DESY 12-129
September 2012

ISSN 0418-9833

arXiv:1205.6201v4 [physics.data-an] 15 Nov 2012

TUnfold, an algorithm for correcting migration effects in high energy physics

Stefan Schmitt, DESY, Notkestraße 85, 22607 Hamburg
email: sschmitt@mail.desy.de

Abstract

TUnfold is a tool for correcting migration and background effects in high energy physics for multi-dimensional distributions. It is based on a least square fit with Tikhonov regularisation and an optional area constraint. For determining the strength of the regularisation parameter, the L-curve method and scans of global correlation coefficients are implemented. The algorithm supports background subtraction and the propagation of statistical and systematic uncertainties, in particular those originating from limited knowledge of the response matrix. The program is interfaced to the ROOT analysis framework.

Accepted by JINST as technical report

1 Introduction

In high energy physics, experiments are usually performed as counting experiments, where events are grouped into certain regions of phase-space, also called bins. However, the kinematic properties of each event, such as four-momenta of particles and derived quantities, are measured only at finite precision due to inevitable detector effects. As a consequence, events may be found in the wrong bin. Furthermore there is the presence of background, such that only a fraction of the events observed in a given bin originates from the reaction one is interested in.

In most cases, algorithms such as GEANT [1] are used to simulate migrations imposed by detector effects, whereas underlying physics processes are simulated using event generators such as PYTHIA [2]. After tracking the generated events through the detector simulation one is able to confront the physics process modelled by the event generator with the background-subtracted data.

However, often one is interested to report results such as differential cross sections, independent of the detector simulation. In that case, the observed event counts have to be corrected for detector effects. The problem may be written as

$$\tilde{y}_i = \sum_{j=1}^{m} A_{ij}\tilde{x}_j,\ 1 \leq i \leq n \tag{1}$$

where the m bins $\tilde{x}_j$ represent the true distribution, A_{ij} is a matrix of probabilities describing the migrations from bin j to any of the n bins on detector level and $\tilde{y}_i$ is the average expected event count at detector level. It is important to note here that the observed event counts y_i may be different from the average $\tilde{y}_i$ due to statistical fluctuations. A schematic view is given in figure 1. The situations becomes

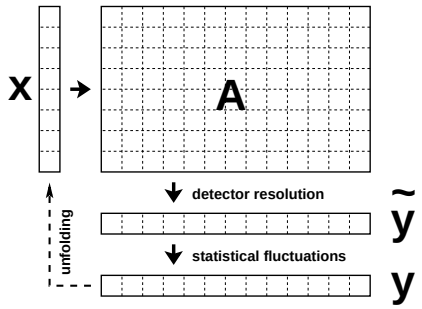

Figure 1: schematic view of migration effects and statistical fluctuations

somewhat more complicated if there is background. In that case the $\tilde{y}_i$ receive an additional contribution from background,

$$\tilde{y}_i = \sum_{j=1}^{m} A_{ij}\tilde{x}_j + b_i,\ 1 \leq i \leq n \tag{2}$$

where b_i is the background showing up in bin i. Both the background and the matrix of probabilities often suffer from systematic uncertainties which have to be considered in addition to the statistical uncertainties.

One may be tempted to replace $\tilde{y}_i \to y_i$ and $\tilde{x}_j \to x_j$ in equations 1 or 2 and then solve for x_j, simply by inverting the matrix of probabilities. However, it turns out that the statistical fluctuations of the y_i are amplified when calculating the x_j this way. Such fluctuations are often damped by imposing certain smoothness conditions on the x_j. This procedure is termed "regularisation".

The TUnfold algorithm [3], described in this paper and interfaced to the ROOT analysis package [5], implements a procedure to estimate the $\tilde{x}_j$ using a least square method with Tikhonov regularisation [4] and an optional area constraint. In order to obtain best results from the least square minimisation, the number of degrees of freedom, $n - m$, has to be larger than zero. It means that the data y_i have

to be measured in finer bins than are extracted by the unfolding procedure. This condition $n \geq m$ is in contrast to some other commonly used unfolding methods, where often the restriction $n = m$ is imposed [6, 7]. Examples of unfolding algorithms which do not have the restriction $n = m$ are [8, 9].

No attempt is made here to give a complete overview of the commonly used unfolding algorithms. The TUnfold algorithm [3] presented here compares best to algorithms based on matrix inversion or singular value decomposition, like [6, 10]. Alternative approaches are often based on iterative methods or on the use of Bayes' theorem, for example [7, 8, 9]. Many reviews on the topic can be found in literature, only two examples are given here [11, 12].

2 The TUnfold algorithm

2.1 Definitions

The TUnfold algorithm gives an estimator of a set of truth parameters, using a single measurement of a set of observables. The observables are described by a vector[1] of random variables, $\boldsymbol{y}$. The random variables $\boldsymbol{y}$ are taken to have a multivariant Gaussian distribution with mean $\tilde{\boldsymbol{y}} = \mathbf{A}\tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}}$ is a vector corresponding to the set of of truth parameters and $\mathbf{A}$ is a matrix. The covariance matrix of $\boldsymbol{y}$ is $\mathbf{V_{yy}}$. The algorithm only works if the dimension of $\tilde{\boldsymbol{x}}$ is less or equal to the dimension of $\tilde{\boldsymbol{y}}$. Furthermore, $\mathbf{V_{yy}}$ has to have full rank and the rows of $\mathbf{A}$ shall be linear independent. The algorithm returns an estimator $\boldsymbol{x}$ of the truth parameters $\tilde{\boldsymbol{x}}$, given an observation $\boldsymbol{y}$. The estimator $\boldsymbol{x}$, when considered as a random variable, has a covariance matrix which is also calculated. It is labelled $\mathbf{V_{xx}}$.

2.2 Algorithm

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

$$\begin{aligned}
\mathcal{L}(x,\lambda) =& \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 && \text{where} && (3)\\
\mathcal{L}_1 =& (\boldsymbol{y} - \mathbf{A}\boldsymbol{x})^{\mathsf{T}} \mathbf{V_{yy}}^{-1} (\boldsymbol{y} - \mathbf{A}\boldsymbol{x}), && && (4)\\
\mathcal{L}_2 =& \tau^2 (\boldsymbol{x} - f_b \boldsymbol{x_0})^{\mathsf{T}} (\mathbf{L}^{\mathsf{T}}\mathbf{L})(\boldsymbol{x} - f_b \boldsymbol{x_0}), && && (5)\\
\mathcal{L}_3 =& \lambda (Y - \boldsymbol{e}^{\mathsf{T}}\boldsymbol{x}) && \text{and} && (6)\\
Y =& \sum_i y_i, && && (7)\\
e_j =& \sum_i A_{ij}, && && (8)
\end{aligned}$$

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

[1] Throughout this paper, matrices ($\mathbf{M}$) and vectors ($\boldsymbol{v}$) are printed in bold. Matrices or vectors without indices, written next to each other, are multiplied. Where needed, brackets with indices are used to refer to specific elements. The notation $\mathbf{M}^{\mathsf{T}}$ indicates that a matrix is transposed, its rows and columns are swapped. The inverse of $\mathbf{M}$ is written as $\mathbf{M}^{-1}$. A vector is treated as a matrix with only one column, such that a transposed vector has only one row. The dot product of two vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ thus is equivalent to the matrix multiplication $\boldsymbol{v_1}^{\mathsf{T}}\boldsymbol{v_2}$. Other examples are $(\mathbf{A}\boldsymbol{x})_i = \sum_j A_{ij}x_j$ and $(\mathbf{A}^{\mathsf{T}})_{ij} = A_{ji}$.

while determining the stationary point of $\mathcal{L}$. The matrix $\mathbf{L}$ has n 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 $f_b = 0$, $n_R = n$ 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.

The term $\mathcal{L}_3$ is an optional area constraint. There is a Lagrangian parameter λ. The sum over all observations is given by Y, equation 7. The efficiency vector $\boldsymbol{e}$ has m rows and is calculated from $\mathbf{A}$ as indicated in equation 8. If the area constraint is used, the normalisation of the result $\boldsymbol{x}$, corrected for the efficiencies $\boldsymbol{e}$, is thus enforced to match the total event count Y. This procedure is applied in order to limit possible biases on the normalisation which are present if the data $\boldsymbol{y}$ follow Poisson's statistics whereas the least square ansatz is strictly valid only for normal distributed measurements. The problem is discussed in more detail in literature, for example in [13].

The minimum or stationary point of $\mathcal{L}(\boldsymbol{x}, \lambda)$ is determined by setting the first derivatives to zero. In the case without area constraint, λ is set to zero and only the derivatives of $\mathcal{L}_1 + \mathcal{L}_2$ with respect to the components of $\boldsymbol{x}$ are set to zero. When including the area constraint, the equations are solved for $\boldsymbol{x}$ and λ together. The partial derivatives of $\mathcal{L}(\boldsymbol{x}, \lambda)$ are

$$\frac{\partial \mathcal{L}(\boldsymbol{x},\lambda)}{\partial x_j} = -2\left(\mathbf{A}^{\mathsf{T}}{\mathbf{V_{yy}}}^{-1}(\boldsymbol{y} - \mathbf{A}\boldsymbol{x})\right)_j + 2\tau^2\left((\mathbf{L}^{\mathsf{T}}\mathbf{L})(\boldsymbol{x} - f_b\boldsymbol{x_0})\right)_j - \lambda e_j, \tag{9}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{x},\lambda)}{\partial \lambda} = Y - \boldsymbol{e}^{\mathsf{T}}\boldsymbol{x}. \tag{10}$$

$$\tag{11}$$

The stationary point $\boldsymbol{x}$ of $\mathcal{L}$ is found as

$$\boldsymbol{x} = \begin{cases} \boldsymbol{x}|_{\boldsymbol{\lambda=0}} & \text{without area constraint} \\ \boldsymbol{x}|_{\boldsymbol{\lambda=0}} + \frac{\lambda}{2}\mathbf{E}\boldsymbol{e} & \text{with area constraint} \end{cases} \qquad \text{where} \tag{12}$$

$$\boldsymbol{x}|_{\boldsymbol{\lambda=0}} = \mathbf{E}\left[\mathbf{A}^{\mathsf{T}}{\mathbf{V_{yy}}}^{-1}\boldsymbol{y} + \tau^2(\mathbf{L}^{\mathsf{T}}\mathbf{L})f_b\boldsymbol{x_0}\right], \tag{13}$$

$$\mathbf{E} = \left(\mathbf{A}^{\mathsf{T}}{\mathbf{V_{yy}}}^{-1}\mathbf{A} + \tau^2(\mathbf{L}^{\mathsf{T}}\mathbf{L})\right)^{-1} \qquad \text{and} \tag{14}$$

$$\frac{\lambda}{2} = \frac{Y - \boldsymbol{e}^{\mathsf{T}}\boldsymbol{x}|_{\boldsymbol{\lambda=0}}}{\boldsymbol{e}^{\mathsf{T}}\mathbf{E}\boldsymbol{e}}. \tag{15}$$

In order to calculate the covariance matrix of $\boldsymbol{x}$, given the covariance matrix of $\boldsymbol{y}$, the corresponding partial derivatives are calculated

$$(\mathbf{D^{xy}})_{ki} := \frac{\partial x_k}{\partial y_i} = \begin{cases} B_{ki} & \text{without area constraint} \\ B_{ki} + (\mathbf{E}\boldsymbol{e})_k \frac{1-(\mathbf{B}^{\mathsf{T}}\boldsymbol{e})_i}{\boldsymbol{e}^{\mathsf{T}}\mathbf{E}\boldsymbol{e}} & \text{with area constraint} \end{cases} \qquad \text{where} \tag{16}$$

$$\mathbf{B} = \mathbf{E}\mathbf{A}^{\mathsf{T}}{\mathbf{V_{yy}}}^{-1}. \tag{17}$$

The covariance matrix of the result $\boldsymbol{x}$, originating from $\mathbf{V_{yy}}$ is thus given by

$$\mathbf{V_{xx}} = \mathbf{D^{xy}}\mathbf{V_{yy}}(\mathbf{D^{xy}})^{\mathsf{T}}. \tag{18}$$

3 Normalisation of the matrix of migrations

In most cases, $\mathbf{A}$ is determined from Monte Carlo simulations. Within TUnfold, it is foreseen to initialise the unfolding from a matrix $\mathbf{M}$ of event counts, determined in a Monte Carlo event simulation, where $\mathbf{M}$ has $n+1$ rows and m columns, one row more than $\mathbf{A}$. The extra row is used to count those events which are generated in a particular bin j but are not found in any of the reconstructed bins. For the purpose of this paper, the extra row of M is denoted with index $i = 0$, whereas all other matrices and vectors

have indices starting from 1. In other words, the matrix elements M_{ij} count the Monte Carlo events generated in bin j of $\boldsymbol{x}$ and reconstructed in bin $i > 0$ of $\boldsymbol{y}$, whereas the matrix elements M_{0j} count the Monte Carlo events generated in bin j and not reconstructed in any of the bins of $\boldsymbol{y}$. For the unfolding algorithm, $\mathbf{A}$ and $\boldsymbol{x_0}$ are initialised from $\mathbf{M}$ as follows

$$A_{ij} = \frac{M_{ij}}{s_j}, \text{ where } i > 0 \qquad \text{and} \tag{19}$$

$$s_j = \sum_{i=0}^{n} M_{ij}, \tag{20}$$

$$(\boldsymbol{x_0})_j = s_j. \tag{21}$$

4 Choice of the regularisation strength

When unfolding, the strength of the regularisation, τ^2, is an unknown parameter. If τ^2 is too small, the unfolding result often has large fluctuations and correspondingly large negative correlations of adjacent bins. If τ^2 is too large, the result is biased towards $f_b \boldsymbol{x_0}$. Several methods to choose the strength of the regularisation are discussed in literature, for example eigenvalue analyses [14], minimisation of correlation coefficients [15], and the L-curve method [16]. At present, in TUnfold a simple version of the L-curve method is implemented to determine τ^2 as well as methods to minimise global correlation coefficients.

4.1 L-curve scan

The idea of the L-curve method is to look at the graph of two variables L_x^{curve} and L_y^{curve} and locate the point where the curvature is maximal. These variables are defined as

$$L_x^{\text{curve}} = \log \mathcal{L}_1 \qquad \text{and} \tag{22}$$

$$L_y^{\text{curve}} = \log \frac{\mathcal{L}_2}{\tau^2}, \tag{23}$$

such that L_x tests the agreement of x with the data and L_y tests the agreement of x with the regularisation condition. Note that L_y^{curve} does not have an explicit dependence on τ^2. For $\tau^2 \to 0$ the value of L_x^{curve} is minimal and L_y^{curve} is maximal, because $\mathcal{L}_2 \to 0$ and $\boldsymbol{x}$ corresponds to the stationary point of $\mathcal{L}_1 + \mathcal{L}_3$. As τ^2 gets large, L_x^{curve} increases whereas L_y^{curve} is getting small, because the Lagrangian is dominated by $\mathcal{L}_2$. It is observed that the parametric plot of L_y^{curve} against L_x^{curve} often shows a kink (is L-shaped). The kink location is chosen to determine τ^2.

In TUnfold, the L-curve algorithm is implemented as follows: the unfolding is repeated for a number of points in $t = \log \tau$, thus scanning the L-curve. The curvature $\mathcal{C}$ of the L-curve is determined as

$$\mathcal{C} = \frac{\mathrm{d}^2 L_y^{\text{curve}} \mathrm{d}L_x^{\text{curve}} - \mathrm{d}^2 L_x^{\text{curve}} \mathrm{d}L_y^{\text{curve}}}{\left((\mathrm{d}L_x^{\text{curve}})^2 + (\mathrm{d}L_y^{\text{curve}})^2\right)^{\frac{3}{2}}}. \tag{24}$$

The first and second derivatives of L_x^{curve} (L_y^{curve}) with respect to t, $\mathrm{d}L_x^{\text{curve}}$ ($\mathrm{d}L_y^{\text{curve}}$) and $\mathrm{d}^2 L_x^{\text{curve}}$ ($\mathrm{d}^2 L_y^{\text{curve}}$), respectively, are approximated using cubic spline parametrisations of the scan results. The maximum of $\mathcal{C}$ is finally determined with the help of a cubic spline parametrisation of $\mathcal{C}(t)$.

4.2 Minimising global correlation coefficients

A method of minimising global correlation coefficients is also implemented. Given the covariance matrix $\mathbf{V_{xx}}$ the global correlation coefficient of a component i of $\boldsymbol{x}$ is defined as

$$\rho_i = \sqrt{1 - \frac{1}{(\mathbf{V_{xx}}^{-1})_{ii} (\mathbf{V_{xx}})_{ii}}}. \tag{25}$$

Two sorts of correlation coefficients scans have been implemented:

1. minimising the average correlation: the regularisation strength τ^2 is chosen such that the average global correlation $\sum_i \rho_i/n$ is minimised, where n is the dimension of $\boldsymbol{x}$.
2. minimising the maximum correlation: the regularisation strength τ^2 is chosen such that the maximum correlation $\max_i(\rho_i)$ is minimised.

Furthermore, it is possible to choose the covariances

1. The covariance matrix V_{xx} may or may not include systematic uncertainties.
2. There is the option to partition the covariance matrix such that only parts of the matrix are used for the calculation of global correlation coefficients.
3. It is possible to merge bins or groups of bins prior to calculating the ρ_i.

When partitioning the covariance matrix, the corresponding unused rows and columns of $\mathbf{V_{xx}}$ are removed prior to inverting the matrix and calculating the global correlation coefficients. When merging bins of groups of bins, the corresponding rows or columns of the matrix are added up.

The scan is implemented such that the unfolding is repeated for a number of points in $t = \log\tau$. For each point the chosen correlation type (maximum or average) is calculated. The minimum is determined using a cubic spline interpolation.

5 Background subtraction

Often there is background present in the measured data $\boldsymbol{y}$. It is worth to mention that the background has to include all types of events which are possibly reconstructed in one of the bins of $\boldsymbol{y}$ but do not originate from any of the bins of $\boldsymbol{x}$. In particular, part of the signal process may be generated outside the phase-space covered by $\boldsymbol{x}$ and thus has to be counted as background. Sometimes it is possible to determine background sources from the data as a part of the unfolding process, for example using a discriminator [17]. In order to achieve that, background normalisation factors are included as extra bins of the vector $\boldsymbol{x}$, corresponding to extra columns of the matrices $\mathbf{A}$, $\mathbf{M}$. The background normalisation is then determined in the unfolding process.

On the other hand, it is often useful to simply subtract the background prior to unfolding. Within TUnfold, the following method of background subtraction is implemented

$$\boldsymbol{y} = \boldsymbol{y^0} - f^b\boldsymbol{b}, \tag{26}$$

$$(\mathbf{V_{yy}})_{ij} = (\mathbf{V^0_{yy}})_{ij} + \delta_{ij}(f^b(\boldsymbol{\delta b})_i)^2 + (\delta f^b)^2 b_i b_j. \tag{27}$$

Here, the components of $\boldsymbol{y^0}$ are the data prior to background subtraction, with covariance matrix $\mathbf{V^0_{yy}}$. The background has a normalisation factor f^b with uncertainty δf^b. The background shape is described by a vector $\boldsymbol{b}$ and the uncertainties on the components of $\boldsymbol{b}$ are given by the vector $\boldsymbol{\delta b}$. Finally, δ_{ij} is the Kronecker symbol.

The covariance matrix $\mathbf{V_{yy}}$ receives contributions from the covariance matrix of $\boldsymbol{y^0}$ as well as from the uncertainties on the background shape, the latter contributing only to the diagonal elements. In addition there are contributions to the covariance matrix from the background normalisation uncertainty. Because the background normalisation is correlated for all analysis bins, it also contributes to the off-diagonal elements of the matrix.

In TUnfold, the background subtraction is generalised such that multiple background sources may be subtracted. The contribution of individual sources of uncertainty to the result's covariance matrix may be studied after unfolding.

6 Systematic uncertainties on the matrix of migrations

The matrix of migrations, $\mathbf{A}$, usually receives uncertainties from various sources. First, there are statistical uncertainties, originating from counting the Monte Carlo events in the matrix $\mathbf{M}$. Second, there may be systematic uncertainties, in many cases described by a variation $\mathbf{M} \rightarrow \mathbf{M} + \boldsymbol{\delta}\mathbf{M}$, corresponding to a variation of experimental conditions.

The statistical uncertainties are bin-to-bin independent uncertainties ΔM_{ij} on $\mathbf{M}$. They are propagated through the unfolding formalism and result in a contribution $\mathbf{V}_{\mathbf{xx}}^{\mathbf{M},\mathrm{stat}}$ to the covariance matrix of $\boldsymbol{x}$. Details are given in the appendix.

A systematic variation $\boldsymbol{\delta}\mathbf{M}$ is propagated to the result vector in the form of a vector of systematic shifts, $\boldsymbol{\delta x}$. The corresponding covariance matrix contribution is given by $\mathbf{V}_{\mathbf{xx}}^{\boldsymbol{\delta}\mathbf{M}} = \boldsymbol{\delta x}(\boldsymbol{\delta x})^{\mathsf{T}}$. The calculation of $\boldsymbol{\delta x}$ is described in the appendix. TUnfold supports multiple sources of systematic variation.

7 Choice of regularisation conditions

Within TUnfold, the matrix of regularisation conditions $\mathbf{L}$ can be chosen with some flexibility. Three basic types of regularisation are supported:

1. rows of $\mathbf{L}$ where only one element is non-zero, corresponding to a regularisation of the amplitude or size of $\boldsymbol{x}$,
2. rows of $\mathbf{L}$ where two elements are non-zero, corresponding to a regularisation of the first derivative of $\boldsymbol{x}$,
3. rows of $\mathbf{L}$ where three elements are non-zero, corresponding to a regularisation of the second derivative (curvature) of $\boldsymbol{x}$.

The first derivatives are approximated by differences of event counts in adjacent bins, $x_{i+1} - x_i$. Similarly, the second derivatives are approximated by $(x_{i+1} - x_i) - (x_i - x_{i-1})$.

When initialising TUnfold, it is possible to choose one of the three basic types of regularisation. This type of regularisation is then applied to all bins of $\boldsymbol{x}$.

1. if TUnfold is initialised to regularise on the size, $\mathbf{L}$ is initialised to the unity matrix.
2. if TUnfold is initialised to regularise on the first derivatives, $\mathbf{L}$ has $n-1$ rows and the non-zero elements are: $L_{i,i} = -1$ and $L_{i,i+1} = 1$.
3. if TUnfold is initialised to regularise on the second derivatives, $\mathbf{L}$ has $n-2$ rows and the non-zero elements are: $L_{i,i} = 1$, $L_{i,i+1} = -2$, $L_{i,i+2} = 1$.

On the other hand, it is also possible to choose neither of the basic types and to set up details of the regularisation for specific bins or groups of bins instead.

7.1 Regularisation of multi-dimensional distributions

In many cases, $\boldsymbol{x}$ is not simply a one-dimensional distribution. Instead, the bins of $\boldsymbol{x}$ may originate from several distributions, for example if there are bins controlling the background normalisation in addition to the signal bins. Furthermore, the signal bins may originate from a multi-dimensional distribution. For example, the signal may have 4×3 bins in two variables P_T and η. The vector x then has 12 bins, where the first 4 bins correspond to the 4 P_T bins of the first η bin, etc. Such a structure is not problematic when regularising on the size, but care has to be taken when regularising on the first or second derivatives.

Within TUnfold there is support to initialise one-, two- or three-dimensional regularisation patterns. For example, when regularising the two-dimensional pattern of 4×3 bins from the (P_T, η) example above on the second derivative, $\mathbf{L}$ is set up as follows:

$$\mathbf{L} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{28}$$

Here, rows 1-2 correspond to the regularisation of the second derivatives on P_T for the first η bin. Similarly, rows 3-4 and 5-6 act on P_T for the second and third η bin, respectively. Finally, rows 7-10 correspond to the second derivatives in η for the four P_T bins.

7.2 Regularisation on the density, multi-dimensional distributions

The regularisation schemes discussed so far do not take into account the effects of non-uniform bin widths. Another complication arises in cases where multidimensional distributions of signal and backgrounds have to be mapped to the one-dimensional vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and to the matrix $\mathbf{M}$. The latest version of TUnfold [3] addresses these issues. Multidimensional distributions are mapped on one axis of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. The regularisation conditions may be refined such that the effects of non-uniform bin widths [19] are taken into account.

7.2.1 Densities

During the unfolding, the bins of $\boldsymbol{x}$ correspond to event counts. However, often is desirable to regularise not on the even count but on the density. The density is calculated by dividing the number of events in a given bin by the width of the bin. For calculating the regularisation conditions, the number of events $\boldsymbol{x}$ is transformed to a density $\hat{\boldsymbol{x}}$,

$$x_j \to \hat{x}_j = x_j \times \frac{f_j^{\rm user}}{\prod_d w_{dj}}. \tag{29}$$

The number of events x_j is divided by the multi-dimensional bin width $\prod_d w_{dj}$, where w_{dj} is the bin width of bin j in the dimension d, as specified by the underlying multidimensional distribution. In addition, there is an arbitrary user function $f_j^{\rm user}$, which may be used to compensate known kinematic factors[2]. In TUnfold, the transformation to the density is implemented by modifying the elements of the matrix $\mathbf{L}$,

$$L_{rj} \to L_{rj} \times \frac{f_j^{\rm user}}{\prod_d w_{dj}}, \tag{30}$$

where the index r is used to enumerate the n_R regularisation conditions.

[2] An example is the use of the "reduced cross section" rather than the ordinary cross section for inclusive deep-inelastic scattering [18]. The ordinary cross section changes by several orders of magnitude as a function of kinematic variables and hence is difficult to regularise. In contrast, the reduced cross section does not vary a lot, and thus is more natural to regularise on.

7.2.2 Derivatives

In the case where the regularisation is made on the derivatives, the bin width may also be included in the approximate calculation of the derivatives. The calculation of first derivatives is modified such that

$$(x_{j_2} - x_{j_1}) \to \frac{\Delta_d}{\delta^d_{j_2 j_1}}(x_{j_2} - x_{j_1}), \tag{31}$$

where j_2 and j_1 are the indices of adjacent bins of a multi-dimensional distribution and d is the dimension of the distribution for which the derivative is calculated. The distance between the two bin centres is $\delta^d_{j_2 j_1}$ and Δ_d is a normalisation constant specific to the dimension d. In TUnfold, the normalisation constant by default is chosen to be the average bin width in dimension d. The Δ_d are relevant if derivatives are considered for multi-dimensional distributions, where often the derivatives along one dimension are different in magnitude from derivatives along another dimension. For example, in the variable P_T the typical bin width may be 10 [GeV], where as in η the typical bin width may be 0.5. In this case, the derivatives in P_T typically are a factor of 20 smaller than those in η, unless the normalisation Δ_d is chosen appropriately.

In analogy to the case of first derivatives, the calculation of second derivatives may be modified to take into account bin widths using the transformation

$$(x_{j_3} - x_{j_2}) - (x_{j_2} - x_{j_1}) \to \frac{(\Delta_d)^2}{\delta^d_{j_2 j_1} + \delta^d_{j_3 j_2}} \left(\frac{x_{j_3} - x_{j_2}}{\delta^d_{j_3 j_2}} - \frac{x_{j_2} - x_{j_1}}{\delta^d_{j_2 j_1}} \right), \tag{32}$$

where j_1, j_2 and j_3 are indices corresponding to a triplet of adjacent bins of a multidimensional distribution. The distance of bin centres and normalisation factors are defined similar to the case of first derivatives.

In TUnfold, the calculation of first or second derivatives including bin widths is implemented by adding the appropriate modifications to the matrix $\mathbf{L}$. It is possible to use the modified calculation of derivatives together with the density calculation explained in section 7.2.1.

7.2.3 Example of a more complicated regularisation scheme

Consider the use of 4×3 bins in (P_T, η), where the bins borders in P_t are $[5, 7, 10, 15, 25]$ and the bin borders in η are $[-2, -0.5, 0.5, 2]$. The dimension $d = 1$ corresponds to P_T and $d = 2$ corresponds to η. In the example, the bin widths along p_T are $[2, 3, 5, 10]$ and those along η are $[1.5, 1, 1.5]$. The first four components of the vector $\boldsymbol{x}$ hold the four bins in P_T of the first η bin, etc. The bin widths are thus given by

$$\begin{array}{c} w_{1,1} = w_{1,5} = w_{1,9} = 2 \\ w_{1,2} = w_{1,6} = w_{1,10} = 3 \\ w_{1,3} = w_{1,7} = w_{1,11} = 5 \\ w_{1,4} = w_{1,8} = w_{1,12} = 10 \end{array} \quad \text{and} \quad \begin{array}{c} w_{2,1} = w_{2,2} = w_{2,3} = w_{2,4} = 1.5 \\ w_{2,5} = w_{2,6} = w_{2,7} = w_{2,8} = 1 \\ w_{2,9} = w_{2,10} = w_{2,11} = w_{2,12} = 1.5 \end{array} \tag{33}$$

and average bin sizes are $\Delta_1 = 5$ and $\Delta_2 = 1.33$. The distances of the bin centres are $[2.5, 4, 7.5]$ along P_t and $[1.25, 1.25]$ along η, respectively, so

$$\begin{array}{c} \delta^1_{2,1} = \delta^1_{6,5} = \delta^1_{10,9} = 2.5 \\ \delta^1_{3,2} = \delta^1_{7,6} = \delta^1_{11,10} = 4 \\ \delta^1_{4,3} = \delta^1_{8,7} = \delta^1_{12,11} = 7.5 \end{array} \quad \text{and} \quad \begin{array}{c} \delta^2_{5,1} = \delta^1_{6,2} = \delta^2_{7,3} = \delta^2_{8,4} = 1.25 \\ \delta^2_{9,5} = \delta^1_{10,6} = \delta^2_{11,7} = \delta^2_{12,8} = 1.25\,. \end{array} \tag{34}$$

The resulting matrix $\mathbf{L}$ for the case of curvature regularisation on the density, including bin width

effects, then looks like

$$\mathsf{L} = \begin{pmatrix} 0.51 & -0.56 & 0.13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.12 & -0.11 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.77 & -0.83 & 0.19 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.18 & -0.17 & 0.03 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.51 & -0.56 & 0.13 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.12 & -0.11 & 0.02 \\ 0.19 & 0 & 0 & 0 & -0.57 & 0 & 0 & 0 & 0.19 & 0 & 0 & 0 \\ 0 & 0.13 & 0 & 0 & 0 & -0.38 & 0 & 0 & 0 & 0.13 & 0 & 0 \\ 0 & 0 & 0.08 & 0 & 0 & 0 & -0.22 & 0 & 0 & 0 & 0.08 & 0 \\ 0 & 0 & 0 & 0.04 & 0 & 0 & 0 & -0.11 & 0 & 0 & 0 & 0.04 \end{pmatrix}, \quad (35)$$

where the numbers have been rounded to two digits.

8 Structure of the TUnfold software package

TUnfold is implemented in the programming language C++ and is interfaced to the ROOT analysis framework. The package is organised in four classes

TUnfold implements the basic unfolding algorithm and L-curve scan.

TUnfoldSys inherits from the TUnfold class and adds functionality to perform background subtraction and propagation of systematic uncertainties.

TUnfoldDensity inherits from the TUnfoldSys class. It adds a method to perform scans of global correlations. More important, it provides support for multidimensional binning schemes, implemented with the help of the class TUnfoldBinning.

TUnfoldBinning is a class to set up binning schemes. The binning schemes are organised in tree-like structures. The nodes of the tree correspond to distinct channels. Each channel may hold a multidimensional distribution in some variables. An example of a binning scheme for the vector $\boldsymbol{x}$ with signal and background bins is shown in figure 2.

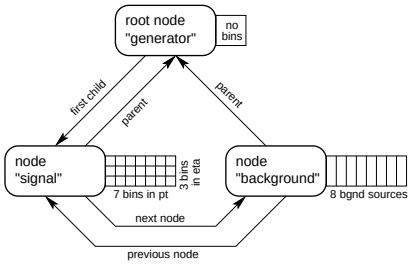

Figure 2: example binning scheme with three nodes. The "generator" node is the root node. It has two child nodes, "signal" and "background". The "signal" node has a two-dimensional binning in two variables, `pt` and `eta`, whereas the background node has unconnected bins corresponding to various background sources.

9 Summary

The mathematical foundations of the TUnfold software package have been presented. TUnfold can be used to correct measurements for migration effects using the well known mathematical techniques of least-square fitting and Tikhonov regularisation. For choosing the strength of the regularisation parameter, two

types of scanning methods are implemented: the L curve method and a flexible minimisation procedure of correlation coefficients. The package offers the possibility to set up non-trivial regularisation schemes for unfolding multi-dimensional distribution. Standard methods to subtract background and to propagate systematic uncertainties are also implemented.

A Partial derivatives used for the propagation of uncertainties

The partial derivatives of A_{ij} with respect to M_{kj} are

$$\frac{\partial A_{ij}}{\partial M_{kj}} = \frac{\delta_{ik} - A_{ij}}{s_j}. \tag{36}$$

The partial derivatives of x with respect to the matrix elements A_{ij} are given by

$$\frac{\partial x_k}{\partial A_{ij}} = C_{kj} z_i - (\mathbf{D^{xy}})_{ki} x_j \qquad \text{where} \tag{37}$$

$$C_{kj} = \begin{cases} E_{kj} & \text{without area constraint} \\ E_{kj} - \frac{(\mathbf{E}\boldsymbol{e})_j (\mathbf{E}\boldsymbol{e})_k}{\boldsymbol{e}^{\mathsf{T}}\mathbf{E}\boldsymbol{e}} & \text{with area constraint} \end{cases} \qquad \text{and} \tag{38}$$

$$z_i = \begin{cases} \left(\mathbf{V_{yy}}^{-1}(\boldsymbol{y} - \mathbf{A}\boldsymbol{x})\right)_i & \text{without area constraint} \\ \left(\mathbf{V_{yy}}^{-1}(\boldsymbol{y} - \mathbf{A}\boldsymbol{x})\right)_i + \frac{\lambda}{2} & \text{with area constraint.} \end{cases} \tag{39}$$

In order to derive this result, the partial derivatives of $\mathbf{E}$ with respect to the elements of the inverse $\mathbf{E}^{-1}$ are expressed by the elements of $\mathbf{E}$,

$$\frac{\partial E_{ij}}{\partial (\mathbf{E}^{-1})_{kl}} = -E_{ik} E_{lj}. \tag{40}$$

The partial derivative of x with respect to the regularisation parameter τ^2 is

$$\frac{\partial x_k}{\partial (\tau^2)} = \begin{cases} \left(\mathbf{E}(\mathbf{L}^{\mathsf{T}}\mathbf{L})(f_b \boldsymbol{x_0} - \boldsymbol{x})\right)_k & \text{without area constraint} \\ \left(\mathbf{E}(\mathbf{L}^{\mathsf{T}}\mathbf{L})(f_b \boldsymbol{x_0} - \boldsymbol{x})\right)_k - \frac{\boldsymbol{e}^{\mathsf{T}}\mathbf{E}(\mathbf{L}^{\mathsf{T}}\mathbf{L})(f_b \boldsymbol{x_0} - \boldsymbol{x})}{\boldsymbol{e}^{\mathsf{T}}\mathbf{E}\boldsymbol{e}} (\mathbf{E}\boldsymbol{e})_k & \text{with area constraint.} \end{cases} \tag{41}$$

B Propagation of systematic uncertainties

Correlated systematic shifts are propagated in the form of systematic shifts of the result. Given a shift $\boldsymbol{\delta}\mathbf{M}$ to the matrix $\mathbf{M}$, one finds the corresponding shift $\boldsymbol{\delta}\mathbf{A}$ of $\mathbf{A}$ using equation 19. The resulting shift on $\boldsymbol{x}$ is then given by

$$\boldsymbol{\delta x} = \sum_{i,j} \frac{\partial \boldsymbol{x}}{\partial A_{ij}} (\boldsymbol{\delta}\mathbf{A})_{ij} = \mathbf{C}(\boldsymbol{\delta}\mathbf{A})^{\mathsf{T}} \boldsymbol{z} - \mathbf{D^{xy}}(\boldsymbol{\delta}\mathbf{A})\boldsymbol{x}, \tag{42}$$

Statistical uncertainties ΔM_{ij} of the elements of $\mathbf{M}$ may also be relevant. The calculation could be done by repeated application of equations 36 and 42 for each independent source of uncertainty ΔM_{ij}. However, the required computing costs are large. In TUnfold, the computation is factorised such that the computing cost is $\mathcal{O}(n^3)$

$$\begin{aligned} (\mathbf{V_{xx}^{M,\mathrm{stat}}})_{ij} = &\sum_k F_{ik} F_{jk} p_k + \sum_k C_{ik} C_{jk} \sum_l Q_{lk} z_l^2 + \sum_k D_{ik}^{xy} D_{jk}^{xy} \sum_l Q_{kl} x_l^2 \\ &- (\mathbf{F}\mathbf{G}^{\mathsf{T}} + \mathbf{G}\mathbf{F}^{\mathsf{T}})_{ij} - (\mathbf{D^{xy}}\mathbf{H}^{\mathsf{T}} + \mathbf{H}(\mathbf{D^{xy}})^{\mathsf{T}})_{ij} \quad \text{where} \end{aligned} \tag{43}$$

$$Q_{ij} = \left(\frac{\Delta M_{ij}}{s_j}\right)^2 \text{ and } p_j = \sum_{i=0}^{n} Q_{ij}, \tag{44}$$

$$F_{ij} = \sum_k \frac{\partial x_i}{\partial A_{kj}} A_{kj} = C_{ij}(\mathbf{A}^\mathsf{T}\boldsymbol{z})_j - (\mathbf{D^{xy}A})_{ij} x_j, \tag{45}$$

$$G_{ij} = \sum_k \frac{\partial x_i}{\partial A_{kj}} Q_{kj} = C_{ij}(\mathbf{Q}^\mathsf{T}\boldsymbol{z})_j - (\mathbf{D^{xy}Q})_{ij} x_j, \tag{46}$$

$$H_{ij} = z_j \sum_k C_{ik} x_k Q_{jk}. \tag{47}$$

References

[1] R. Brun *et al.*, CERN-DD/EE-84-1 (1987).

[2] T. Sjostrand, P. Eden, C. Friberg, L. Lonnblad, G. Miu, S. Mrenna and E. Norr bin, "High-energy physics event generation with PYTHIA 6.1," Comput. Phys. Commun. **135**, 238 (2001) [hep-ph/0010017].

[3] S. Schmitt, TUnfold version 17.0, `http://www.desy.de/~sschmitt/tunfold.html`.

[4] A. N. Tikhonov, Soviet Math. Dokl. **4** (1963), 1035.

[5] R. Brun and F. Rademakers, Nucl. Instrum. Meth. A **389** (1997) 81.

[6] A. Hocker and V. Kartvelishvili, Nucl. Instrum. Meth. A **372** (1996) 469 [hep-ph/9509307].

[7] G. D'Agostini, Nucl. Instrum. Meth. A **362**, 487 (1995).

[8] G. D'Agostini, arXiv:1010.0632.

[9] G. Choudalakis, arXiv:1201.4612 [physics.data-an].

[10] V. Blobel, arXiv:hep-ex/0208022.

[11] V. B. Anykeev, A. A. Spiridonov and V. P. Zhigunov, Nucl. Instrum. Meth. A **303** (1991) 350.

[12] V. Blobel, proceedings of the PHYSTAT 2011 workshop, Eds. H. B. Prosper, L.Lynons, Geneva (2011) 240.

[13] Glen Cowan, statistical data analysis, Oxford University Press (1998), ISBN 0198501560

[14] V. Blobel, Unfolding methods in high energy physics experiments, in Proceedings of the 1984. CERN School of Computing, CERN 85-09 (1985).

[15] V. Blobel, "Data unfolding", talk given at the Terascale statistics School, Hamburg (2010) `http://www.desy.de/~blobel/`.

[16] P. C. Hansen, The L-curve and Its Use in the Numerical Treatment of Inverse Problems, Computational Inverse Problems in Electrocardiology, ed. P. Johnston (2000).

[17] F. D. Aaron *et al.* [H1 Collaboration], Eur. Phys. J. C **66** (2010) 17 [arXiv:0910.5631].

[18] C. Adloff *et al.* [H1 Collaboration], Eur. Phys. J. C **13** (2000) 609 [hep-ex/9908059].

[19] Amnon Harel, private communication (March 2011).