Scaffolding Simulations with Deep Learning for High-dimensional Deconvolution

Anonymous authors
Paper under double-blind review

Abstract

A common setting for scientific inference is the ability to sample from a high-fidelity forward model (simulation) without having an explicit probability density of the data. We propose a simulation-based maximum likelihood deconvolution approach in this setting called OMNIFOLD. Deep learning enables this approach to be naturally unbinned and (variable-, and) high-dimensional. In contrast to model parameter estimation, the goal of deconvolution is to remove detector distortions in order to enable a variety of down-stream inference tasks. Our approach is the deep learning generalization of the common Richardson-Lucy approach that is also called Iterative Bayesian Unfolding in particle physics. We show how OMNIFOLD can not only remove detector distortions, but it can also account for noise processes and acceptance effects.

1 Introduction

In many scientific applications, an important step in data analysis is the mitigation of detector distortions. When this is done at the level of an individual datum, this process is known as *calibration* while when it is done at the level of an entire dataset, it is known as *deconvolution*. If detector distortions are known analytically, then one can perform maximum likelihood estimation to remove detector effects: $\hat{x}_T = \arg\max_{x_G} p_{S|G}(x_D|x_G)$, where T, D, S, and G stand for truth, data, simulation, and generation, respectively. The data (D) are observed in a detector and we want to infer the pre-detector truth (T). Our model of this process uses synthetic pre-detector generation (G) and post-detector simulation (S). We often do not know $p_{S|G}$ analytically, especially when each data point is in a high- (and possibly variable-) dimensional space. However, we often do have detailed simulations of our experimental devices and can sample from $p_{S|G}$.

Traditional deconvolution methods convert the problem of estimating $\hat{x}_T$ into a linear algebra problem. If the data features are discretized into a finite number of histogram bins, then the entire dataset $x = \{x_1, ..., x_n\}$ can be represented as a vector where the entries are the histogram bin contents. The conditional density $p_{S|G}$ is then a matrix and can be estimated numerically by sampling. A variety of regularized matrix inversion methods have been proposed and one of the most popular methods across scientific domains is the Richardson-Lucy (RL) (Richardson (1972); Lucy (1974)) approach that is also called Iterative Bayesian Unfolding (D'Agostini (1995)) in particle physics. The RL algorithm is an iterative approach where at each step, $p_{G|S}$ is estimated from $p_{S|G}$ and the estimate of $\hat{x}_T$ from the previous step, and then this matrix is applied to x_D to arrive at a new estimate of $\hat{x}_T$. This approach can be phrased in terms of an expectation-maximization (EM) algorithm and correspondingly proven to converge to the maximum likelihood estimate (Shepp & Vardi (1982)).

While RL is a widely used algorithm, it has three key challenges. First, it requires the data to be represented in a fixed number of bins. This discretization is fixed at the beginning of the analysis and cannot be changed later (except to make it coarser). Second, it is often impractical to add more dimensions and so typically strong dimensional reduction proceeds the deconvolution. Third, because the number of dimensions are limited, the deconvolution cannot benefit from all possible auxiliary features that control the detector distortions.

Deep learning has the potential to solve all three of these challenges. We will introduce the OMNIFOLD algorithm (Anonymous (2020)), which is an iterative approach based on likelihood-ratio estimation with deep learning classifiers. A variety of other machine learning approaches have also been proposed, but they cannot process unbinned data (Gagunashvili (2010); Glazov (2017);

Datta et al. (2018); Bunse et al. (2018)) or do not have desirable statistical properties (e.g. prior independence via maximum likelihood estimation) (Bellagente et al. (2020b;a)). This paper is organized as follows. Section 2 introduces the OMNIFOLD algorithm and extends the approach beyond Ref. (Anonymous (2020)) by accounting for noise processes and acceptance effects. Numerical examples are presented in Sec. 3 and the paper ends with brief conclusions and outlook in Sec. 4.

2 THE OMNIFOLD ALGORITHM

A complete deconvolution algorithm must be able to account for four effects:

Part (1) Noise processes. In many cases, the data is a coherent superposition of a signal process and a background process. This background process must be statistically subtracted before removing detector distortions.

Part (2) Detector acceptance. The detector elements may not capture all signal process examples due to finite thresholds and other acceptance effects.

Part (3) Detector distortions. This is the classical convolution of the data with a noise function that must be statistically removed.

Part (4) Detector efficiency. Sometimes there are also truth-level thresholds so that some synthetic signal examples register synthetic detector signals but are not recorded as synthetic generated examples.

Binned approaches like RL can account for the above affects via

$$\hat{x}_T = p_{S|G}^{-1} \left((x_D - x_N) \odot \mathrm{Pr}_{S \leftarrow G} \right) \odot \mathrm{Pr}_{G \rightarrow S}^{-1}, \tag{1}$$

where $\odot$ is the Hadamard (component-wise) product, $p_{S|G}^{-1}$ represents regularized matrix inversion (Part (3)), x_N is a binned estimate of the noise process to be removed (Part (1)), $\mathrm{Pr}_{S \leftarrow G}$ represents the fraction of simulation-level examples that have a corresponding generator-level example (Part (2)), and $\mathrm{Pr}_{G \rightarrow S}$ represents the fraction of generator-level examples that have a corresponding simulation-level example (Part (4)). The vector inverse $\mathrm{Pr}_{G \rightarrow S}$ is computed component-wise.

The OMNIFOLD method achieves Steps (1)–(4) without binning using a series of binary, weighted classification tasks. In particular, let $x_{D,i}$ be examples from data, $(x_{G,i}, x_{S,i})$ be synthetic examples of the signal process, and $x_{N,i}$ are noise examples. For the signal process, if one of $x_{G,i}$ or $x_{S,i}$ does not exist (acceptance or efficiency effects), it is set to a fixed dummy value $\emptyset$. It is sometimes the case that the examples $(x_{G,i}, x_{S,i})$ and $x_{N,i}$ represent more or fewer than one real event from data and so they come with weights $w_{\mathrm{synth},i}$ and $w_{\mathrm{noise},i}$, respectively. Let g be parameterized as a deep neural network. Then define,

$$f(A, W_A; B, W_B) = \arg\max_g \left(\sum_{a \in A} w_a \log g(a) + \sum_{b \in B} w_b \log(1 - g(b)) \right), \tag{2}$$

where A, B, W_A, W_B are sets and W_A, W_B correspond to sets of weights for the examples in A and B, respectively. When the weights w are all unity, then the neural networks f are trained using the standard binary cross entropy loss functions. Finally, define $\tilde{f}(A, W_A; B, W_B) = f/(1 - f)$. It is well-known (Hastie et al. (2001); Sugiyama et al. (2012)) that when f is sufficiently flexible and well-trained, $\tilde{f}$ approximates the weighted likelihood ratio of x given A or B. Any other loss function that results in a (known) monotonic rescaling of the likelihood ratio would also work instead of Eq. 2.

With all of these components, the OMNIFOLD protocol is as follows:

OMNIFOLD: *(extending Ref. (Anonymous (2020)) by including Parts (1), (2), (4))*

(i) $w_{D,i} = \tilde{f}(\{x_{D,i}\} \cup \{x_{N,i}\}, \{1,...,1\} \cup \{-w_{\text{noise},i}\}; \{x_{D,i}\}, \{1,...,1\})(x_{D,i})$

(ii) For k in $N_{\text{iterations}}$ do:

  (a) $w_{\text{Step I},i} = \tilde{f}(\{x_{S,i}|x_{S,i} \neq \emptyset\}, \{w_{\text{synth},i}|x_{S,i} \neq \emptyset\}; \{x_{D,i}\}, \{w_{D,i}\})(x_{S,i})$

  (b) $\tilde{f}_{\text{miss},\text{I}} = \tilde{f}(\{x_{G,i}|x_{S,i} \neq \emptyset\}, \{w_{\text{Step I},i}|x_{S,i} \neq \emptyset\}; \{x_{G,i}|x_{S,i} \neq \emptyset\}, \{1,...,1\})$

  (c) $w_{\text{Pull},i} = w_{\text{Step I},i}$ if $x_{S,i} \neq \emptyset$ and $w_{\text{Pull},i} = \tilde{f}_{\text{miss},\text{I}}(x_{G,i})$ otherwise

  (d) $w_{\text{Step II},i} = \tilde{f}(\{x_{G,i}\}, \{w_{\text{Pull},i}\}; \{x_{G,i}\}, \{w_{\text{synth},i}\})(x_{G,i})$

  (e) $\tilde{f}_{\text{miss},\text{II}} = \tilde{f}(\{x_{S,i}|x_{G,i} \neq \emptyset\}, \{w_{\text{Step II},i}|x_{G,i} \neq \emptyset\}; \{x_{S,i}|x_{G,i} \neq \emptyset\}, \{1,...,1\})$

  (f) $w_{\text{Push},i} = w_{\text{Step II},i}$ if $x_{G,i} \neq \emptyset$ and $w_{\text{Push},i} = \tilde{f}_{\text{miss},\text{II}}(x_{S,i})$ otherwise

  (g) $w_{\text{synth},i} = w_{\text{Push},i}$

The final result is the set $\{x_{G,i}|x_{G,i} \neq \emptyset\}$ with weights $\{w_{\text{synth},i}|x_{G,i} \neq \emptyset\}$, from which one can compute empirical probability densities, expectation values, etc. of weighted statistics.

The intuition for the various steps are as follows. OMNIFOLD (i) achieves Part (1) through positive reweighting (Nachman & Thaler (2020)). The core parts of the iterative process are (ii), (a), (c), (d), and (f). Ref. (Anonymous (2020)) proved that these steps can be modeled as an EM-type algorithm and achieve the maximum likelihood estimate. Step I weights ensure that modified detector-level simulation are statistically identical to the measured data. The pairing between the pre- and post-detector data in simulation allow the Step I weights to be associated ('pulled') to the pre-detector features. These weights are not sufficient because they are not a proper function of the pre-detector features since the same example can be mapped to different features through the stochastic mapping of the detector. This requires the Step II weighting, which produces a proper function of the pre-detector features. These weights can be associated ('pushed') with the post-detector features and the entire process can be repeated. OMNIFOLD (ii), (b) and (e) are required to handle acceptance and efficiency effects, respectively. For examples that are missing pre- or post-detector features, we use positive reweighting (Nachman & Thaler (2020)) again to assign the average weight. An alternative choice would be to assign a weight of unity, which would mean that the prior is used for these examples. When there is no noise process and no efficiency/acceptance effects, (i), (ii), (b), and (ii), (e) can be eliminated. Finally, note that when the data x_i are binned, OMNIFOLD approximates the RL algorithm step by step.

3 NUMERICAL EXAMPLES

To illustrate all of the deconvolution parts and the various OMNIFOLD steps, we simulate a one-dimensional Gaussian. In particular, $X_T \sim \mathcal{N}(0.2, 0.8)$, $X_G \sim \mathcal{N}(0, 1)$, and $X_D = X_T + Z$, $X_S = X_G + Z$ where $Z \sim \mathcal{N}(0, 0.5)$. There is a background noise process that is distributed as $\mathcal{N}(0, 1.2)$ that occurs in 10% of the measured examples. Furthermore, 10% of the measured values have no corresponding truth value and vice versa. There are 10^5 data and simulation examples.

The neural networks are all implemented in TENSORFLOW 1.15.0 (Abadi et al. (2016)) and optimized using ADAM (Kingma & Ba (2014)) with default values unless otherwise specified. These networks are fully connected with three hidden layers with 50 nodes each and the rectified linear unit between hidden layers and a sigmoid as the output. The networks were trained for 200 epochs with a batch size of 2000 and early stopping using a patience of 10 epochs.

Figure 1 presents the results of OMNIFOLD with three iterations. After three iterations, the results are nearly unchanged. The unfilled histogram in the left plot of Fig. 1 shows OMNIFOLD (i), which agrees well with the noiseless data (dark grey). This comparison would not be possible in practice as the noiseless data are not available. The outcome of OMNIFOLD (ii) is shown at detector-level on the left and prior to detector-effects on the right. The orange dashed line (detector-level) agrees well with the black unfilled histogram (left) and the black unfilled histogram on the right agrees well with the green filled histogram. The data are binned for illustration, but the result is naturally unbinned.
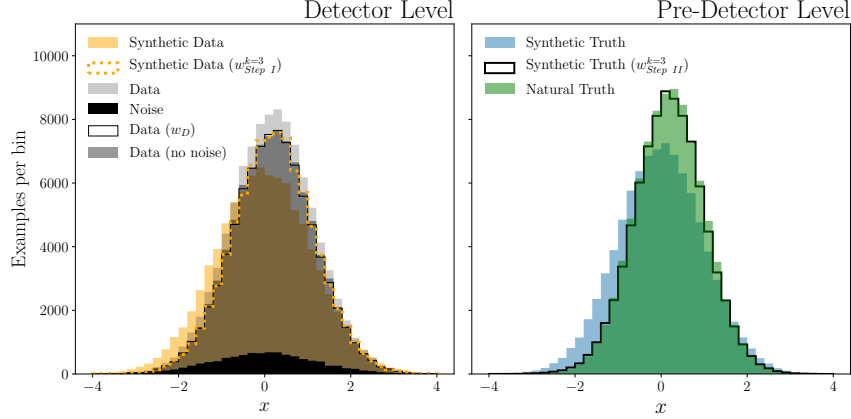
Figure 1: The complete OMNIFOLD protocol for the one-dimensional Gaussian example. Histograms on the left (right) are shown after (before) detector distortions. The final result after three iterations is indicated with the weights $w_{\text{Step I}}^{k=3}$ on the left and $w_{\text{Step II}}^{k=3}$ on the right.

iterations → N	$\bar{x}$ mean ($\times 10^2$)				$\bar{x}$ standard deviation ($\times 10^3$)			
	1	2	4	8	1	2	4	8
1	21.62(8)	25.13(8)	28.12(8)	29.67(8)	8.4(5)	8.3(6)	8.0(7)	7.9(7)
2	28.54(5)	29.24(6)	29.88(6)	**30.06(5)**	5.3(4)	5.6(4)	5.2(4)	4.8(4)
3	29.54(4)	**29.91(4)**	**30.02(4)**	**30.00(4)**	3.6(3)	4.4(4)	3.6(3)	4.0(3)
4	29.89(3)	**30.01(3)**	**30.01(3)**	**30.01(3)**	**3.2(2)**	**2.8(2)**	**3.1(2)**	**3.1(2)**
5	**30.04(3)**	**30.00(3)**	**29.99(4)**	**30.06(3)**	3.5(3)	**3.1(2)**	3.8(3)	**3.1(2)**

Table 1: The average and standard deviation of the mean $\bar{x}$ of the unfolded $\hat{X}_T$ in the multidimensional Gaussian example. Uncertainties are determined over 100 identical experiments. Bolded values are within 2σ of the correct value (left) or are consistent with the smallest uncertainty (right).

OMNIFOLD can readily process multidimensional data. Table 1 presents the results of experiments conducted for $X_T \sim \mathcal{N}(0.3, 0.5)$, $X_G \sim \mathcal{N}(0, 1)$ and $X_D = X_T + \sum_{i=1}^{4} Z_i$, $X_S = X_G + \sum_{i=1}^{4} Z_i$ where $Z_i \sim \mathcal{N}(0, 1)$. There are no noise, efficiency, or acceptance effects. We consider 5 scenarios where the deconvolution is performed using only X_D and X_S, $X_D, X_S, Z_0, ...,$ and $X_D, X_S, Z_0, ..., Z_4$. For the five-dimensional case, there are actually no resolution effects since X_T can be uniquely determined from X_D and the Z_i. The table shows that more features leads to better accuracy and precision for fewer iterations of the algorithm.

By using classifiers instead of ratios of density estimators for the weighting steps, we can also make use of advances in classification networks to handle complex and structured data. For example, deep sets (Zaheer et al. (2018); Komiske et al. (2019)) was used for a permutation-invariant and variable-length example from collider physics in Ref. (Anonymous (2020)), excluding background noise and acceptance/efficiency effects.

4   CONCLUSIONS AND OUTLOOK

In this paper, we have introduced the OMNIFOLD method, which scaffolds a forward simulation with deep learning in order to perform unbinned and high-dimensional deconvolution. We have extended the framework from Ref. (Anonymous (2020)) to include the full suite of experimental effects that are encountered in practice and we have demonstrated this approach using a Gaussian example. Our code is available at `https://github.com/hep-lbdl/OmniFold` and can be deployed for a wide variety of scientific applications ranging from differential cross section measurements in high energy physics to quantum computing (Nachman et al. (2020)).

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Anonymous. OmniFold: A Method to Simultaneously Unfold All Observables. *Phys. Rev. Lett.*, 124(18):182001, 2020. doi: 10.1103/PhysRevLett.124.182001.

Marco Bellagente, Anja Butter, Gregor Kasieczka, Tilman Plehn, Armand Rousselot, Ramon Winterhalder, Lynton Ardizzone, and Ullrich Köthe. Invertible Networks or Partons to Detector and Back Again. *SciPost Phys.*, 9:074, 2020a. doi: 10.21468/SciPostPhys.9.5.074.

Marco Bellagente, Anja Butter, Gregor Kasieczka, Tilman Plehn, and Ramon Winterhalder. How to GAN away Detector Effects. *SciPost Phys.*, 8(4):070, 2020b. doi: 10.21468/SciPostPhys.8.4.070.

Mirko Bunse, Nico Piatkowski, Tim Ruhe, Wolfgang Rhode, and Katharina Morik. Unification of deconvolution algorithms for Cherenkov astronomy. In *5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 21–30. IEEE, 2018.

G. D'Agostini. A Multidimensional unfolding method based on Bayes' theorem. *Nucl. Instrum. Meth.*, A362:487–498, 1995. doi: 10.1016/0168-9002(95)00274-X.

Kaustuv Datta, Deepak Kar, and Debarati Roy. Unfolding with Generative Adversarial Networks. 2018.

Nikolai D. Gagunashvili. Machine learning approach to inverse problem and unfolding procedure. 2010.

Alexander Glazov. Machine learning as an instrument for data unfolding. 2017.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019. doi: 10.1007/JHEP01(2019)121.

L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79:745, June 1974. doi: 10.1086/111605.

B. Nachman, M. Urbanek, W. de Jong, and C. Bauer. Unfolding Quantum Computer Readout Noise. *npj Quantum Information*, 6, 2020. doi: 10.1038/s41534-020-00309-7.

Benjamin Nachman and Jesse Thaler. Neural resampler for Monte Carlo reweighting with preserved uncertainties. *Phys. Rev. D*, 102(7):076004, 2020. doi: 10.1103/PhysRevD.102.076004.

William Hadley Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1):55–59, Jan 1972. doi: 10.1364/JOSA.62.000055. URL http://www.osapublishing.org/abstract.cfm?URI=josa-62-1-55.

L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE transactions on medical imaging*, 1(2):113–122, 1982. doi: 10.1109/TMI.1982.4307558.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012. doi: 10.1017/CBO9781139035613.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018.