Version
October 11, 2013

Training NeuroBayes® with sPlot weights

Phi-T Physics Information Technologies GmbH

1 Motivation

Let's suppose a sample of N events each belonging to one of n_{sp} species (e.g. signal or background events). For each event a set of variables $\vec{v} = (v_1, \ldots, v_{n_{var}})$ is measured. Given that, the distributions of these variables (including the target distribution) contain contributions of all species and can not be used the train Neurobayes. A sideband-subtraction allows to remove the background events in the signal region by adding background events from sidebands which carry negative weights. If the yields are estimated correctly these events statistically compensate all background events found below the signal. Hence, the sideband-subtracted distributions can be used to train Neurobayes without referring to any simulated signal events. The sPlot formalism [1] is a advanced sideband subtraction technique which provides binwise sideband subtraction. More generally speaking: sPlot is a statistical tool to unfold data distributions. Furthermore correct normalization and statistical uncertainties are provided by the sPlot formalism.

2 Sideband Subtraction in the sPlot formalism

In the simple case where the target variable y is split in $n_y = 2$ regions and the number of species n_{sp} also equals 2, determining the sWeights is obvious. Let's consider a data sample consisting of signal and background events distributed according to the PDFs:

$$\begin{aligned} f_s &= \begin{cases} 0 & y \leq y_0 \\ \frac{1}{y_{max}-y_0} & y_0 < y \leq y_{max} \end{cases} \\ f_b &= \frac{1}{y_{max}} \,. \end{aligned}$$

In Figure 1 the PDFs and different regions are illustrated. One can easily express the number of signal and background events (N_s, N_b) by the number of events in regions $R_l : [0, y_0]$ and $R_r : [y_0, y_{max}]$:

$$\begin{aligned} N_l &= N \int_0^{y_0} f(y)dy = \int_0^{y_0} \left(N_s f_s(y) + N_b f_b(y)\right) dy = N_b \int_0^{y_0} f_b(y)dy = N_b \frac{y_0}{y_{max}} \qquad (1) \\ N_r &= N \int_{y_0}^{y_{max}} f(y)dy = (y_{max} - y_0)\left(N_s \frac{1}{y_{max}-y_0} + N_b \frac{1}{y_{max}}\right) = N_s + N_b\left(1 - \frac{y_0}{y_{max}}\right) \,. \end{aligned}$$

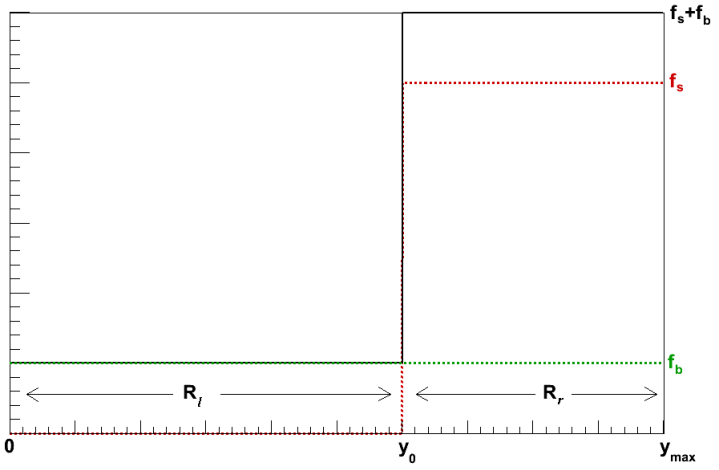

Figure 1: PDFs for signal and background

Solving Equation 1 for N_s and N_b leads directly to the weights for signal and background in the regions R_l and R_r:

$$\begin{aligned} N_b &= \frac{y_{max}}{y_0} N_l = w_b^{(l)} \cdot N_l + w_b^{(r)} \cdot N_r \\ N_s &= -\frac{y_{max} - y_0}{y_0} N_l + N_r = w_s^{(l)} \cdot N_l + w_s^{(r)} \cdot N_r \end{aligned}$$

The weight for events in the region R_α of species i is w_i^α:

$$\begin{aligned} w_b^{(l)} &= \frac{y_{max}}{y_0} \\ w_s^{(l)} &= -\frac{y_{max} - y_0}{y_0} \\ w_b^{(r)} &= 0 \\ w_s^{(r)} &= 1 \end{aligned}$$

The signal distribution for a training variable x, for example, is reproduced by assigning weights to all events. Events in the region R_l receive the weight $w_s^{(l)}$ whereas the rest of the events, which all belong to the region R_r , are weighted with $w_s^{(r)}$. The calculated weight w_i^α is equal to the sWeight ${}_s\mathcal{P}_i(y_\alpha)$[1].

The above derivation works for the general case where the number of species equals the number of regions ($n_{sp} = n_y$). In this case the matrix equation corresponding to equation 1 for a histogram with n_y bins is:

$$N^\alpha = \sum_{i=1}^{n_{sp}} N_i F_i^\alpha, \tag{2}$$

where

- N^α is the number of events in the bin α
- N_i is the number of events for species i
- $F_i^\alpha = \int_{bin\ \alpha} f_i(y)dy$.

Now, N^α is inverted:

$$N_i = \sum_{\alpha=1}^{n_y} N^\alpha \left(F^{-1}\right)_i^\alpha = \sum_{\alpha=1}^{n_y} N^\alpha w_i^\alpha. \tag{3}$$

Thus the elements of F^{-1} are the weights w_i^α needed to reconstruct the PDFs of the different species.

3 sPlot formalism

The above procedure does not apply if the number of bins is greater than the number of species ($n_y > n_{sp}$) because Equation 2 is not sufficient to determine N_i. Nevertheless the yields N_i can be determined by an extended maximum Likelihood fit to the target distribution. The log-Likelihood function to be minimized is

$$\mathcal{L} = \sum_{e=1}^{N} \left(\ln \left\{ \sum_{i=1}^{n_{sp}} N_i f_i(y_e, \theta_i) \right\} \right) - \sum_{i=1}^{n_{sp}} N_i, \tag{4}$$

where

- N is the number of events in the sample
- n_{sp} is the number of species
- N_i is an estimate for the number of events for the ith species
- y_e is the value of the target variable y for event e
- f_i is the PDF of the target variable y for the ith species
- θ_i is the parameter set of f_i.

For the target variable the PDF for each species f_i has to be known whereas the yields N_i and the parameter set θ_i are determined by the fit. Now, the sWeight ${}_s\mathcal{P}_n$ for a given event e reads :

$${}_s\mathcal{P}_n(y_e) = \frac{\sum_{i=1}^{n_{sp}} \mathbf{V}_{ni} f_i(y_e)}{\sum_{k=1}^{n_{sp}} N_k f_k(y_e)} \, . \tag{5}$$

where the covariance matrix $\mathbf{V}_{ni}$ can be calculated by inverting

$$\mathbf{V}_{ni}^{-1} \;=\; \sum_{events} \frac{f_i(y_e) f_n(y_e)}{\left(\sum_k N_k f_k(y_e)\right)^2} \, .$$

Alternativley to the likelihood fit, the yields N_i and the parameter set θ_i can be determined by a χ^2 fit.

3.1 Normalization and Statistical Uncertainties

To check the validity of the calculated sWeights the following normalization properties are of great use:

$$\sum_{i=1}^{n_{sp}} {}_s\mathcal{P}_i(y_e) \;=\; 1 \;\forall \text{ event } e \tag{6}$$

$$\sum_{e=1}^{N} {}_s\mathcal{P}_i(y_e) \;=\; N_i \, . \tag{7}$$

In each x-bin the statistical uncertainty on the expected number of events per species i is

$$\sigma(\langle N_i^{\delta x} \rangle) = \sqrt{\sum_{e \subset \delta x} \left({}_s\mathcal{P}_i(y_e)\right)^2} \, . \tag{8}$$

The covariance V_{ij} reads

$$V_{ij} = \sum_{e \subset \delta x} \left({}_s\mathcal{P}_i(y_e) \cdot {}_s\mathcal{P}_j(y_e) \right) \tag{9}$$

A proof of the following equations is given in reference [1].

4 Training NeuroBayes with sPlot Weights

The sPlot formalism permits a training of the neural network with experimental data only. No simulated signal is necessary because both signal and background distributions are reconstructed by weighting all events with the appropriate sWeights. To assure a correct treatment of such a training within NeuroBayes the code of the program itself had to be adapted . The main changes are:

- Check for sPlot training: If the target has one value different from $\{$-1,0,1$\}$ and the training is a classification then the training is treated as an sPlot training
- The target variable has to be set by the user to the sWeight for signal ${}_s\mathcal{P}_{sig}$ defined by Equation 5. In the case of a simple sideband subtraction with as many bins as species the weights defined by Equation 3 can be used. The result is the same but the derivation is much simpler.
- Events are passed only once to the network. However, internally Neurobayes uses ${}_s\mathcal{P}_{sig}$ to reconstruct the signal distributions and ${}_s\mathcal{P}_{back} = 1 - {}_s\mathcal{P}_{sig}$ to estimate the background distributions for all input variables.
- The quadratic and the entropy error function of the neural network as well as their derivatives have been adapted. For each event there is a signal and background contribution. As an example the quadratic error function now reads :

$$E \sim \sum_{i=1}^{n_{events}} (t_i - o_i)^2 \longrightarrow E \sim \sum_{i=1}^{n_{events}} \left({}_s\mathcal{P}_{sig} \cdot (1 - o_i)^2 + {}_s\mathcal{P}_{back} \cdot (0 - o_i)^2 \right) .$$

  Here t_i is the target value for event i and o_i stands for the network output of event i.

- The calculation of the statistical uncertainties for the signal purity p_s has been adapted for sPlot training. These errors are used during preprocessing by the orthogonal polynomial fit (preprocessing flag 14) and the monotonous spline fit (preprocessing flag 15).

References

[1] "*SPlot: A Statistical tool to unfold data distributions*", http://arxiv.org/abs/physics/0402083