Appendix B

Horizontal Weighting

The current appendix addresses the issue of manipulating some input distribution, *source*, such that it becomes statistically compatible with a given, *target*, distribution. A common method to do that involves binning the two distributions. This method is sometimes called *vertical weighting* and it is described in the following section. The main motivation behind the technique advertised in the current appendix is to circumvent problems arising from binning distributions, especially with increasing number of bins or in case of small sample sizes. Note that the bigger the number of bins the more precise the matching of the two distributions is. The advertised technique is described in the current appendix and a choice for its name could be *horizontal weighting*. The technique is inspired by discussions with Gerhard Raven and Diego Martinez Santos.

Lastly a matching example is given at the of the current appendix. This example is a typical problem in high energy physics originating from non perfect simulation. Specifically certain kinematic distributions might differ significantly between simulated events and data. Given that simulated events are commonly used to control acceptance, resolution or other detector effects; what is typically done is to correct the simulated data sample by matching its kinematic distributions to the ones observed in data. An example of a situation where this problem arises can be found in Section 4.2.3.

Vertical Weighting

The vertical weighting approach to matching distributions is straightforward. The first step is to bin the source (S) and target (T) distributions with the same binning scheme. After that a weight, w_i, is assigned to each entry in a given bin, i, such that the bin contents of the source distributions matches these of the target, see Eq. B.1. Essentially this technique moves vertically each bin of the source distribution in order to match the bin contents of the target distribution.

$$w_i = \frac{T_{\text{bin}i}}{S_{\text{bin}i}} \tag{B.1}$$

The advantages of the vertical weighting approach is that it is easy to understand and implement. However, there are some disadvantages that result from the binning itself. For example it can happen that a given source or target bin has zero entries for a given binning scheme. This situation becomes more pronounced in the case of large number of bins, which as already mentioned improves the precision of the matching. In addition it can also happen that any of the distributions is weighted and the sum of weights in a given bin is negative. Both of the above situation require some justification as to how these problematic cases can be handled. In addition, it can also happen that the source to target matching needs to be done in many dimensions, as mentioned in the introduction of the current appendix. In that case the number of bins increase rapidly, and thus the number of problematic bins as well, to the point that it is no longer possible to match the source distributions to the target one. Note that variables corresponding to these dimensions are in general correlated with each other. Hence, doing several one dimensional weighting steps will simply ignore these correlations.

Horizontal Weighting

The horizontal approach to matching distributions is meant to bypass the problem of binning, especially in many dimensions, and thus make it possible to match an arbitrary number of variables between source and target. The basic idea of the approach is to apply as chain of transformations to both source and target distributions, such that they become uncorrelated, see Eq. B.2a and Eq. B.2b respectively. Subsequently the transformation chain that has been applied to the target distributions are inverted and then applied to the source ones, Eq. B.2c. The three transformations involved in the current approach are: Transformation (A) converts the input distribution to flat. Transformation (B) converts a flat distribution to a normal distribution while (C) removes the correlation between two distributions. The necessary mathematical tools to perform the above transformations are presented in the next paragraph.

$$S \times A_{\mathrm{S}} \rightarrow S_{\mathrm{flat}} \times B_{\mathrm{S}} \rightarrow S_{\mathrm{gaus}} \times C_{\mathrm{S}} \rightarrow S_{\mathrm{uncor\ gaus}} \tag{B.2a}$$

$$T \times A_{\mathrm{T}} \rightarrow T_{\mathrm{flat}} \times B_{\mathrm{T}} \rightarrow T_{\mathrm{gaus}} \times C_{\mathrm{T}} \rightarrow T_{\mathrm{uncor\ gaus}} \tag{B.2b}$$

$$S_{\mathrm{matched}} = S_{\mathrm{uncorgaus}} \times C_{\mathrm{T}}{}^{-1} \times B_{\mathrm{T}}{}^{-1} \times A_{\mathrm{T}}{}^{-1} \tag{B.2c}$$

The transformation steps, A and B, that appear in the above logical steps make use of well known mathematical theorems, namely *Inverse Transformation Sampling* and *Probability Integral Transform*. No prof of the above theorems is included since the intention of the current appendix is to quickly demonstrate the advertised technique. However, there is plethora of examples online.

Theorem 1 (Inverse Transformation Sampling) *Let u be a uniformly distributed variable. Consider another random variable x that has a Cumulative Distribution Function (CDF), call it F_x. Then the variable $x' = F_x^{-1}(u)$ is distributed the same way as x does.*

Theorem 2 (Probability Integral Transform) *Let x be a random variable distributed according to a PDF, $P(x)$. Let F_x be the CDF of $P(x)$. Then the variable $u = F_x(x)$ is uniform. This theorem is the inverse of Theorem 1.*

For completion the CDF, $F_x(y)$, of a certain PDF, $P(x)$, where y takes its values from the same domain as x, is by definition:

$$F_x(y) = \int_{-\infty}^{y} P(t)dt = \text{Probability}(x \leq y) \tag{B.3}$$

Both of the above theorems can only be directly applied to continuous distributions. However, exact analytic shape of the PDFs or CDFs involved is not required. This is because the CDF of any variable x can be built by essential computing the integral-sum of Eq. B.3. In order to do that x is binned so that the cumulative sum of each bin can be computed. This is a straightforward step, where the number of entries in each bin are added to the number of entries of the next bin until the range of x is exhausted. It is important to point out that this binning can be arbitrarily fine without having the problems explained in the introduction, since it is not used to much any distribution. However, from implementation point of you the larger the number of bins the slower the algorithm performs. (Here is where the built-in function `numpy.digitise()` of `python` proves to be useful, as the timing scales nicely with the number of bins.)

The transformations A and B mentioned in Eq. B.2, are direct implementations of Theorem 2. Similarly the inverted versions, A^{-1} and B^{-1}, of the previous transformations are direct implementations of Theorem 1. The only deference between is case of the transformations B and B^{-1} the shapes of the PDF and CDF involved are known analytically. The previous transformations involve the *inverse error function* and the *complementary error function* respectively, which are both well known distributions.

Coming now to the last transformation step, C, of Eq. B.2. This step essentially performs a linear transformation to two correlated Gaussian distributed variables such that they become uncorrelated. The method followed is essentially a standard *Matrix diagonalization*; The relevant steps required are described in Method 1

Method 1 *Let $\vec{x}$ be a set of correlated variables, with a corresponding covariance matrix C. Let P^{-1} be the matrix that has the eigen-vectors* *of C as columns. The set of values $\vec{x'} = P\vec{x}$ is an uncorrelated set of $\vec{x}$. The set of values $\vec{x} = P^{-1}\vec{x}$ is the corresponding correlated set of $\vec{x'}$.*

The covariance between variables is computed using the standard formula of:

$$c_{ij} = \frac{1}{N}\sum(x_i - \hat{x_i})(x_j - \hat{x_j}), \text{ with } \hat{x} = \frac{1}{N}\sum x \tag{B.4}$$

Finding the eigen-vectors of C is a bit more lengthy to quote in the current appendix. Nevertheless, it is a straightforward well known problem for which there are many implemented algorithms as well.

Lastly an important implementation issues is clarified. Specifically, the CDF is built by binning a given distribution, x, and associating the a value of the integral of Eq. B.3, name it c_i, to each bin x_i. In that case the CDF is just an $x_i \rightarrow c_i$ *map* structure, implying that the $x \rightarrow c$ function is not continuous by definition. Of the binning can be increased but one could argue that this is more brute force that solving the problem. The way to make the $x \rightarrow c$ mapping continues is using the well known technique of *linear interpolation*, shown in Figure B.5, and thus solving the first of the two implementation issues.

$$c(x) = v_i + \frac{c_{i+1} - c_i}{x_{i+1} - x_i} * (x - x_i) \tag{B.5}$$

The ingredients necessary to apply the horizontal weighting technique presented have been covered.

Example-Discussion

To demonstrate the advertised technique a typical problem in high energy physics is addressed. Specifically, the K^{*0} particle from the $\mathrm{B}_s^0 \rightarrow \mathrm{J}/\psi\,\overline{\mathrm{K}}^{*0}$ mode decays into a K and a π. The momenta distributions of these two particles ($p(\mathrm{K}) - p(\pi)$) is found to defer between the

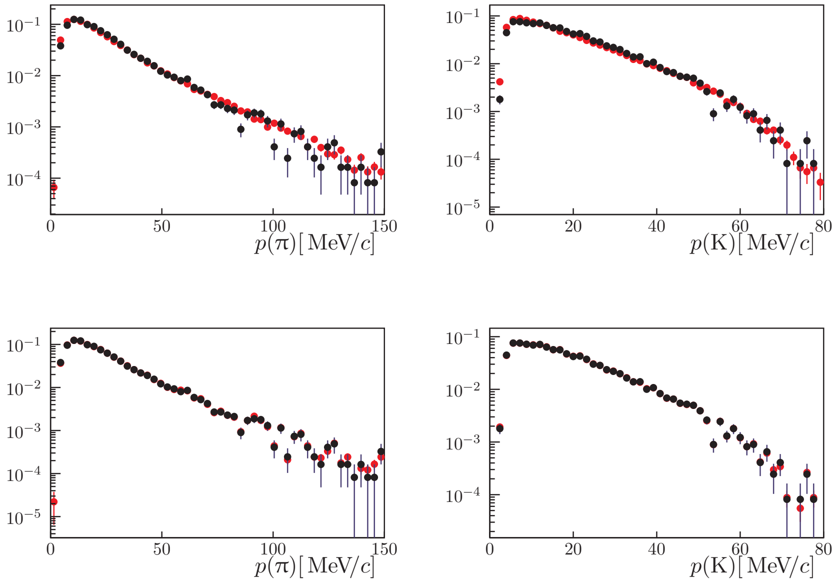

FIGURE B.1: Comparison before and after matching. Source (Target) distributions are shown in red (blue) color. The upper (lower) two distributions are the original (matched) distributions.

simulation sample (source) and background subtracted data (target). Furthermore, the previous distributions are correlated and combined with a fine binning will yield problems as explained earlier in the vertical weighting subsection.

distribution	KS before matching	KS after matching
p (K)	10^{-9}	0.998
p (π)	10^{-25}	1.000

TABLE B.1: KS test between source and target for each of the two distributions (p(K) – $p(\pi)$). Better agreement is achieved after matching.

After applying the advertised technique the $(p(\mathrm{K}) - p(\pi))$ distributions become statistically compatible, avoiding the risks associated to a multidimensional fine binning scheme. The matched distributions can be seen at the bottom of Figure B.1. In addition, a Kolmogorov–Smirnov test is performed to quantify the matching of each source-target distribution. The results are summarized in Table B.1.