Statistical Methods in Particle Physics

10. Unfolding

Heidelberg University, WS 2020/21

Klaus Reygers (lectures)
Rainer Stamen, Martin Völkl (tutorials)

Deconvolution

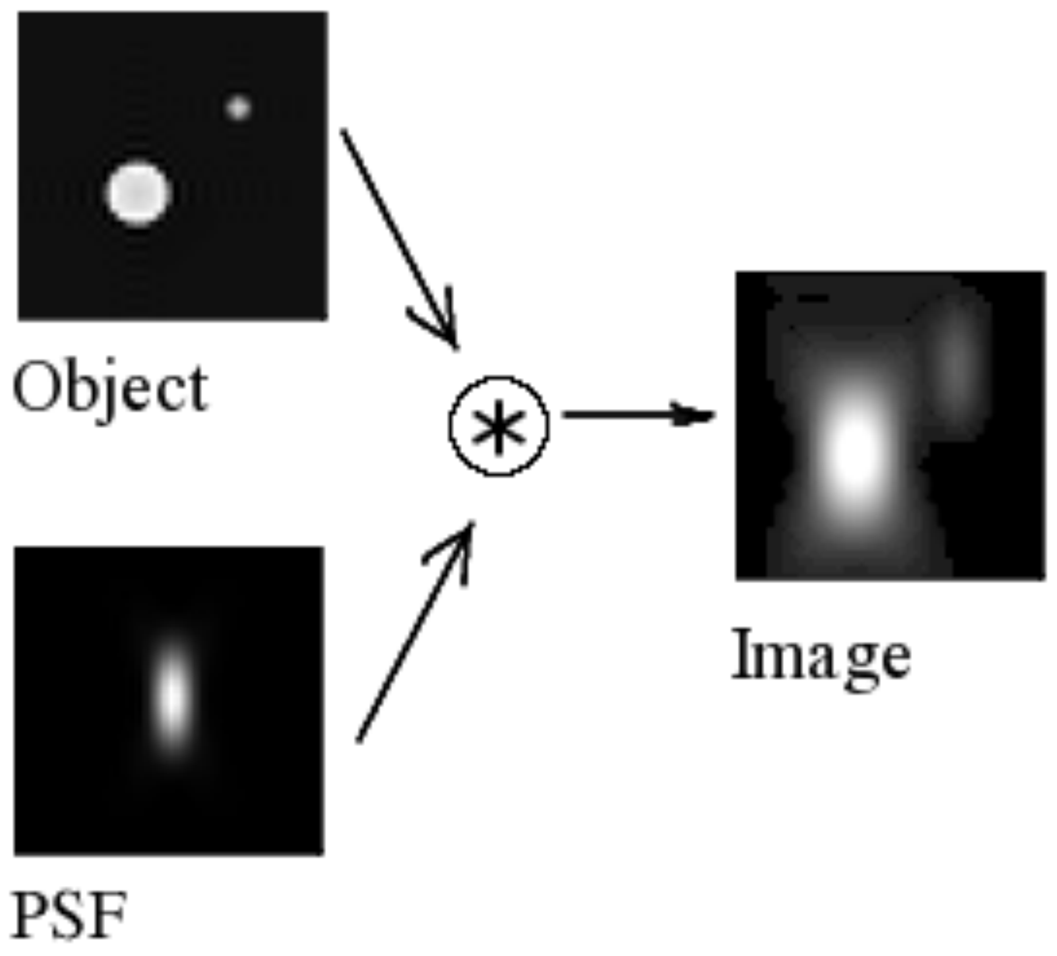
Finite resolution of the detector smears the quantities we're interested in.

Goal:
smeared information
→ original information

This is called *deconvolution* or *unfolding*

"Inverse problem"

Problem can be ill-posed in the sense that unfolded result can be very sensitive to small perturbations in the data



Example:
Smearing of a telescope image

https://en.wikipedia.org/wiki/Point_spread_function

To unfold or not to unfold?

[C. Pruneau,
Data Analysis Techniques for Physical Scientists]

From S. Oser's lecture:

> The most important advice I can give about deconvolution is "Don't".
>
> It's a lot of work, and often produces biased or otherwise unsatisfactory results. Moreover it's often unnecessary.

"Forward fitting" is much easier

- Take theory prediction
- Convolve it with the response of the detector
- Compare smeared theory directly with the data

When unfolding makes sense

1. Results from experiment A and B with different response function are to be compared

2. It is too complicated to publish the response function of the detector along with the data

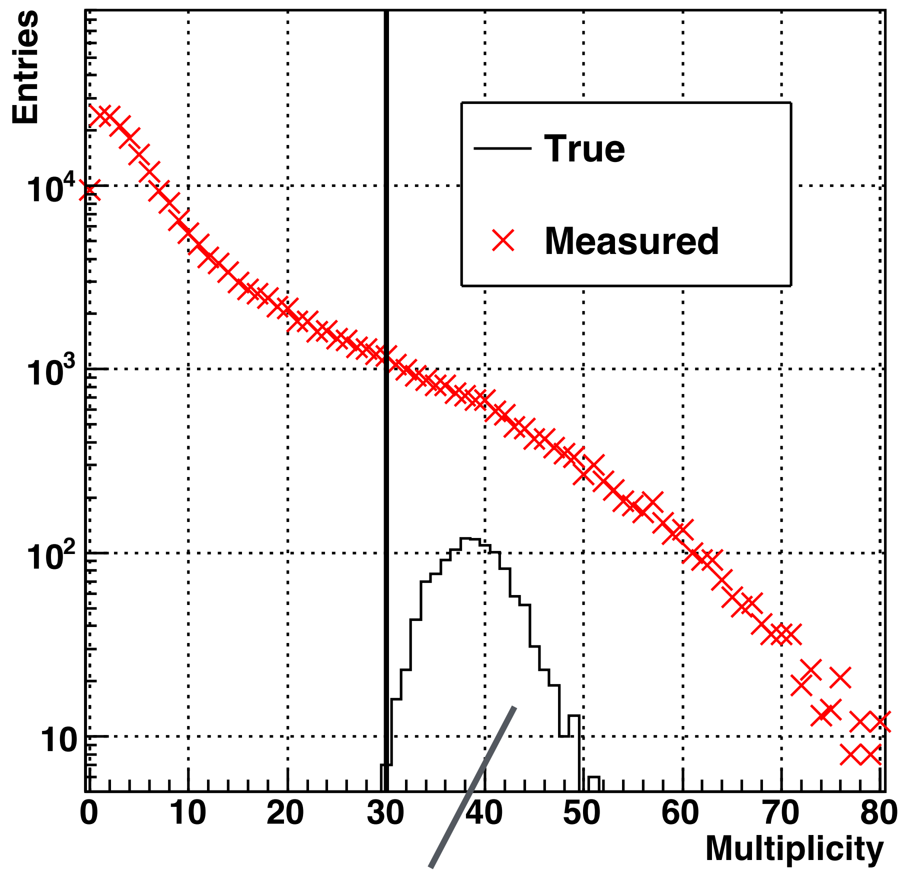
   - Detector response might be very complex, e.g., time dependent
   - Sometimes computer code reflecting the response would have to be published
   - Danger that future users don't use the filter correctly

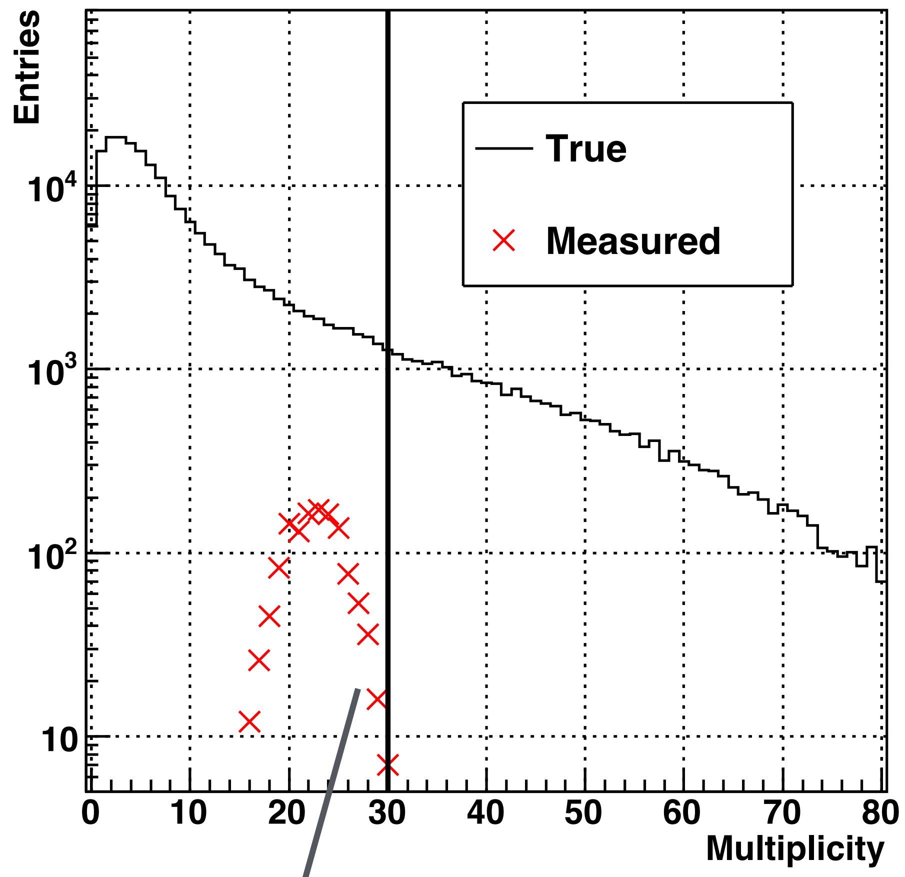
Examples

- Multiplicity distributions $P(N_{ch})$
  - Measured multiplicity differs from true charged particle multiplicity due to detector effects (efficiency, fake hits, …)
- p_T spectra, e.g., π^0 spectrum measured with a calorimeter
  - finite energy resolution and shower overlaps in a calorimeter affect the p_T of the reconstructed shower

Example: multiplicity distributions in pp collisions

arXiv:0912.0023





true N's contributing to measured N = 30

measured N's for a true N = 30

Response matrix (1)

Suppose we deal with continues variables (e.g., transverse momentum)

$f_t(x_t)$	:	distribution of true values (normalized to unity)
$f_m(x_m)$	:	distribution of measured values (normalized to unity)
$f_b(x_m)$	:	distribution of background (normalized to unity)

Response function *R*:

$$\underbrace{R(x_m|x_t) = r(x_m|x_t)}_{\text{"smearing"}} \times \underbrace{\varepsilon(x_t)}_{\text{"efficiency"}}$$

probability (density) to observe x_m given x_t

By construction, one has

$$\int_{\Omega_m} r(x_m|x_t)\,\mathrm{d}x_m = 1$$

Response Matrix (2)

Further definitions:

m_tot	:	total number of true events
n_tot	:	total number of measured events
b_tot	:	total number of background events

$$\mu_\text{tot} = E[m_\text{tot}], \quad \nu_\text{tot} = E[n_\text{tot}], \quad \beta_\text{tot} = E[b_\text{tot}]$$

It is practical to work with discrete bins. E.g., probability to find x_t in bin j:

$$p_j = \int_{\text{bin } j} \mathrm{d}x_t \, f_t(x_t), \quad \mu_j = \mu_\text{tot} \times p_j$$

Ignoring backgrounds, the expected measured number of entries in bin i is:

$$\begin{aligned} \nu_i &= \mu_\text{tot} \int_{\Omega_t} \mathrm{d}x_t \, \text{Prob}(x_m \text{ in } i | \text{true } x_t, \text{ detected}) \\ &\quad \times \text{Prob}(\text{detect } x_t) \times \text{Prob}(\text{produce } x_t) \\ &= \mu_\text{tot} \int_{\text{bin } i} \mathrm{d}x_m \int_{\Omega_t} \mathrm{d}x_t \, r(x_m|x_t)\varepsilon(x_t) f_t(x_t) \end{aligned}$$

Response Matrix (3)

Further definitions:

$$\int_{\Omega_t} \mathrm{d}x_t = \sum_{j=1}^{M} \int_{\text{bin } j} \mathrm{d}x_t$$

$$\nu_i = \mu_{\text{tot}} \int_{\text{bin } i} \mathrm{d}x_m \sum_{j=1}^{M} \int_{\text{bin } j} \mathrm{d}x_t \, r(x_m|x_t)\varepsilon(x_t)f_t(x_t)$$

$$= \sum_{j=1}^{M} \int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \, \frac{r(x_m|x_t)\varepsilon(x_t)f_t(x_t)}{\mu_j/\mu_{\text{tot}}} \mu_j$$

This may be written as

$$\nu_i = \sum_{j=1}^{M} R_{ij}\mu_j$$

with the components of the response matrix R_{ij} given by

$$R_{ij} = \frac{\int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t \, r(x_m|x_t)\varepsilon(x_t)f_t(x_t)}{\int_{\text{bin } j} \mathrm{d}x_t f(x_t)}$$

Response matrix (4)

In other words:

$$R_{ij} = \mathrm{Prob}(\text{observed in bin } i | \text{true in bin } j)$$

Obviously, summing the response matrix over *i* gives the efficiency:

$$\sum_{i=1}^{N} R_{ij} = \varepsilon_j$$

In compact matrix form (including background):

$$\nu_i = \sum_{j=1}^{M} R_{ij}\mu_j + \beta_i \qquad\qquad \vec{\nu} = R\vec{\mu} + \vec{\beta}$$

Response matrix depends on $f_t(x_t)$ which we want to know. However, if we make the bins small enough $f_t(x_t) \approx$ const. within a bin and drops from the ratio:

$$R_{ij} = \frac{\int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t\, r(x_m|x_t)\varepsilon(x_t) f_t(x_t)}{\int_{\text{bin } j} \mathrm{d}x_t f(x_t)} \approx \frac{1}{\Delta x_{t,j}} \int_{\text{bin } i} dx_m \int_{\text{bin } j} \mathrm{d}x_t\, r(x_m|x_t)\varepsilon(x_t)$$

Unfolding by inverting the response matrix (1)

We have

$$\vec{\nu} = R\vec{\mu} + \vec{\beta}$$

Replace $\vec{\nu}$ by $\vec{n}$ to obtain and obvious estimator for the true distribution:

$$\hat{\vec{\mu}} = R^{-1}(\vec{n} - \vec{\beta})$$

This solution minimizes

$$\chi^2(\vec{\mu}) = (\vec{\nu}(\vec{\mu}) - \vec{n})^{\mathsf{T}} V^{-1} (\vec{\nu}(\vec{\mu}) - \vec{n}) \qquad \text{where} \qquad V_{i,j} = \mathrm{cov}[n_i, n_j]$$

It can be shown that the covariance matrix of the solution is given by

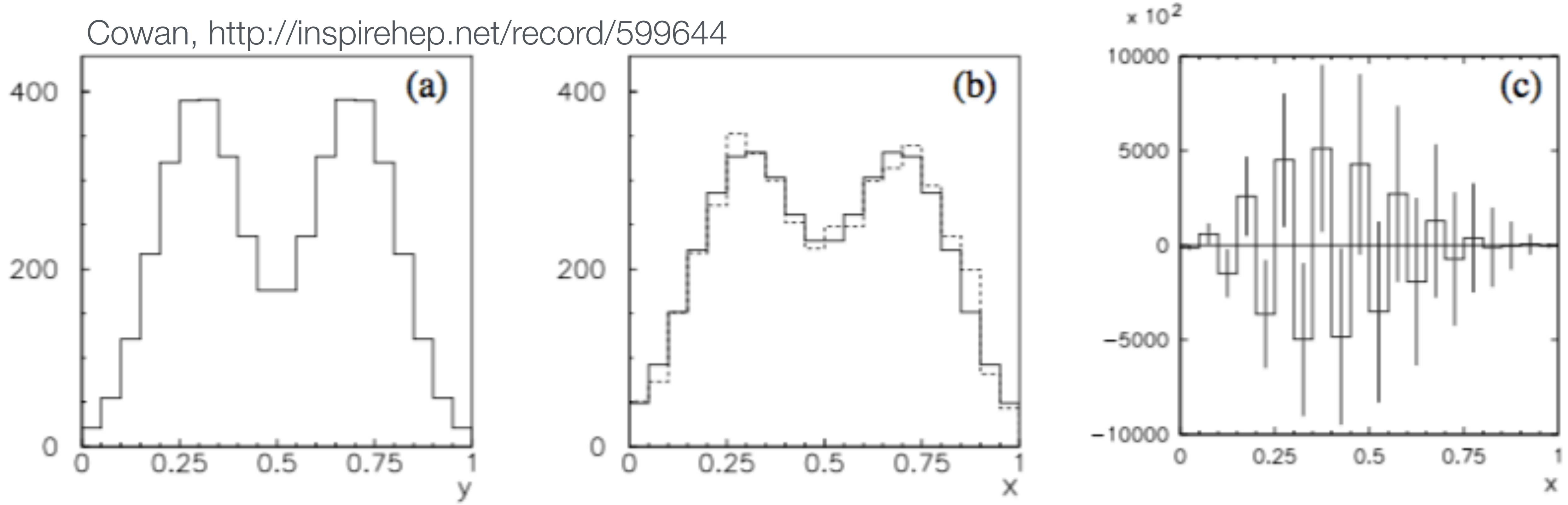
$$U = R^{-1} V (R^{-1})^{\mathsf{T}}$$

Unfolding by inverting the response matrix (2)

It can also be shown that matrix inversion is unbiased an has minimal variance.

This sounds good … let's try it.

Cowan, http://inspirehep.net/record/599644



true distribution $\vec{\mu}$

solid line: $\vec{\nu} = R\vec{\mu}$

dashed line: sampled data $\vec{n}$ — from Poisson(n_i, ν_i)

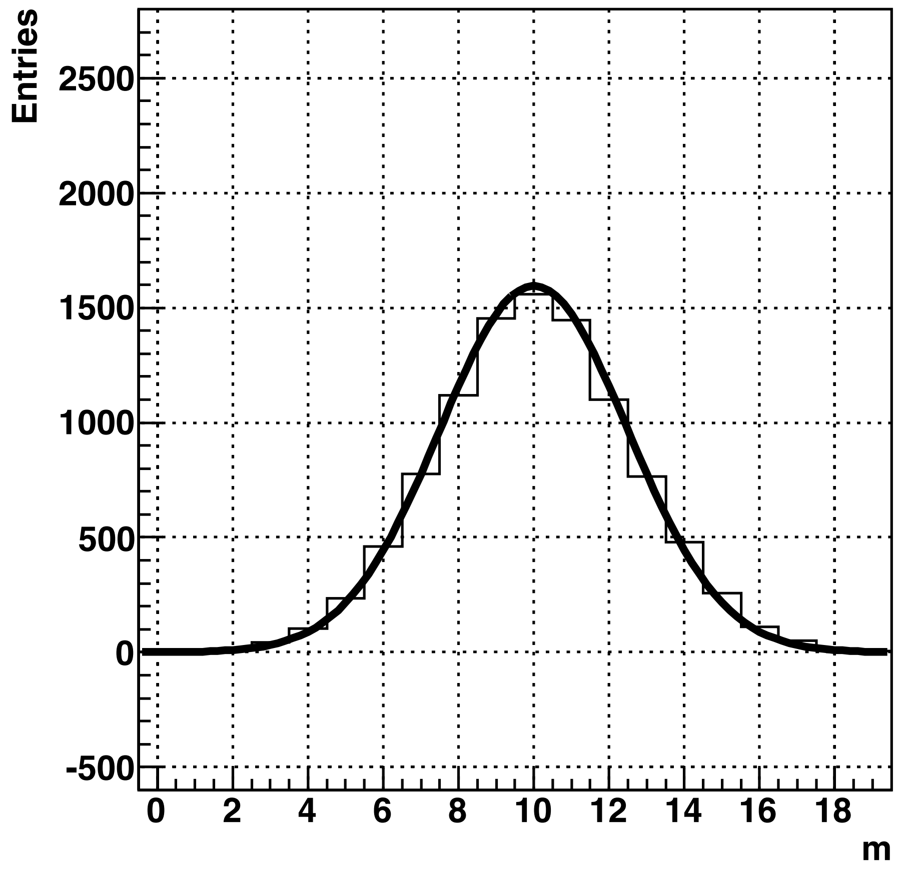
estimate $\hat{\vec{\mu}} = R^{-1}\vec{n}$

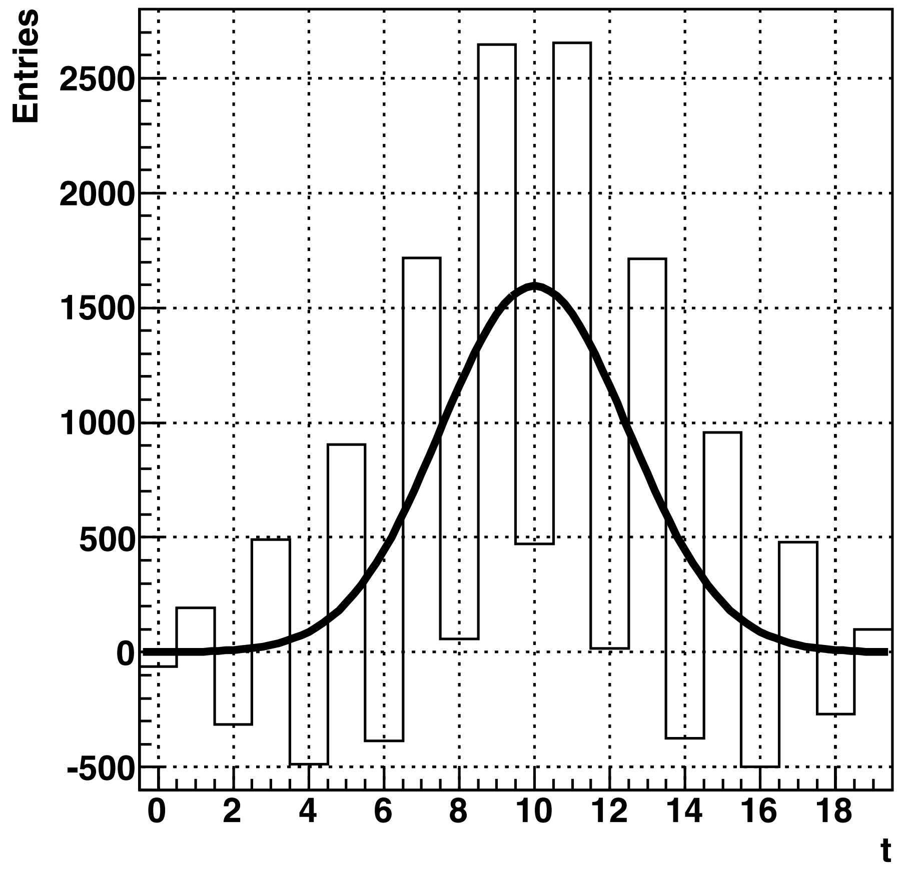
This looks like a disaster … unfolded distribution very different from the true one

Unfolding by inverting the response matrix (3)

Another example:

$$R = \begin{pmatrix} 0.75 & 0.25 & 0 & & \cdots \\ 0.25 & 0.50 & 0.25 & 0 & \\ 0 & 0.25 & 0.50 & 0.25 & \\ & 0 & 0.25 & 0.50 & \\ \vdots & & & & \ddots \end{pmatrix}.$$





Same conclusion: we don't get the desired (smooth) answer

What's wrong with the matrix inversion method?

Unbiased, minimum variance, actually also a ML estimator … all very nice!

The result is not wrong, it is just not desirable

- Does not really look like the original distribution
- Large correlation between bins

"Applying the response matrix R smears out fine structure
$\rightarrow$ applying R^{-1} creates (usually unwanted) structure"

More desirable solution by adding (smoothness) constraints.
However, this will produce a bias.

The art of unfolding is to find an acceptable balance between bias and smoothness.

Unfolding software: RooUnfold (ROOT Unfolding framework)

https://gitlab.cern.ch/RooUnfold/RooUnfold

RooUnfold is a framework for unfolding (AKA "deconvolution" or "unsmearing"). It currently implements six methods:

- iterative ("Bayesian"; as proposed by D'Agostini);
- singular value decomposition (SVD, as proposed by Höcker and Kartvelishvili and implemented in TSVDUnfold);
- bin-by-bin (simple correction factors);
- an interface to the TUnfold method developed by Stefan Schmitt;
- simple inversion of the response matrix without regularisation; and
- iterative dynamically stabilized unfolding (IDS) by Bogdan Malaescu, implemented by Chris Meyer.

It can be used from the ROOT prompt, from C++ (Cling) or Python (PyROOT) scripts, or linked against the ROOT libraries.

Bin-by-bin method (1)

Used very often, but has issues …

Assume shape of true spectrum and determine correction factor for each bin (usually determined from Monte Carlo simulation):

$$\mu_i = C_i(n_i - \beta_i) \qquad\qquad C_i = \frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}}$$

Works if smearing (bin-to-bin sharing) is negligible, only loss due to finite efficiency:

$$R_{ij} \approx \delta_{ij}\varepsilon_j$$

Obviously works, too, if MC = nature.

Expectation value for corrected data:

$$E[\hat{\mu}_i] = C_i E[n_i - \beta_i] = C_i(\nu_i - \beta_i) \equiv C_i \nu_i^{\mathrm{sig}}$$

Bin-by-bin method (2)

Inserting the C_i's one can determine the bias:

$$E[\hat{\mu}_i] = \frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}}\nu_i^{\mathrm{sig}} = \underbrace{\left(\frac{\mu_i^{\mathrm{MC}}}{\nu_i^{\mathrm{MC}}} - \frac{\mu_i}{\nu_i^{\mathrm{sig}}}\right)\nu_i^{\mathrm{sig}}}_{\text{bias}} + \mu_i$$

no bias only if MC = nature

Covariance matrix of the corrected data (smearing fluctuations independent between bins)

$$U_{ij} = \mathrm{cov}[\hat{\mu}_\mathrm{i}, \hat{\mu}_\mathrm{j}] = C_i C_j \underbrace{\mathrm{cov}[n_i^{\mathrm{sig}}, n_j^{\mathrm{sig}}]}_{0 \text{ for } i \neq j} = C_i^2 \mathrm{Var}[n_i^{\mathrm{sig}}]\delta_{ij}$$

Iterative bin-by-by method

- Start with plausible guess of true spectrum
- Apply correction to measurement
- Generate new correction factors from corrected spectrum of previous iteration
- And so on … usually a few iterations sufficient

Regularized unfolding

Matrix inversion is the maximum likelihood solution:

Independent Poisson fluctuations:

$$\ln L(\vec{\mu}) = \sum_{i=1}^{M} (n_i \ln \nu_i - \nu_i)$$

ML estimator:

$$\hat{\vec{\nu}} = \vec{n}$$
$$\rightarrow \hat{\vec{\mu}} = R^{-1}(\vec{n} - \vec{\beta})$$

Idea: accept solutions that are close to maximum likelihood estimate:

$$\ln L(\vec{\mu}) \geq \ln L(\vec{\mu}_{\mathrm{max}}) - \Delta \ln L(\vec{\mu})$$

Define a smoothness function *S* that gets bigger when the unfolded solution becomes smoother.

The task then is to maximize

$$\phi(\vec{\mu}) = \ln L(\vec{\mu}) + \underbrace{\tau}_{\text{parameter that controls the strength go the regularization}} \underbrace{S(\mu)}_{\text{smoothness function}}$$

Tikhonov regularization

Smoothness measure for the deconvoluted function *f*:

$$S[f] = -\int \mathrm{d}x \left(\frac{\mathrm{d}^k f}{\mathrm{d}^k x}\right)^2 \qquad k = 1, 2, 3, \ldots$$

Minus sign makes *S* small when *k*-th derivative is large

Tikhonov for *k* = 2:

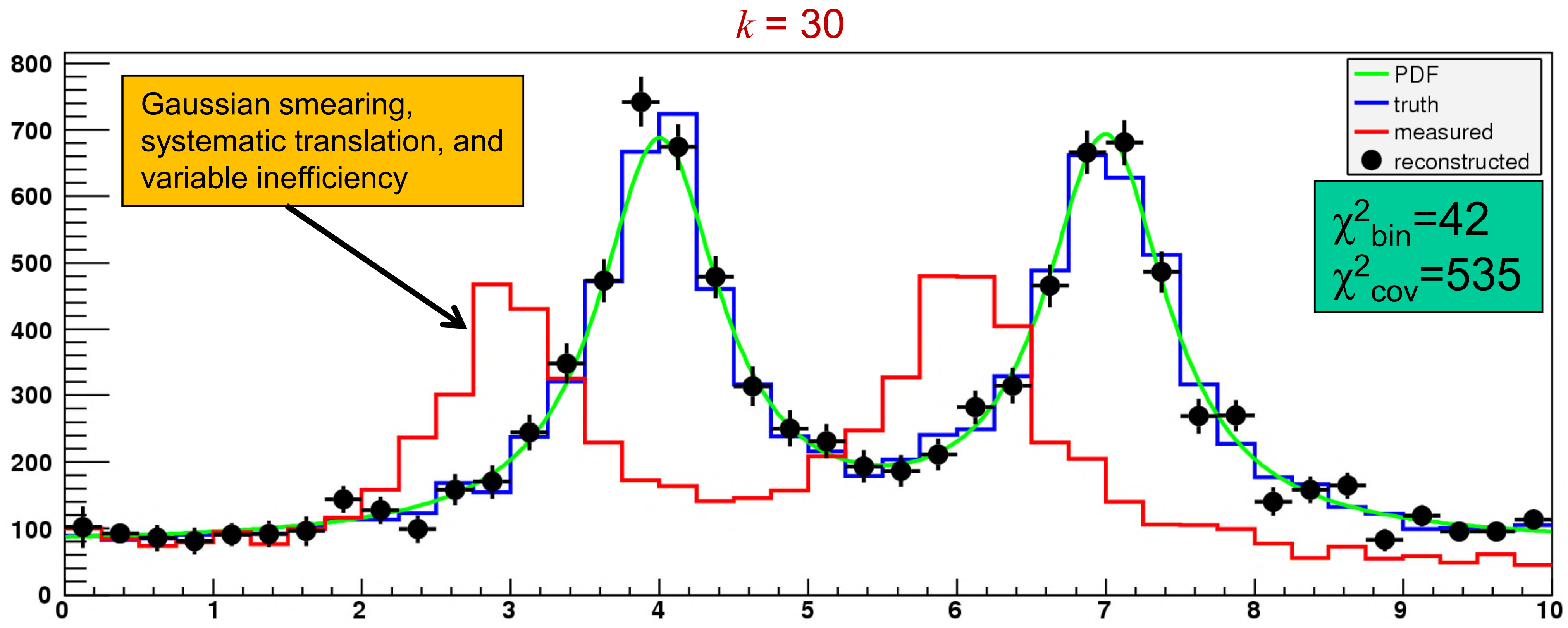
numerical estimate of the second derivative:

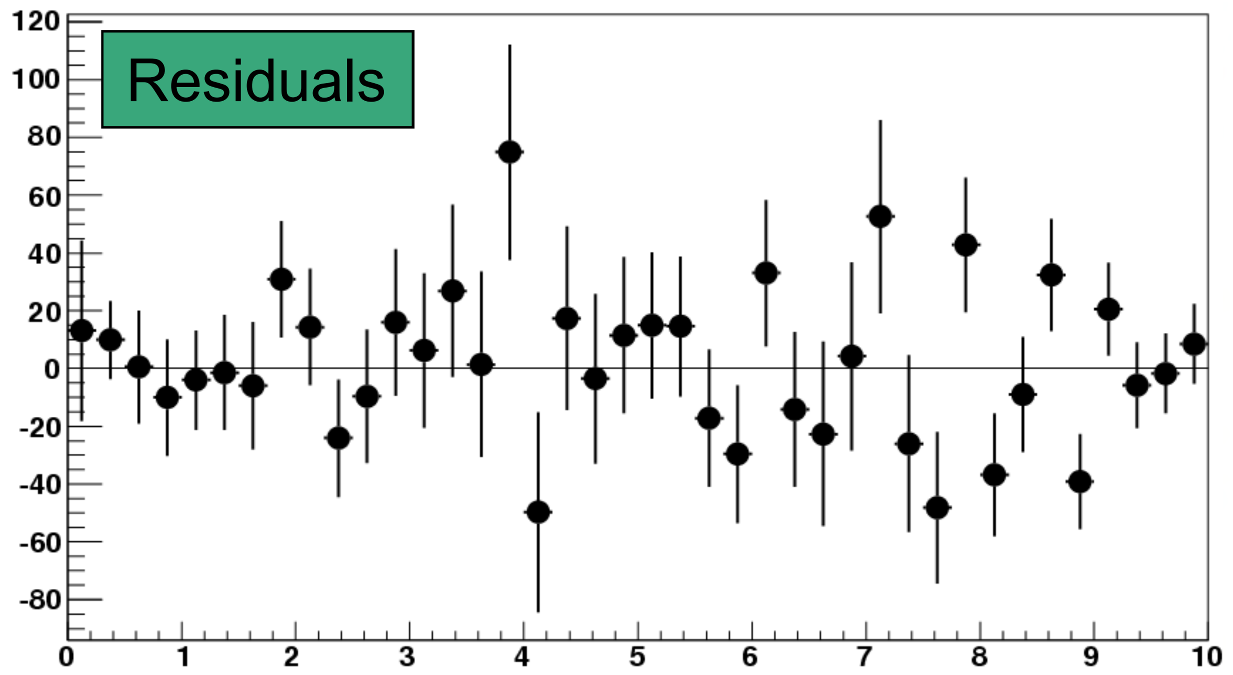
$$\phi(\vec{\mu}) = \ln L(\vec{\mu}) + \tau S(\mu), \qquad S(\vec{\mu}) = -\sum_{i=1}^{M-2} (-\mu_i + 2\mu_{i+1} - \mu_{i+2})^2$$

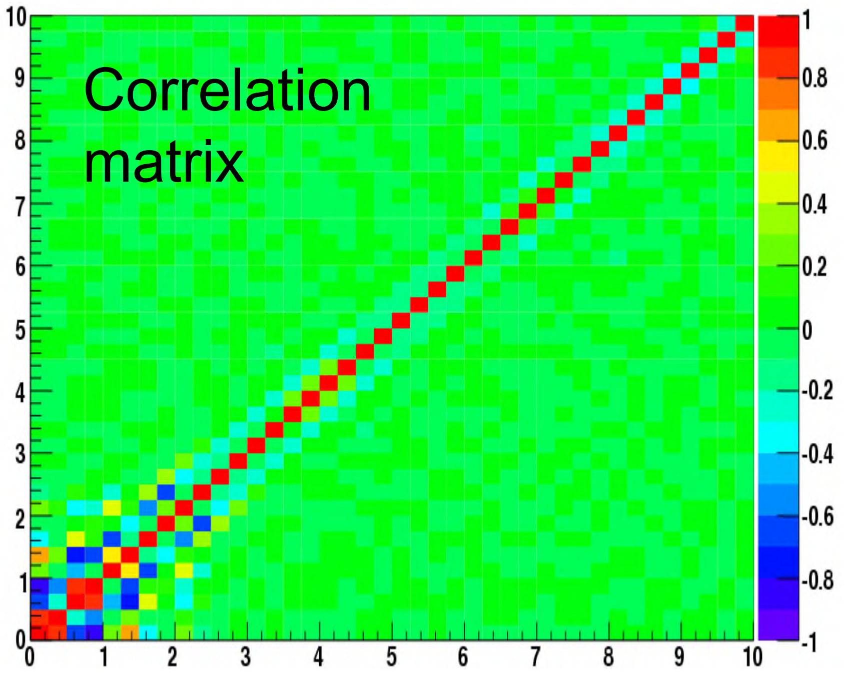
Implementation by A. Höcker, V. Kartvelishvili: *Singular Value Decomposition*
(NIM A372 (1996) 469, hep-ph/9509307, TSVDUnfold in ROOT)

Minimizes $$\chi^2(\vec{\mu}) + \tau \sum_i \left[(\mu_{i+1} - \mu_i) - (\mu_i - \mu_{i-1})\right]^2$$

RooUnfold with SVD algorithm





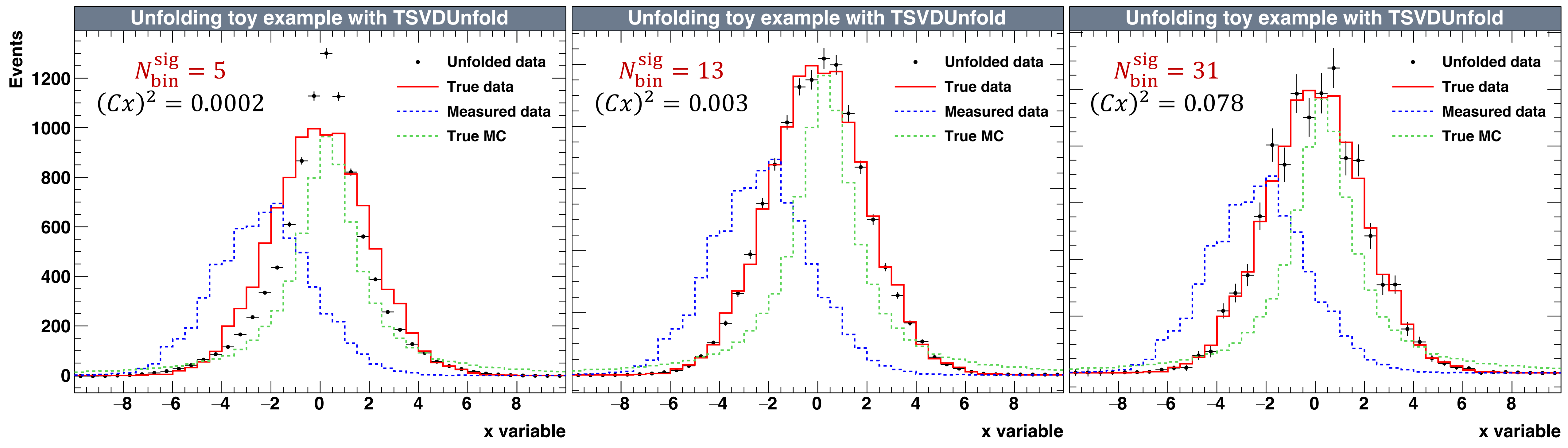


http://hepunx.rl.ac.uk/~adye/software/unfold/RooUnfold.html
Tim Adye, Unfolding in HEP, https://indico.cern.ch/event/671301/contributions/2745801

Over- and under-regularized unfolding

Andreas Höcker, https://indico.cern.ch/event/634037/

Over-regularised | Best regularisation choice | Under-regularised



The parameter determines the strength of the regularization

- τ too small → oscillations
- τ too large → unfolded spectrum biased towards MC

Regularization based on entropy

Shannon entropy of a discrete probably distribution:

$$H = -\sum_{i=1}^{M} p_i \ln p_i$$

all p_i equal → maximum entropy (max. smoothness)

p_i = 1, all others 0 → minimum entropy

Use entropy as regularization function:

$$S(\vec{\mu}) = H(\vec{\mu}) = -\sum_{i=1}^{M} \frac{\mu_i}{\mu_{\text{tot}}} \ln \frac{\mu_i}{\mu_{\text{tot}}}$$

This gives the distribution with the maximum entropy consistent within the selected tolerance with the ML solution

Entropy related to number of different ways μ_{tot} objects can be distributed to obtains histogram $\vec{\mu} = (\mu_1, \ldots, \mu_M)$

$$\Omega(\vec{\mu}) = \frac{\mu_{\text{tot}}!}{\mu_1! \cdot \ldots \cdot \mu_M!} \quad \overset{\text{Stirling's approximation}}{\rightsquigarrow} \quad \ln \Omega(\vec{\mu}) \approx \mu_{\text{tot}} H(\vec{\mu})$$

Relation of maximum entropy regularization and Bayesian approach

With a certain prior belief we obtain in the Bayesian approach:

$$f(\vec{\mu}|\vec{n}) \propto L(\vec{n}|\vec{\mu}) \cdot \pi(\vec{\mu})$$

Maximum entropy prior:

$$\pi(\vec{\mu}) \propto \Omega(\vec{\mu}) = \exp(\mu_{\mathrm{tot}} H(\vec{\mu}))$$

Logarithm of Bayesian posterior distribution:

$$\begin{aligned} \ln f(\vec{\mu}|\vec{n}) &\propto \ln L(\vec{n}|\vec{\mu}) + \ln \pi(\vec{\mu}) \\ &= \ln L(\vec{n}|\vec{\mu}) + \mu_{\mathrm{tot}} H(\vec{\mu}) \end{aligned}$$

We see: Bayesian approach = maximum likelihood with entropy regularization (with $\tau = \mu_{\mathrm{tot}}$)

Cross entropy regularization

Maximum entropy regularization penalizes deviation from flat distribution.

In some situations that might be the right assumption.

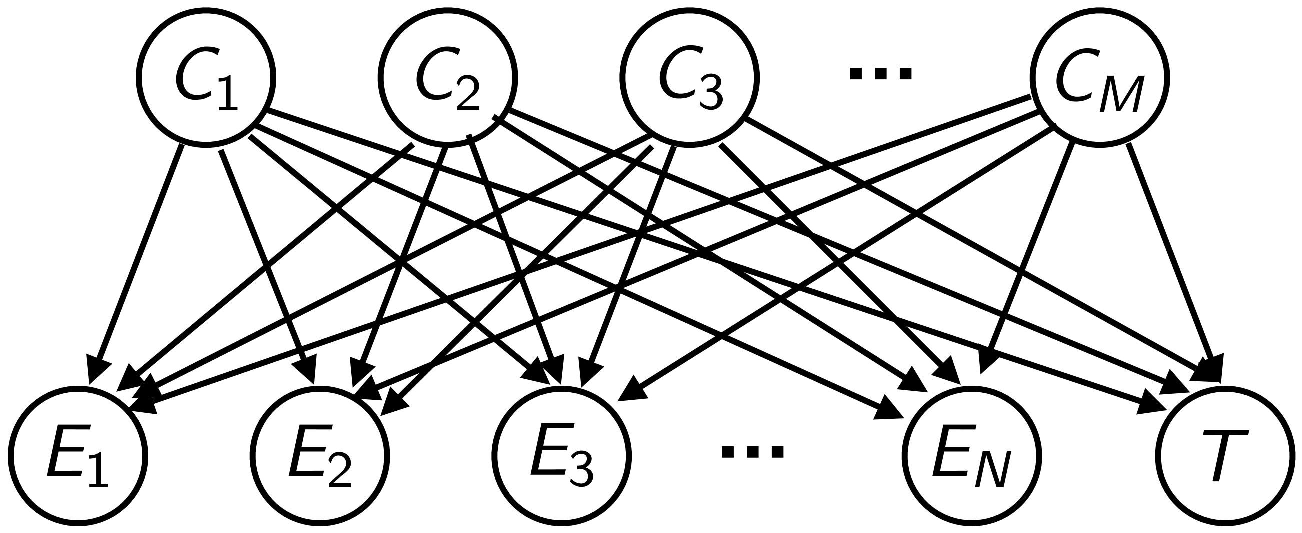
But if we have prior knowledge that the true events follow some distribution q, then we might want to use cross entropy:

$$K(f;q) = -\sum_{i=1}^{M} f_i \ln \frac{f_i}{M q_i}$$

If reference distribution q_i is flat, this reduces to regular entropy.

Iterative unfolding (1) [a.k.a. “Bayesian” unfolding]

Causal network:



M causes

N effects

trash bin for
undetected events

Response matrix:

I: prior knowledge about
probabilities of the causes C_i

$$R_{ji} = P(E_j | C_i, I)$$

$$E[n_j | \mu_i] = P(E_j | C_i, I) \cdot \mu_i = R_{ji} \cdot \mu_i$$

Iterative unfolding (2)

Bayes theorem:

$$P(C_i|E_j, I) = \frac{P(E_j|C_i, I) \cdot P(C_i|I)}{\sum_{k=1}^{M} P(E_j|C_k, I) \cdot P(C_k|I)}$$

We can write this as

$$\theta_{ij} := P(C_i|E_j, I) = \frac{R_{ji} \cdot P(C_i|I)}{\sum_{k=1}^{M} R_{jk} \cdot P(C_k|I)}$$

Estimate for number of true events in bin *i* given that we measure n_j events in bin *j*:

$$\mu_i|_{n_j} = \frac{P(C_i|E_j, I) \cdot n_j}{\varepsilon_i} = \frac{\theta_{ij} \cdot n_j}{\varepsilon_i}$$

Iterative unfolding (3)

Summing over all observed bins:

$$\mu_i|_{\vec{n}} = \frac{1}{\varepsilon_i} \sum_{j=1}^{N} \theta_{ij} \cdot n_j$$

By definition we can write the efficiency as

$$\varepsilon_i = \sum_{j=1}^{N} P(E_j|C_i, I) = \sum_{j=1}^{N} R_{ji}$$

Response matrix usually from Monte Carlo simulation

Iterative unfolding (4)

This procedure is applied iteratively:

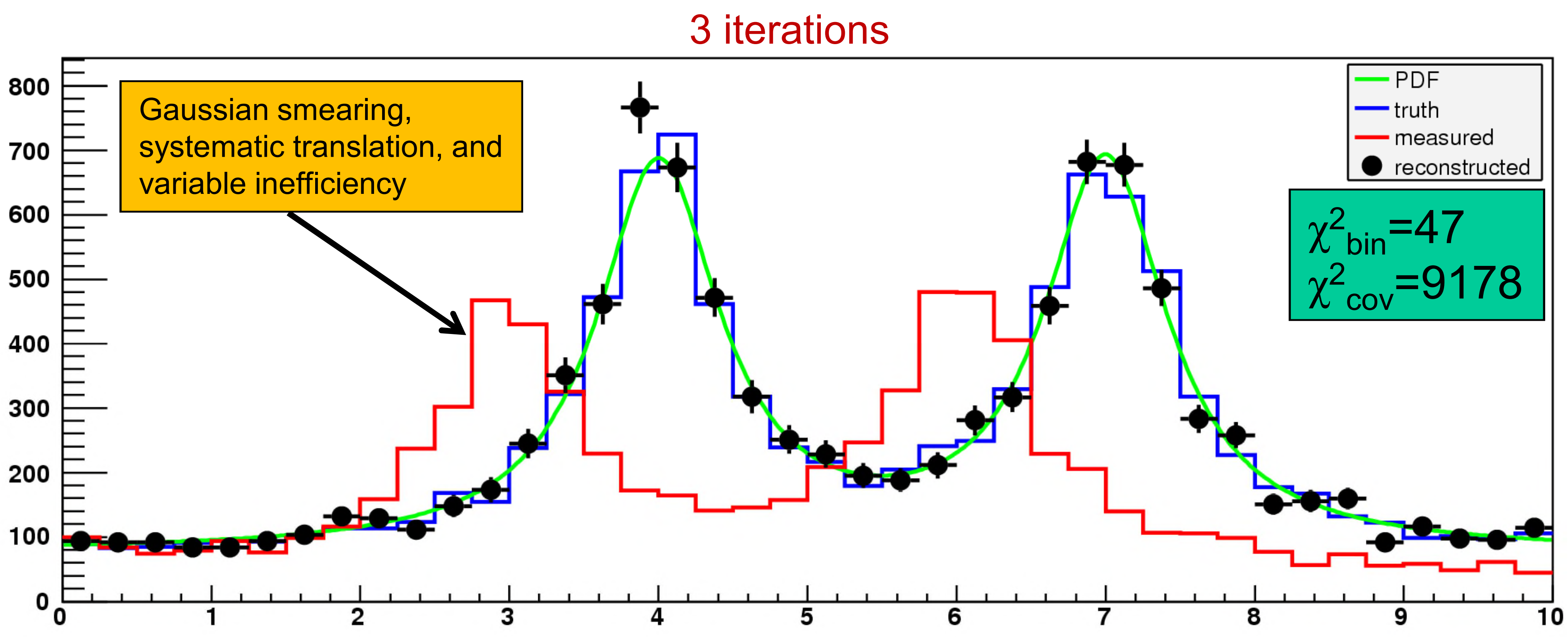
- Choose prior distribution $P(C_i,I)$
- Often prior = measured distribution
- Update prior according to measured values
- iterate
- Limited number of iterations provides implicit regularization

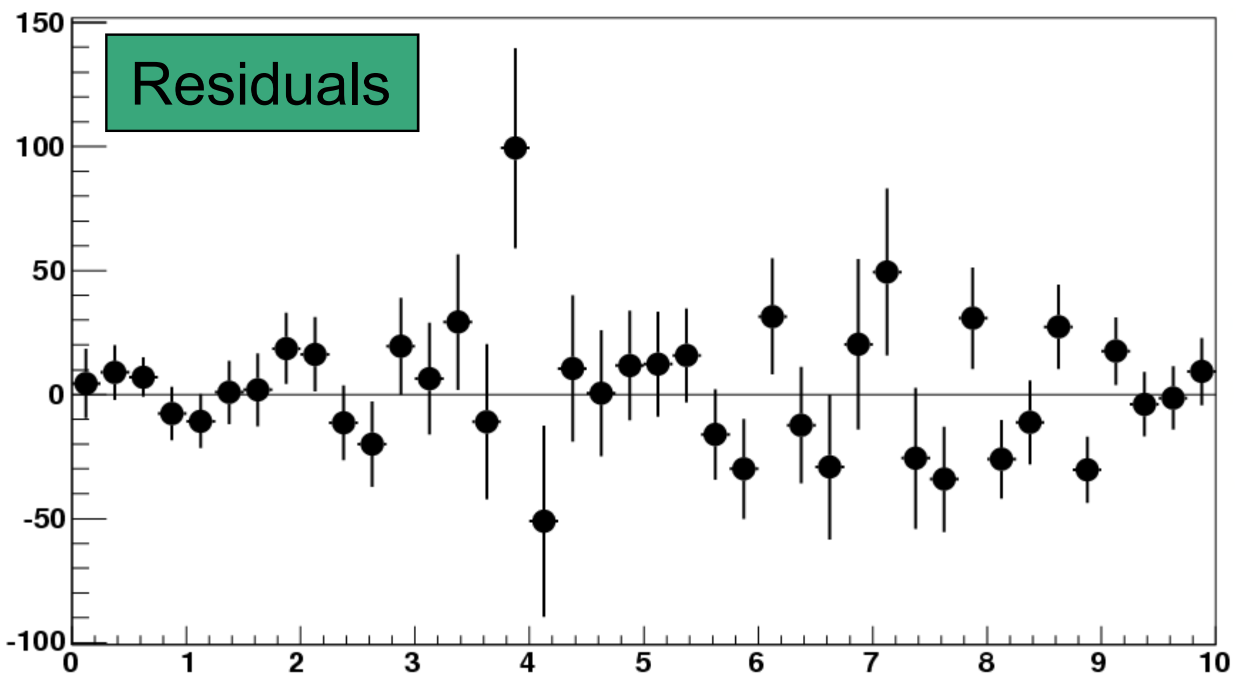
Shepp/Vardi 1982, Mülthei/Schorr 1986

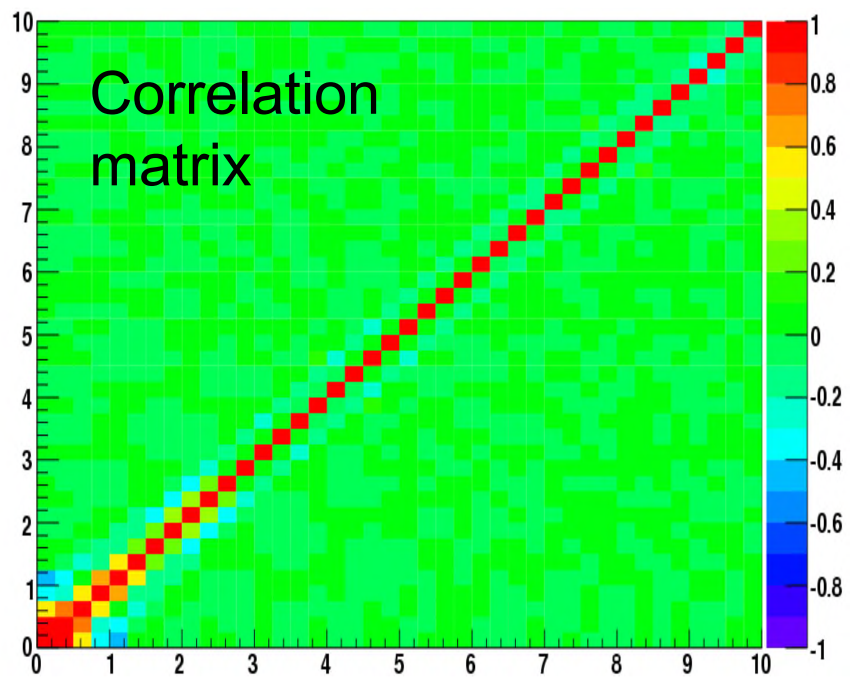
G. D'Agostini, A Multidimensional unfolding method based on Bayes' theorem'', Nucl. Instrum. Meth. A 362 (1995) 487 (see also arXiv:1010.0632)

V. Blobel, Unfolding methods in high-energy physics experiments, https://cds.cern.ch/record/157405

RooUnfold with iterative Bayesian algorithm







http://hepunx.rl.ac.uk/~adye/software/unfold/RooUnfold.html
Tim Adye, Unfolding in HEP, https://indico.cern.ch/event/671301/contributions/2745801

Which method to choose?

There is no "best" method. Depends on the analysis.

Main questions:

How to choose regularization parameter?

After how many iterations to stop in the iterative Bayesian unfolding?

Danger: Regularization and early stopping in iterative unfolding introduce a bias

Don't forget:
it some cases it is most useful to publish the folding matrix with the result

Stefan Schmitt, DESY
http://www.desy.de/~sschmitt/talks/UnfoldStatSchool2014.pdf