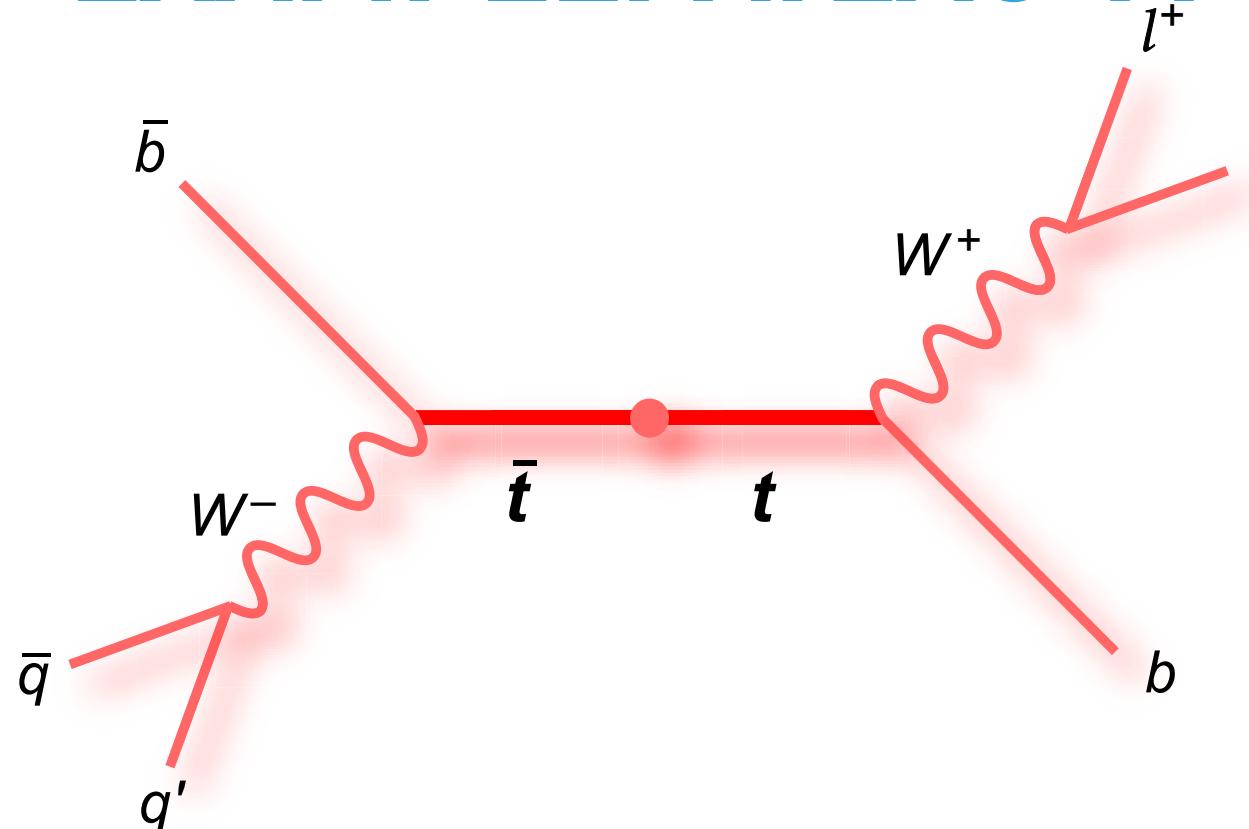
Per ogni differente realizzazione della matrice calcolare la matrice "inversa" $\boldsymbol{\theta}_i = (\theta_{i1}, \dots, \theta_{iN})$

Gli eventi vengono poi ripartiti nelle celle "causa" tramite una multinomiale governata dalla matrice "inversa"

$$\boldsymbol{\mu}|_{\nu_j} \sim \text{Mult} [\nu_j, \boldsymbol{\theta}_j] \quad \Rightarrow \quad \boldsymbol{\mu} = \sum_{j=1}^N \boldsymbol{\mu}|_{\nu_j}$$

Rinse and repeat

EXAMPLE: ATLAS $t\bar{t}$ CHARGE ASYMMETRY

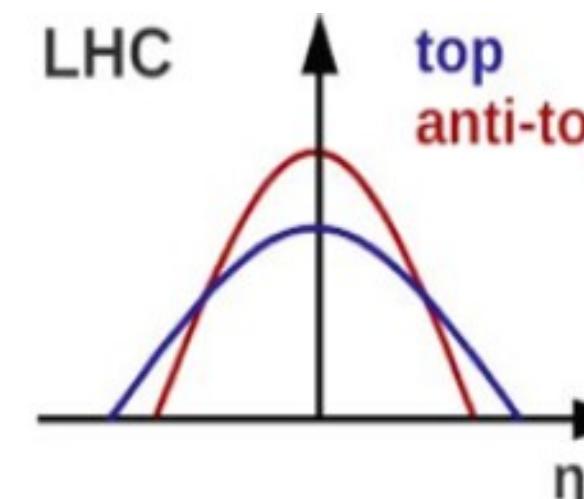


with ATLAS @ LHC

$\int L dt = 1 \text{ fb}^{-1}$ (2011)

- Stop when A_C changes by less than 0.1% on MC
- Stat uncertainty checked with pseudoexperiments
- Syst uncertainty propagated to response matrix and bkg
- Re-weight $t\bar{t}$ events to vary A_C and check unfolding linearity.

- Expect

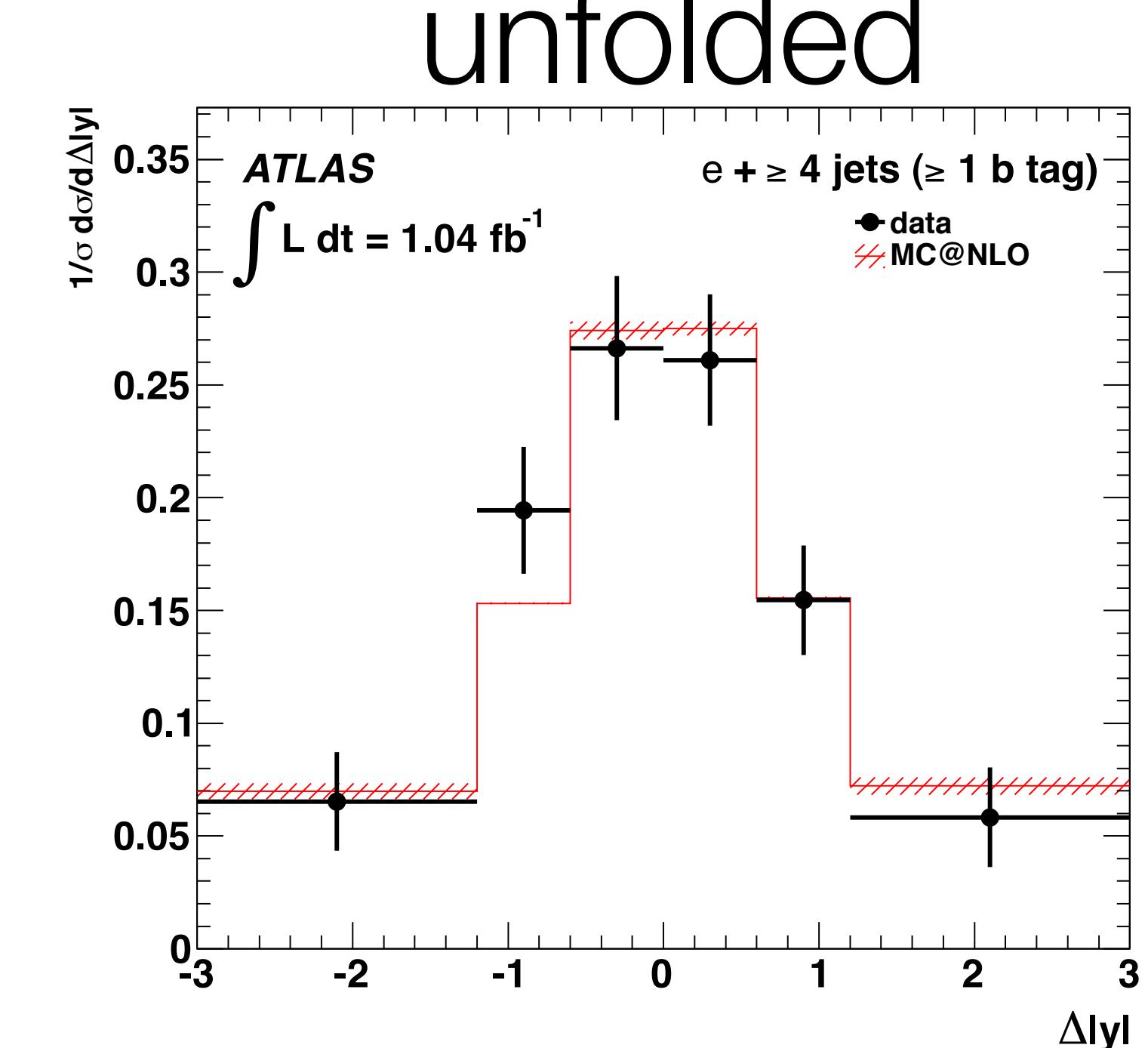
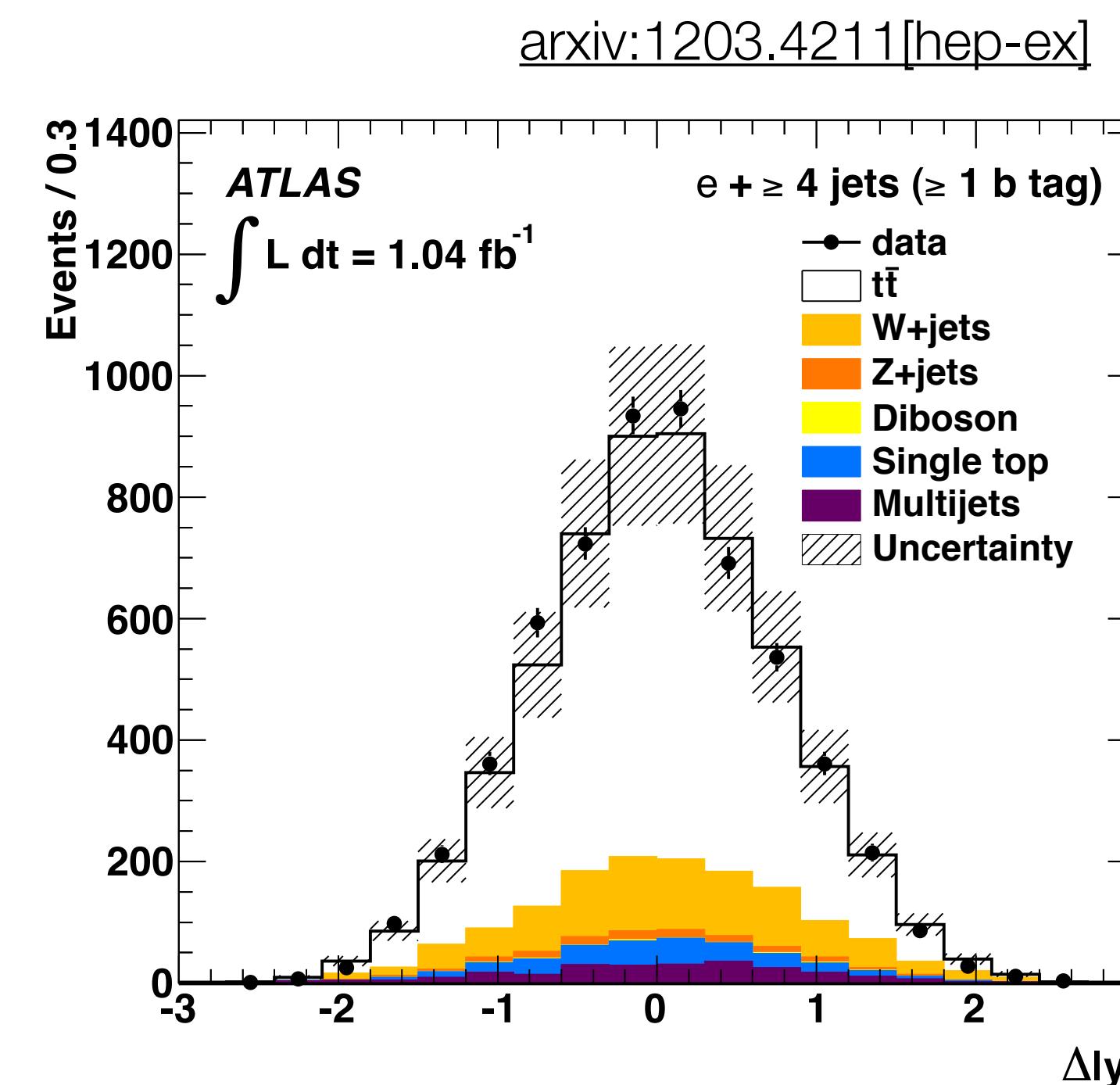


MC@NLO@ 7TeV LHC predicts $A_C =$

$$0.006 \pm 0.002$$

- Reconstruct $t\bar{t}$ and study

$$A_C = \frac{N(\Delta|Y| > 0) - N(\Delta|Y| < 0)}{N(\Delta|Y| > 0) + N(\Delta|Y| < 0)}$$



ITERATIVE CORRECTION FACTOR

Utilizzato da AMS-02 per la misura di flussi di nuclei

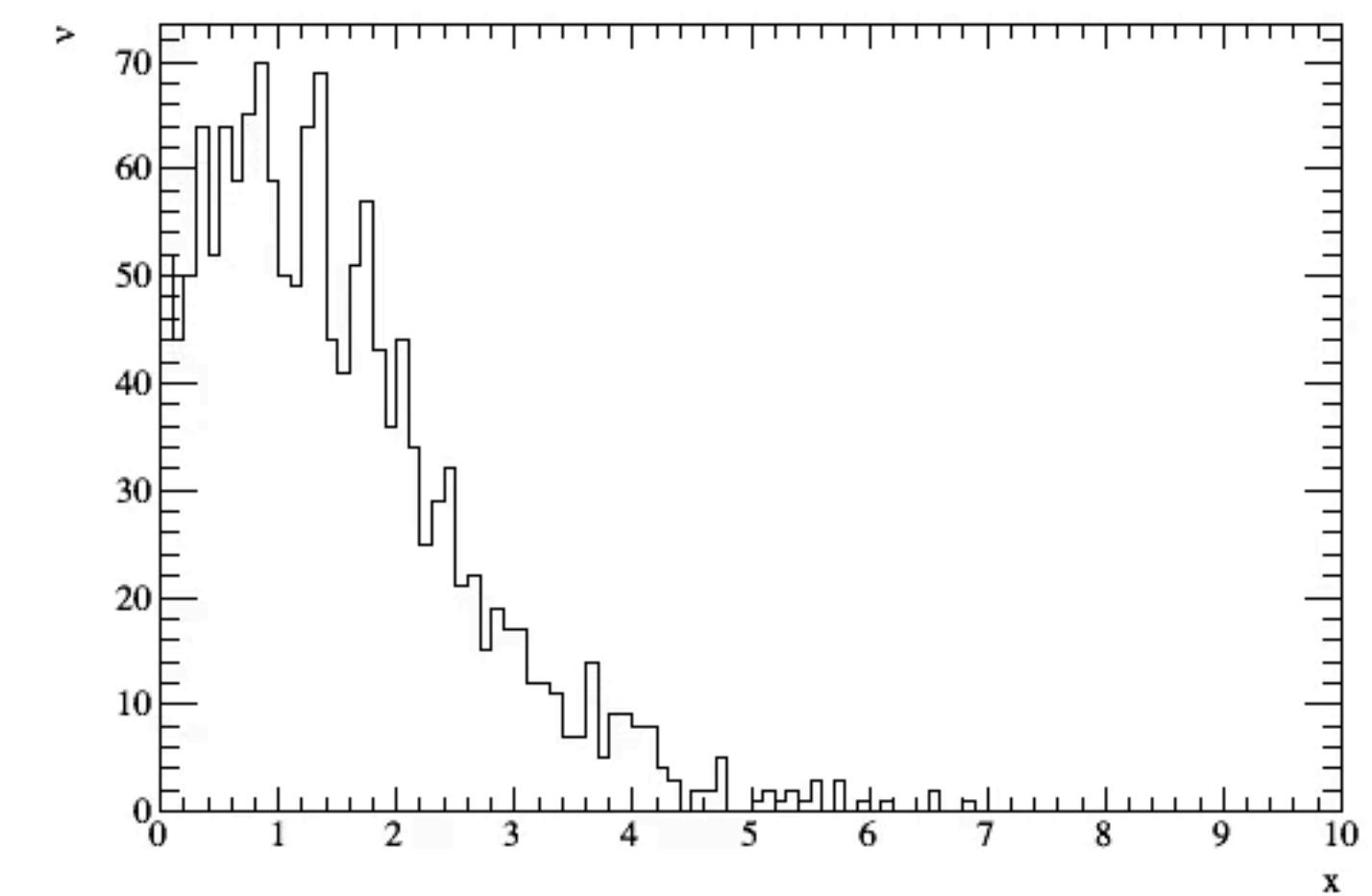
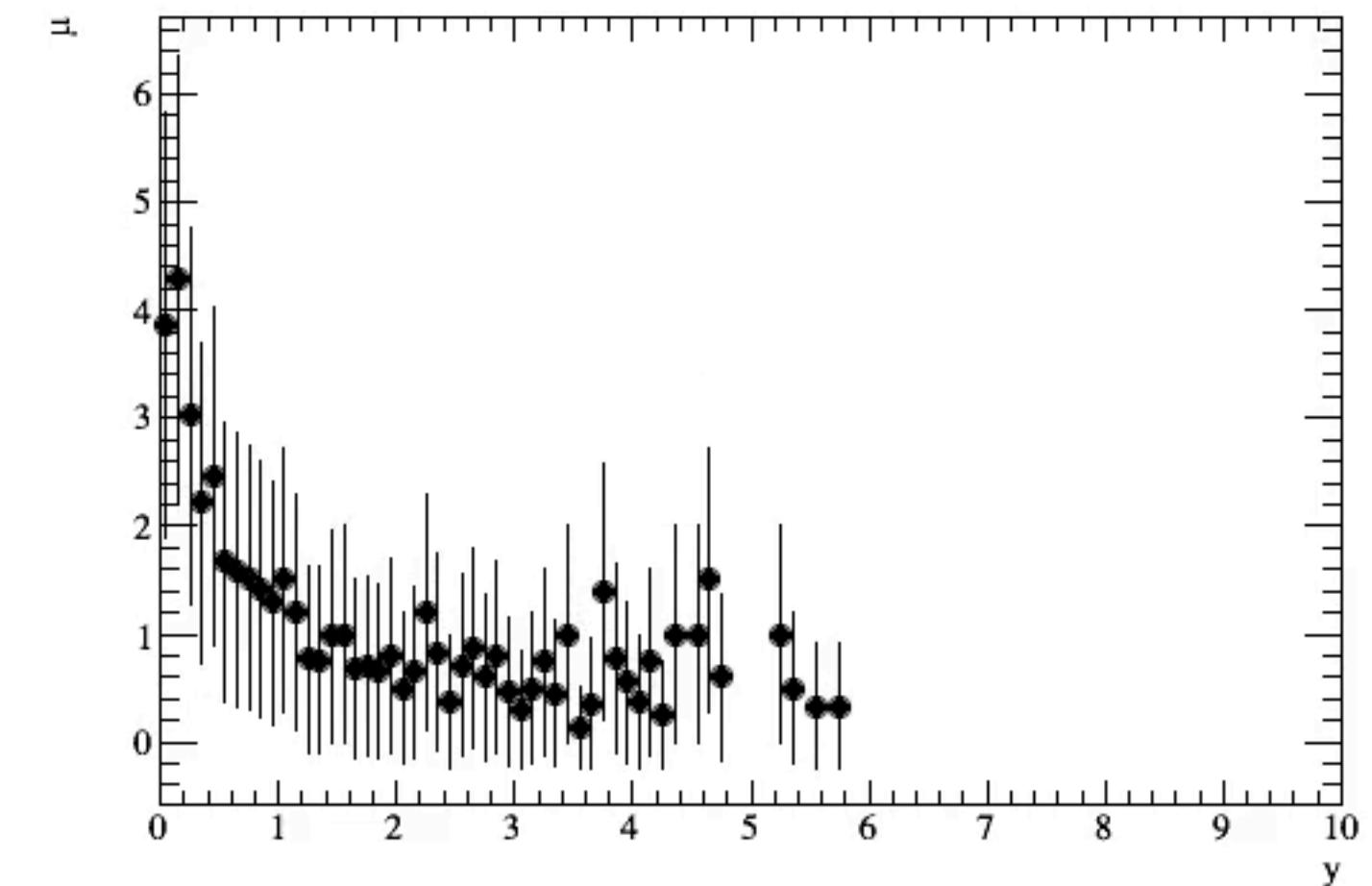
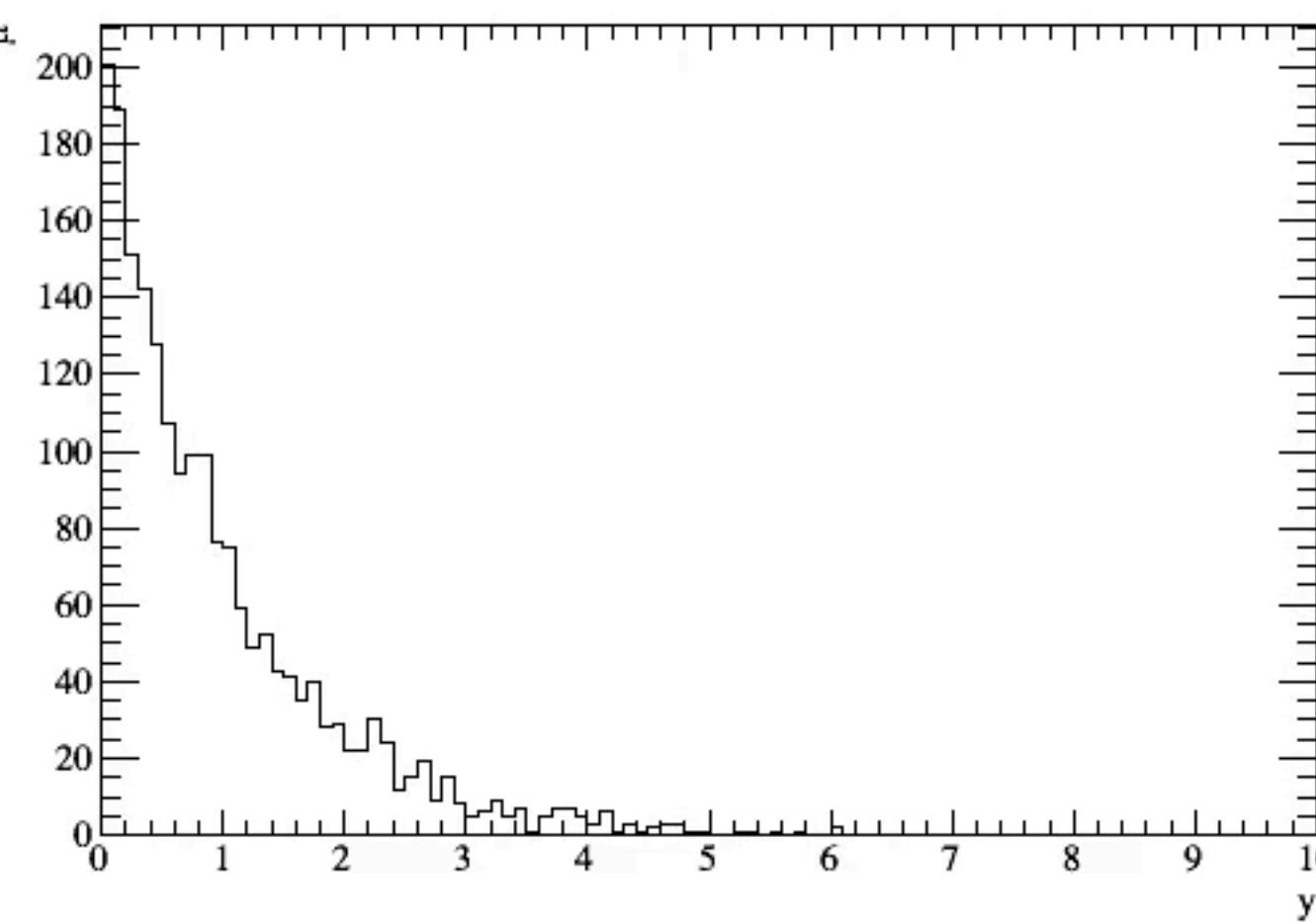
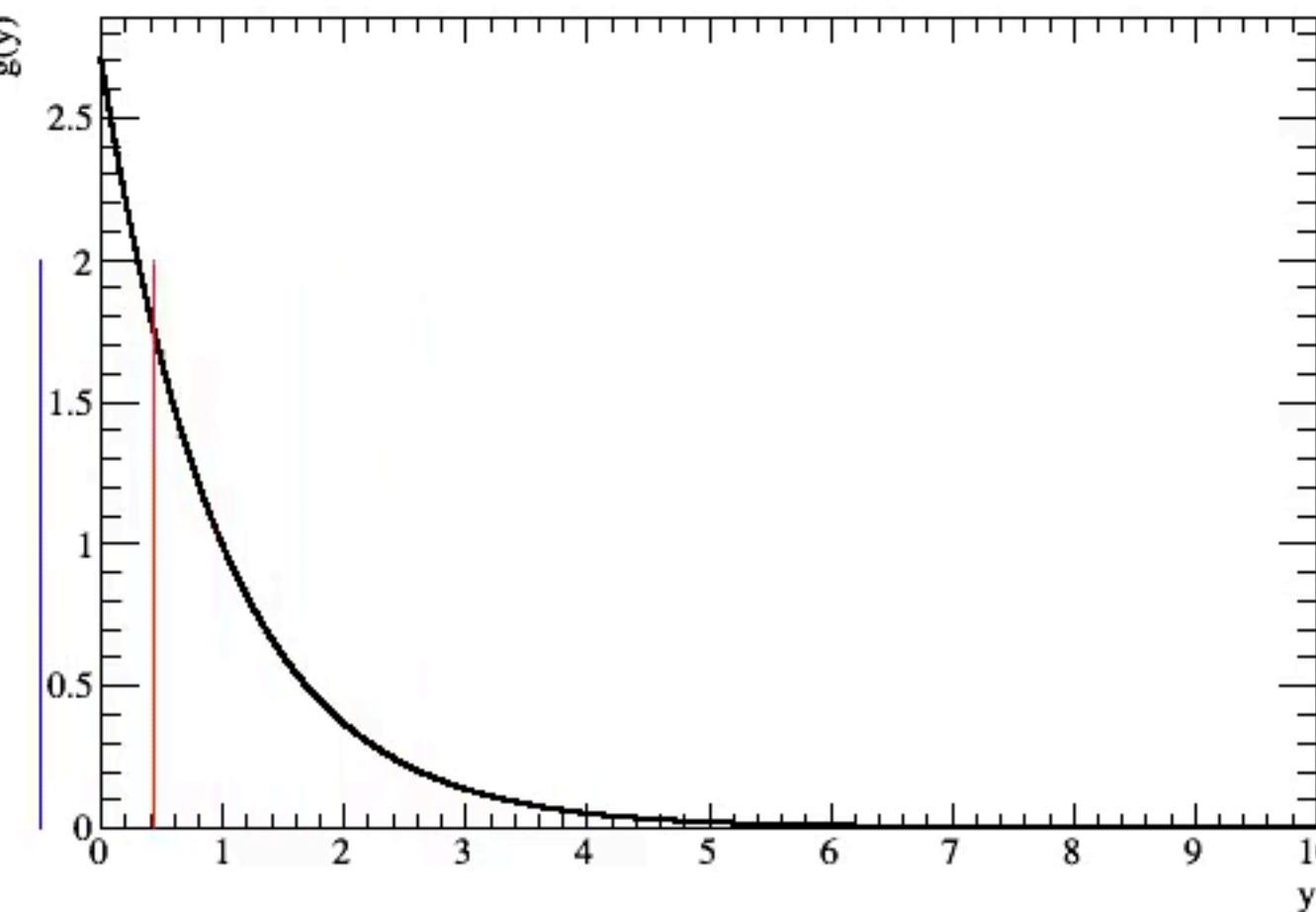
$$\mu_i = C_i(n_i - \beta_i)$$

Partiamo da una distribuzione $g_0(y)$ iniziale. Simuliamo eventi e costruiamo istogrammi in funzione della variabile vera e di quella misurata.

$$C_i = \mu_i^{\text{MC}} / \nu_i^{\text{MC}}$$

$$\mu_i^{\text{MC}} = \sum_{j=1}^{N_{\text{ev}}} g_0(y_j) \Pi \left(\frac{y_i - \tilde{y}_i}{\Delta y_i} \right)$$

$$\nu_i^{\text{MC}} = \sum_{j=1}^{N_{\text{ev}}} g_0(y_j) \Pi \left(\frac{x_i - \tilde{x}_i}{\Delta x_i} \right)$$



ITERATIVE CORRECTION FACTOR

$$\mu_i^0 = C_i^0(n_i - \beta_i) \quad , \quad C_i^0 = \mu_i^{\text{MC}} / \nu_i^{\text{MC}}$$

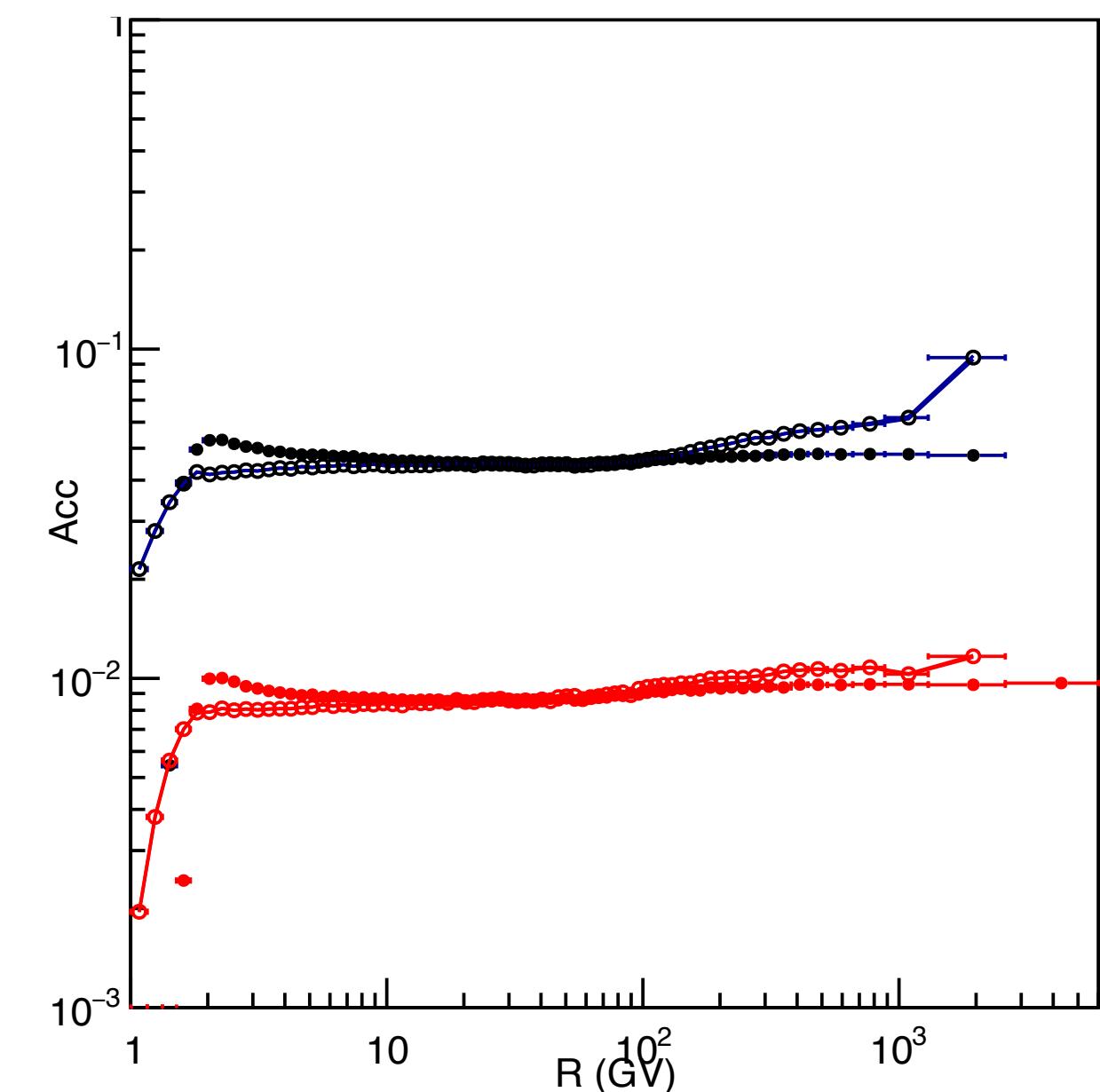
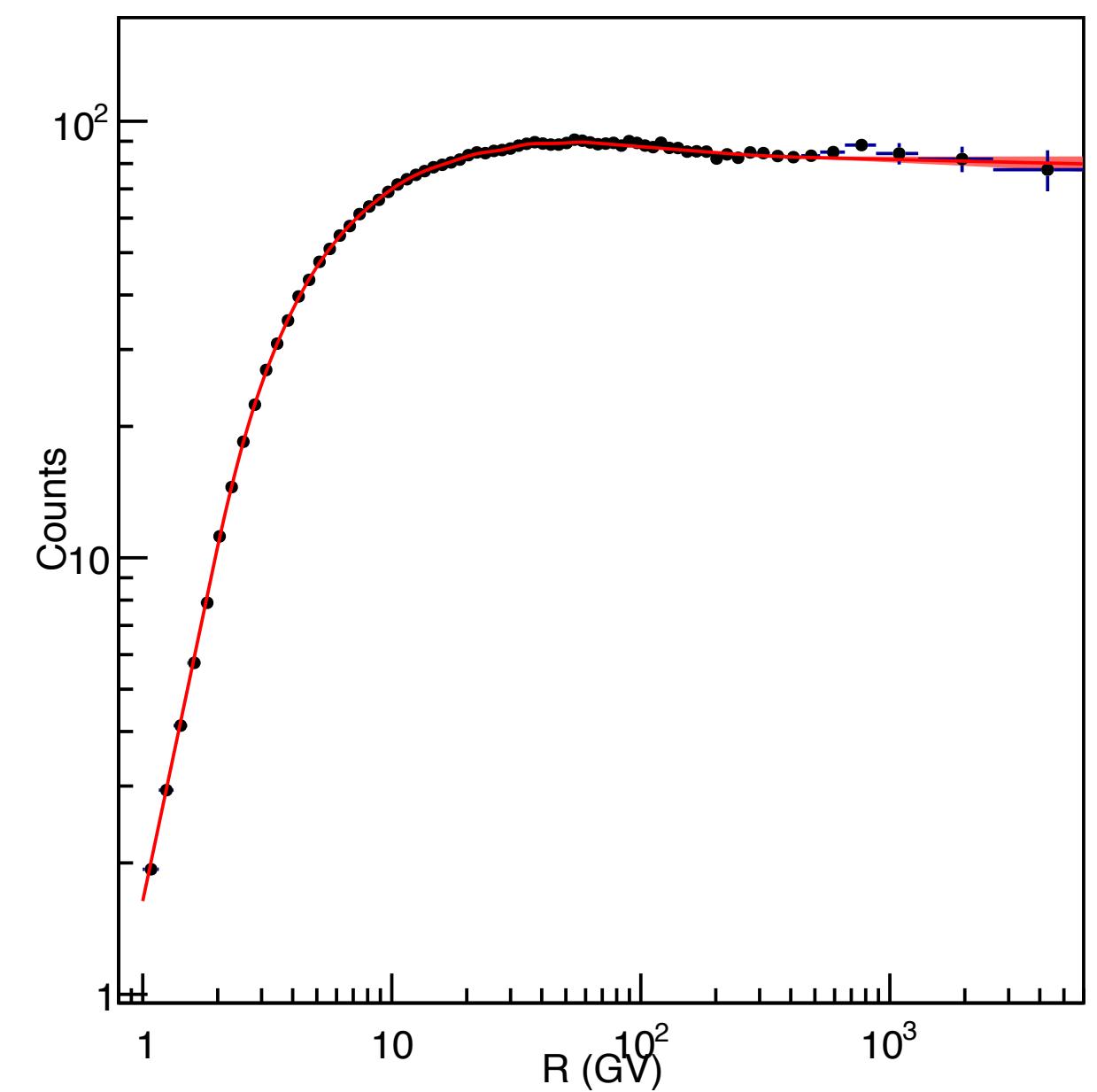
Definiamo una nuova funzione di peso

$$g_1(y) = f(\tilde{\mu}) \quad , \quad \tilde{\mu}_i = \frac{\mu_i}{\sum_{j=1}^N \mu_j}$$

tale che sia una approssimazione sufficientemente liscia del risultato al passo 0. Usiamola per ricalcolare il fattore correttivo.

$$\mu_i^{\text{MC}} = \sum_{j=1}^{N_{\text{ev}}} g_1(y_j) \Pi \left(\frac{y_j - \tilde{y}_j}{\Delta y_j} \right) \quad , \quad \nu_i^{\text{MC}} = \sum_{j=1}^{N_{\text{ev}}} g_1(y_j) \Pi \left(\frac{x_j - \tilde{x}_j}{\Delta x_j} \right)$$

Ripetere la procedura finché la differenza tra due iterazioni successive diventa arbitrariamente piccola.



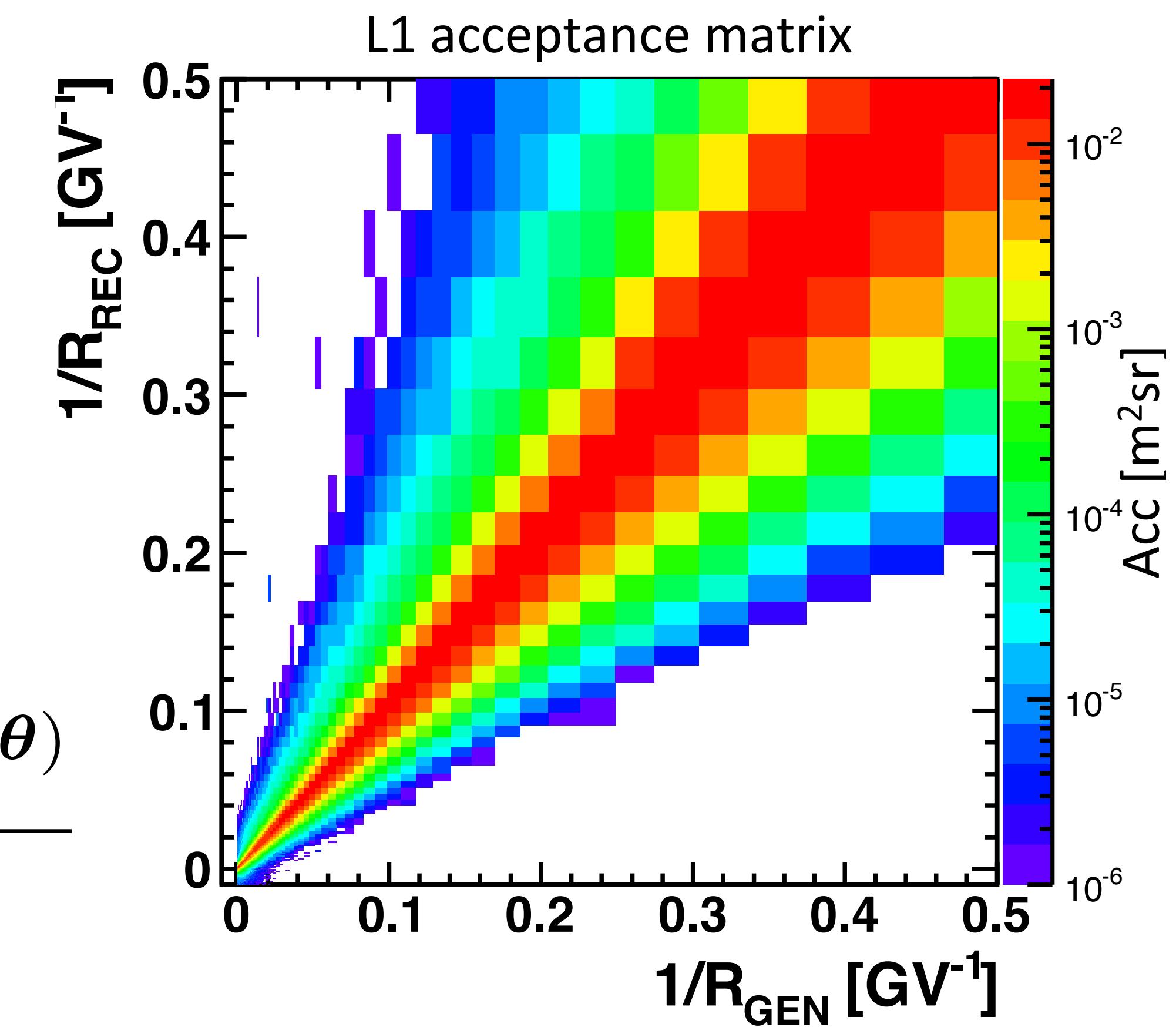
FORWARD FOLDING

Se avete disponibile un ansatz sulla distribuzione che state cercando potete cercare di massimizzare una likelihood

$$\hat{\mu}_i(\theta) = \int_{y_i}^{y_{i+1}} f(y; \theta) dy$$

$$\hat{\nu}_i(\theta) = \sum_{j=1}^N R_{ij} \hat{\mu}_j(\theta)$$

$$\mathcal{L}(\theta | \mathbf{n}) = \prod_{i=1}^N \mathbb{P}(n_i | \hat{\nu}_i(\theta)) = \prod_{i=1}^N \hat{\nu}_i(\theta)^{n_i} \frac{e^{-\hat{\nu}_i(\theta)}}{n_i!}$$



La differenza col metodo iniziale sta nel fatto che la $f(y)$ ora è smooth per costruzione!

FORWARD FOLDING

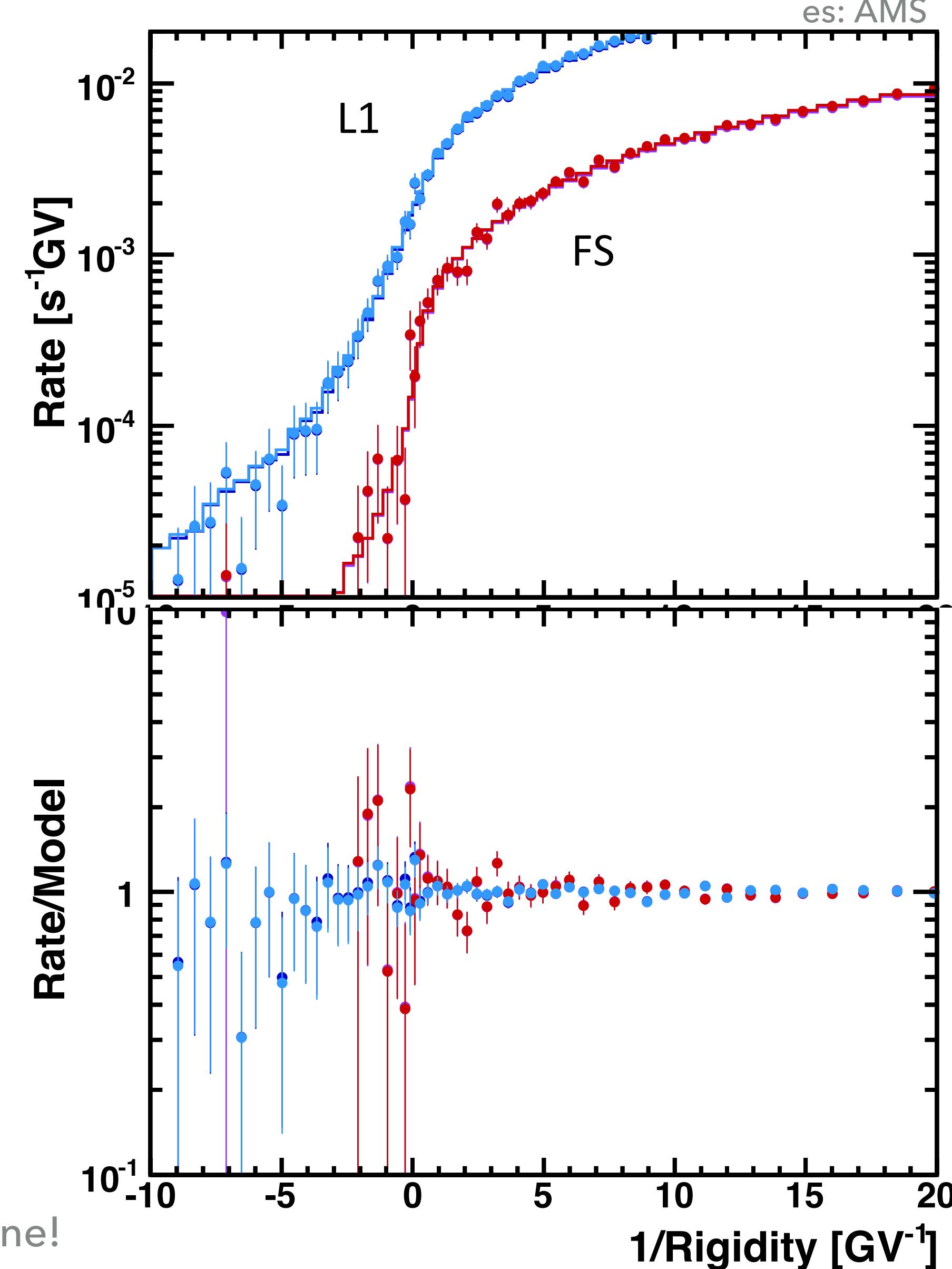
Se avete disponibile un ansatz sulla distribuzione che state cercando potete cercare di massimizzare una likelihood

$$\hat{\mu}_i(\theta) = \int_{y_i}^{y_{i+1}} f(y; \theta) dy$$

$$\hat{\nu}_i(\theta) = \sum_{j=1}^N R_{ij} \hat{\mu}_j(\theta)$$

$$\mathcal{L}(\theta | \mathbf{n}) = \prod_{i=1}^N \mathbb{P}(n_i | \hat{\nu}_i(\theta)) = \prod_{i=1}^N \hat{\nu}_i(\theta)^{n_i} \frac{e^{-\hat{\nu}_i(\theta)}}{n_i!}$$

La differenza col metodo iniziale sta nel fatto che la $f(y)$ ora è smooth per costruzione!

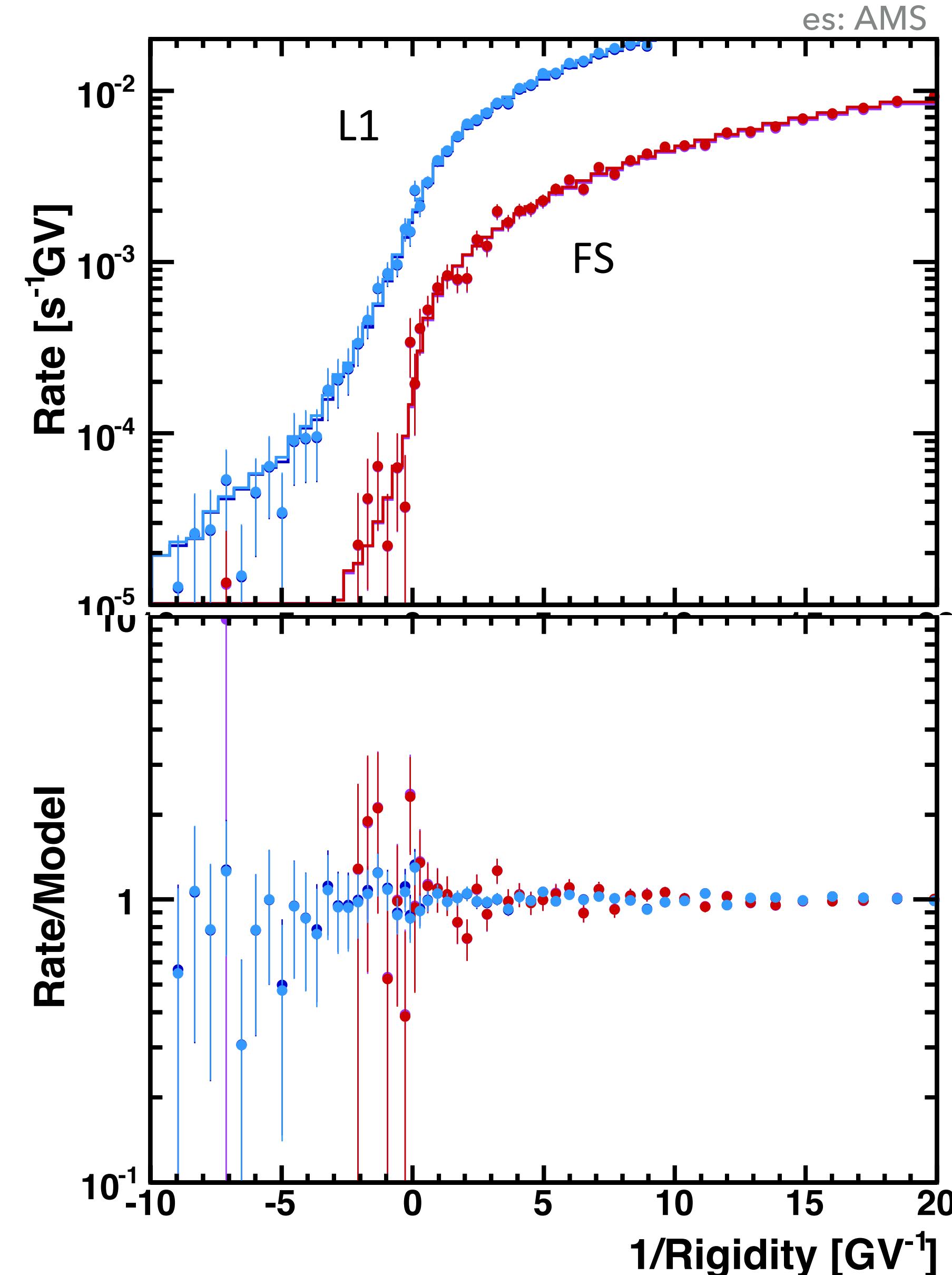


FORWARD FOLDING

$$\mathcal{L}(\theta|\mathbf{n}) = \prod_{i=1}^N \mathbb{P}(n_i | \hat{\nu}_i(\theta)) = \prod_{i=1}^N \hat{\nu}_i(\theta)^{n_i} \frac{e^{-\hat{\nu}_i(\theta)}}{n_i!}$$

Ovviamente ci sono dei caveat:

- Funziona solo se avete a che fare con distribuzioni lisce in partenza.
- Se potete costruire una funzione liscia il cui integrale rappresenta direttamente i conteggi siete a cavallo. Altrimenti in genere qualsiasi trasformazione della p.d.f. rompe l'assunzione di aspettazione poissoniana
(es: in AMS si usa il rate di conteggi come funzione di input. Il rate è calcolato dividendo i conteggi per la funzione di trasmissione geomagnetica -> l'integrale del rate all'interno di un bin non è più una poissoniana!)
- Difficile stimare le incertezze sul risultato
(toy MC, propagazione degli errori; ma non è banale come problema)
- Può essere ostico far convergere il fit.



QUALCHE ALTRO ESEMPIO...

- ▶ IDS: iterative dynamically stabilized, B. Malaescu, arxiv: 0907.3791 [phys.data-an]
 - ▶ used in ATLAS paper <http://arxiv.org/abs/1112.6297>
- ▶ Binning free Iterative Deconvolution, Lindemann, Zech, Nucl.Instr. Meth A 354 (1995) 516-521
- ▶ Satellite Method, see G. Bohm and G. Zech, Introduction to Statistics and Data Analysis for Physicists, Verlag Deutsches Elektronen-Synchrotron (2010), available at <http://www-library.desy.de/elbook.html>
- ▶ SPlot, M Pivk, F. Le Diberder, arXiv:physics/0402083v

Domani lezione pratica! Bring your own laptop! (data and slides: https://github.com/valerioformato/Lesson_Unfolding)