Machine Learning on sWeighted data

March 13, 2018

Maxim Borisyak[1,3], Nikita Kazeev[1,2,3]

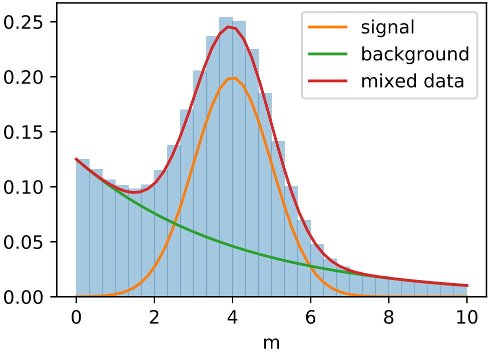
[1] National Research University Higher School of Economics [2] Sapienza University of Rome [3] Yandex School of Data Analysis

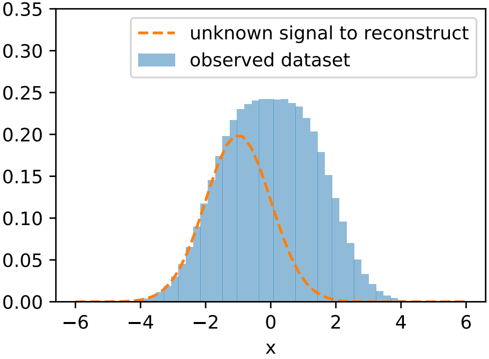
Problem domain

- A dataset consisting of examples from several sources
- No reliable information on the source from which came each particular example
- Known distributions of feature m for all sources
- We want to get the distribution of feature x for the signal source, x distribution is independent from m

Toy example

Two sources, signal and background:





Enter sWeights

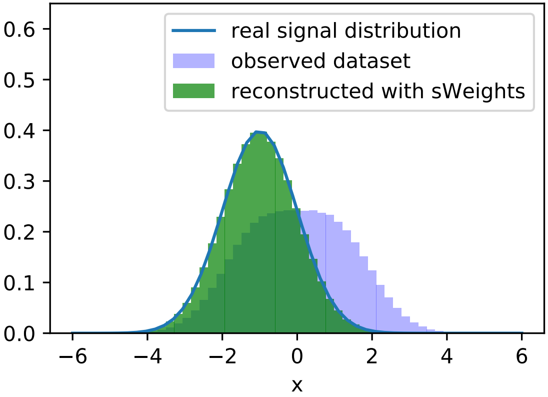
$$\mathbf{P} = \begin{array}{ccl} p\,(\text{signal}|m) & p(\text{background}|m) & \\ \left[\begin{matrix} p_{1,1} \\ p_{2,1} \\ p_{3,1} \\ \dots \end{matrix}\right. & \left.\begin{matrix} 1-p_{1,1} \\ 1-p_{2,1} \\ 1-p_{3,1} \\ \dots \end{matrix}\right] & \begin{matrix} \text{example 1} \\ \text{example 2} \\ \text{example 3} \\ \dots \end{matrix} \end{array}$$

$$\text{sWeights} = \mathbf{W} = \mathbf{P} \cdot \left(\left(\mathbf{P}^T \cdot \mathbf{P}\right)^{-1} \cdot \left[\sum p_{i,1}, \sum 1 - p_{i,1} \right] \right)$$

$$\mathbf{P} = \left(\mathbf{W} \cdot \left(\mathbf{W}^T \cdot \mathbf{W}\right)^{-T} \right) \cdot \left[\sum w_{i,1}, \sum 1 - w_{i,1} \right]$$

Paper [1], ROOT implementation, Python implementation

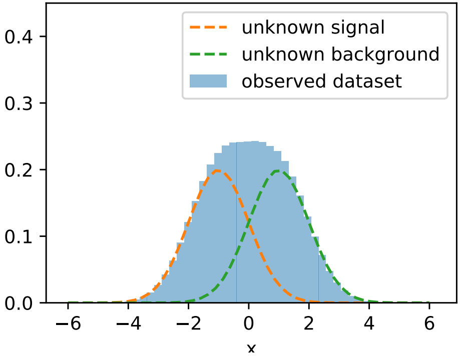
Apply sWeights



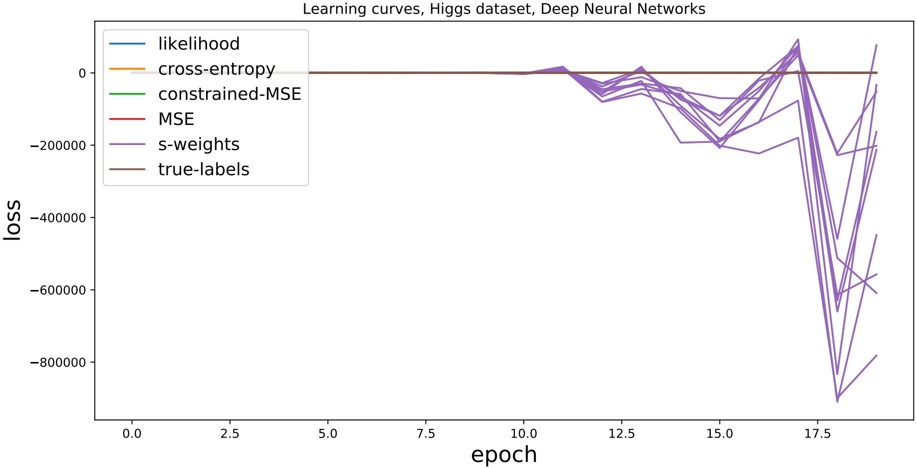
Enter Machine Learning

We want to train a machine learning algorithm to separate signal from background using the information in x

Paper [2]: Use each example twice, once as signal, once as background with corresponding sWeights as example weights for a classifier



Let’s train an NN



Why can't I just use sWeight as sample_weight?

Some sWeights are by design negative. Take logloss and a signal example with negative weight w:

$$L = -w \cdot \log(p),$$

where p is the signal probability.

$$\lim_{p \to 0} L = -(-|w|) \lim_{p \to 0} \log(p) = -\infty$$

If the algorithm is able to isolate a negative weight example, it can optimize the total loss into $-\infty$ ignoring the rest of the dataset

Collapsing sWeights to probability: intuition

- Data distribution is a mix of signal and background distributions
- It should be possible to reweight the dataset with ordinary positive weights equal to $p_{\text{signal}}(x) = \frac{\text{pdf}_{\text{signal}}(x)}{\text{pdf}_{\text{mix}}(x)}$
- Using sWeights results in the same distribution

Collapsing sWeights to probability

> To get the probability that an example with given features x is signal, we need to find the average sWeight for examples with features x

Proof is in the backup

Collapsing sWeights to probability

> To get the probability that an example with given features x is signal, we need to find the average sWeight for examples with features x

One problem: x usually is a high-dimensional real vector, we have a single example for each x value

Proof is in the backup

Collapsing sWeights to probability: practical

> Train a regression bound to $[0, 1]$ to predict sWeight given x as features. Use the result as the weights further in the training pipeline.

There is no one-to-one mapping of x to w – by the design of the sWeights. However, for a regression using mean squared error the minimum is achieved when prediction is equal to $\mathbb{E}\left(\text{sWeight}|x\right)$

Signal vs. background: likelihood

We also propose the following loss:

$$-\log\left[p\left(\text{signal}|m\right)\cdot f(x)+p\left(\text{background}|m\right)\cdot(1-f(x))\right]$$

- $p\left(\text{signal, background}|m\right)$ are the probabilities obtained from the m distributions that are normally used to compute sWeights
- $f(x)\in[0,1]$ is the signal probability predicted by the classifier

Proof is in the backup

Experiments

Two problems:

- Classifications of the same signal vs. background as were used in building sWeights
- Classification of one sWeighted dataset vs. another sWeighted dataset

Two open datasets:

- ATLAS Higgs, not using weights, sWeights added artificially, 28 tabular features, $8.8 \cdot 10^6$ train, $2.2 \cdot 10^6$ test
- LHCb Muon ID, includes sWeights, 123 features, $7 \cdot 10^6$ train, $1.7 \cdot 10^6$ test, pion vs muon, not using momentum and momentum reweighting

Two models:

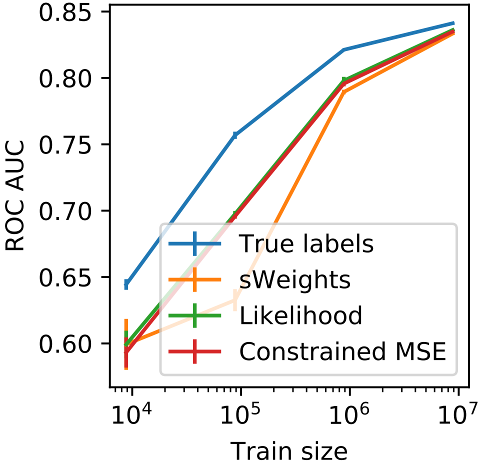
- Catboost
- Deep fully-connected neural network (NN)

Higgs – NN

Fully-connected neural network (NN), 3 layers, 128, 64, 32 neurons in layer, leaky relu (0.05), adam(learning_rate=1e-3, beta1=0.9, beta2=0.999)

- True labels – logloss using the true labels
- sWeights – using sWeights as weights for logloss
- Likelihood – our likelihood
- Constrained MSE – our regression

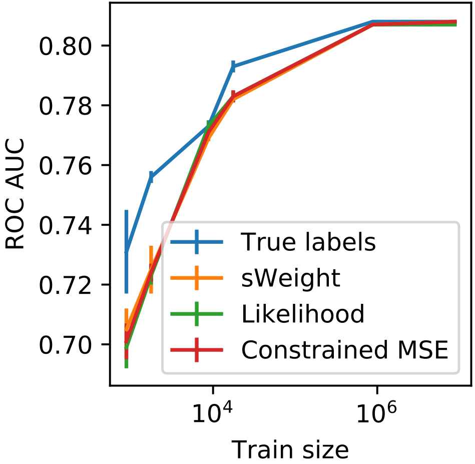
Training epochs is the right moment so that the training doesn't explode completely



Higgs – Catboost

Catboost with 1000 trees

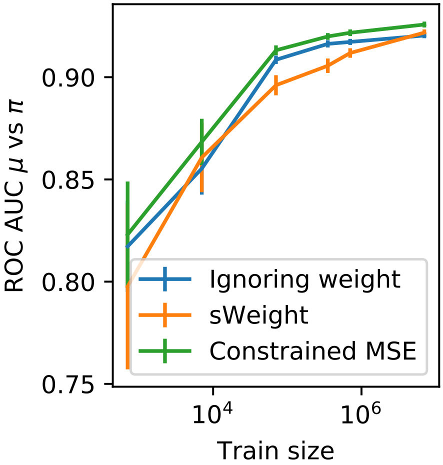
- True labels – logloss loss using the true labels
- sWeights – using sWeights as weights for logloss
- Likelihood – our likelihood
- Constrained MSE – our regression



MuID – Catboost

Catboost with 1000 trees, separate sWeights to probabilty regressions per particle type

- Ignoring weight – logloss without weights
- sWeights – using sWeights as weights for logloss
- Constrained MSE – our regression



Conclusion

- Training an MLP classifier on sWeighted data results in chaotic behaviour
- We propose two mathematically rigorous loss functions for traininig a classifier on sWeighted data
- We show our methods outperform directly using sWeights as example weights; effect size decreases with sample size increase

Code for Catboost that implements regression constrained to $[0, 1]$ and the likelihood

Acknowledgments

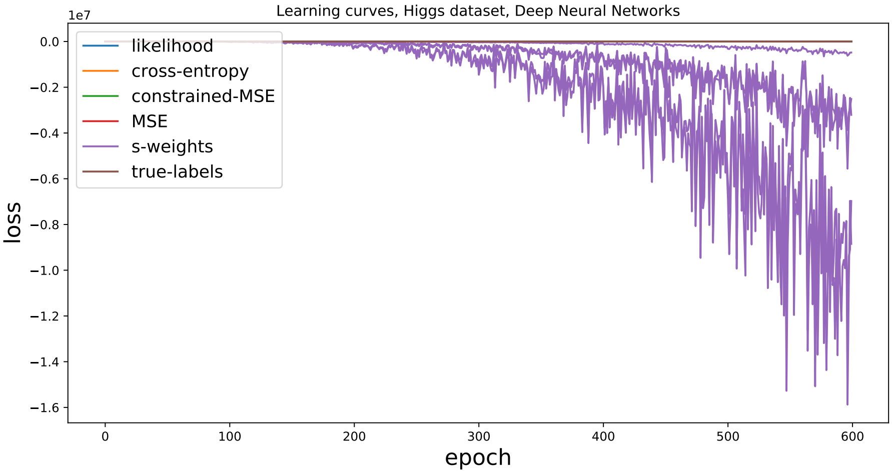
- Artem Maevskiy for suggestions on improving learning stability and the weighted ROC AUC code
- Denis Derkach for suggestions on experiment and presentation design
- Fedor Ratnikov for suggestions on presentation design
- Andrey Ustyuzhanin for suggestions on presentation design
- LHCb and ATLAS for opening their data
- The research leading to these results has received funding from Russian Science Foundation under grant agreement n° 17-72-20127

References

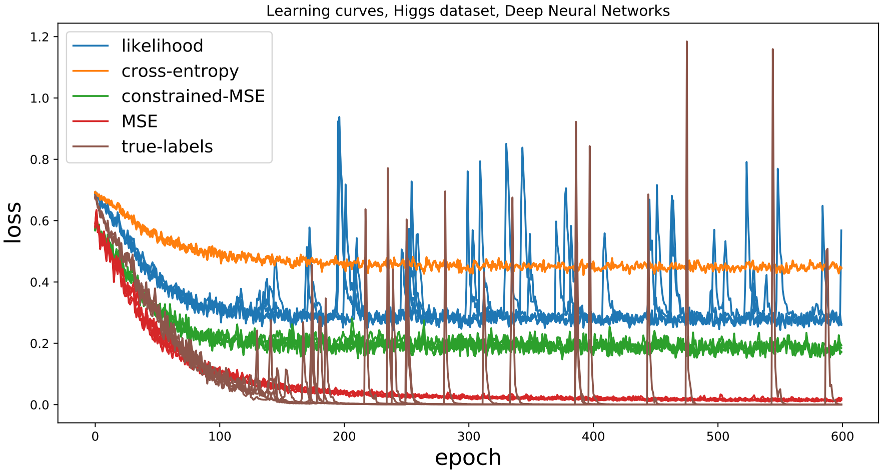
- Pivk, Muriel, and Francois R. Le Diberder. ”Plots: A statistical tool to unfold data distributions.” Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 555.1-2 (2005): 356-369.
- Keck, Thomas. Machine learning algorithms for the Belle II experiment and their validation on Belle data. No. ETP-KA-2017-31. 2017.
- Natarajan, Nagarajan, et al. ”Learning with noisy labels.” Advances in neural information processing systems. 2013.

Backup

Learning curves – Higgs results, sWeights



Learning curves – Higgs results, other



Collapsing sWeights to probability – proof

Let $f(x)$ be any function of the features x, such as output of a machine learning algorithm, $w(m)$ the sWeight

$$E_{x\ p_{\text{sig}}}\left[f(x)\right] = \int dx f(x) p_{\text{sig}}(x)$$

$$W(x) = \frac{p_{\text{sig}}(x)}{p_{\text{mix}}(x)}$$

$$E_{x\ p_{\text{sig}}}\left[f(x)\right] = \int dx f(x) W(x) p_{\text{mix}}(x) \tag{1}$$

Let m be the variable used to compute sWeights:

$$E_{x\ p_{\text{sig}}}\left[f(x)\right] = \int dx dm w(m) f(x) p_{\text{mix}}(x, m)$$

Collapsing sWeights to probability

sPlot requires that x and m are independent:

$$E_{x\ p_{\text{sig}}}\left[f(x)\right] = \int dx dm w(m) f(x) p_{\text{mix}}(x) p_{\text{mix}}(m|x)$$

$$E_{x\ p_{\text{sig}}}\left[f(x)\right] = \int dx f(x) p_{\text{mix}}(x) \int dm w(m) p_{\text{mix}}(m|x)$$

From (1)

$$\int dx f(x)\, W(x) p_{\text{mix}}(x) = \int dx f(x) p_{\text{mix}}(x) \int dm w(m) p_{\text{mix}}(m|x)$$

$$W(x) = \int dm w(m) p_{\text{mix}}(m|x)$$

Likelihood – proof

s – the example is signal, b – is background, $f(x)$ – predicted signal probability

$$\begin{aligned} p(m, x|\text{model}) &= p(m, x|\text{model}, s)p(s) + p(m, x|\text{model}, s)p(b) \\ &\sim p(m|s)p(x|s, \text{model}) + p(m|b)p(x|b, \text{model}) \\ &= p(m|s)\frac{p(s|x, \text{model})p(s)}{p(x)} + \text{same for b} \end{aligned}$$

$$\begin{aligned} L &= \log p\left(m, x|\text{model}\right) \\ &= \log\left[p(m|s)p(s|x, \text{model}) + p(m|b)p(b|x, \text{model})\right] - \log p(x) \\ &= \log\left[p(m|s)f(x) + p(m|b)(1 - f(x))\right] + \text{const} \end{aligned}$$

Loss might be convex

Paper [3] has proof that sWighted (they don't use the term though) loss with just two m values is convex if the original loss is symmetric