REWEIGHTING DISTRIBUTIONS WITH BOOSTED DECISION TREES

Alex Rogozhnikov
Yandex, NRU HSE

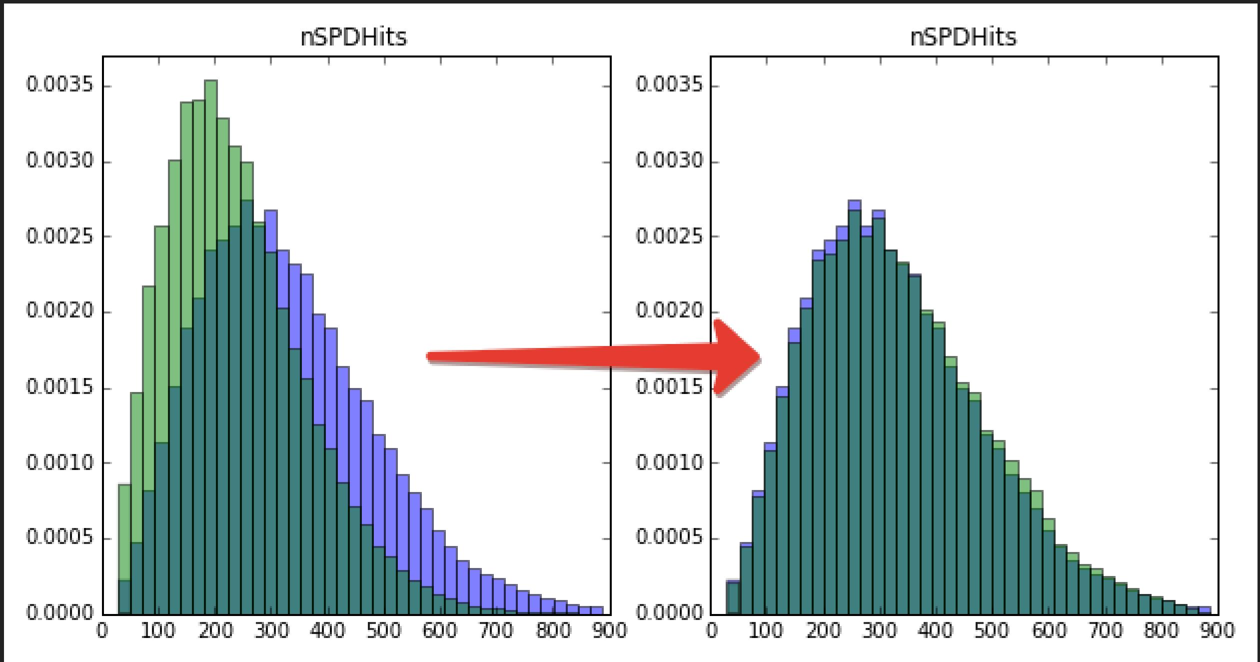
ACAT, Valparaiso, 2016

WHEN REWEIGHTING IS APPLIED?

Reweighting in HEP is used to minimize difference between real data (RD) and Monte-Carlo (MC) simulation.

Known process is used, for which real data can be obtained.

Target of reweighting: assign weights to MC s.t. MC and RD distributions coincide:



APPLICATIONS BEYOND PHYSICS

Introducing corrections to fight non-response bias: assigning higher weight to answers from groups with low response.

See e.g. R. Kizilcec, "Reducing non-response bias with survey reweighting: Applications for online learning researchers", 2014.

I'll talk in physical terms (RD and MC), but everything is applicable to any reweighting.

TYPICAL APPROACH

Usually histogram reweighting is used, in each bin the weight of original distribution is multiplied by:

$$\text{multiplier}_{\text{bin}} = \frac{w_{\text{target, bin}}}{w_{\text{original, bin}}}$$

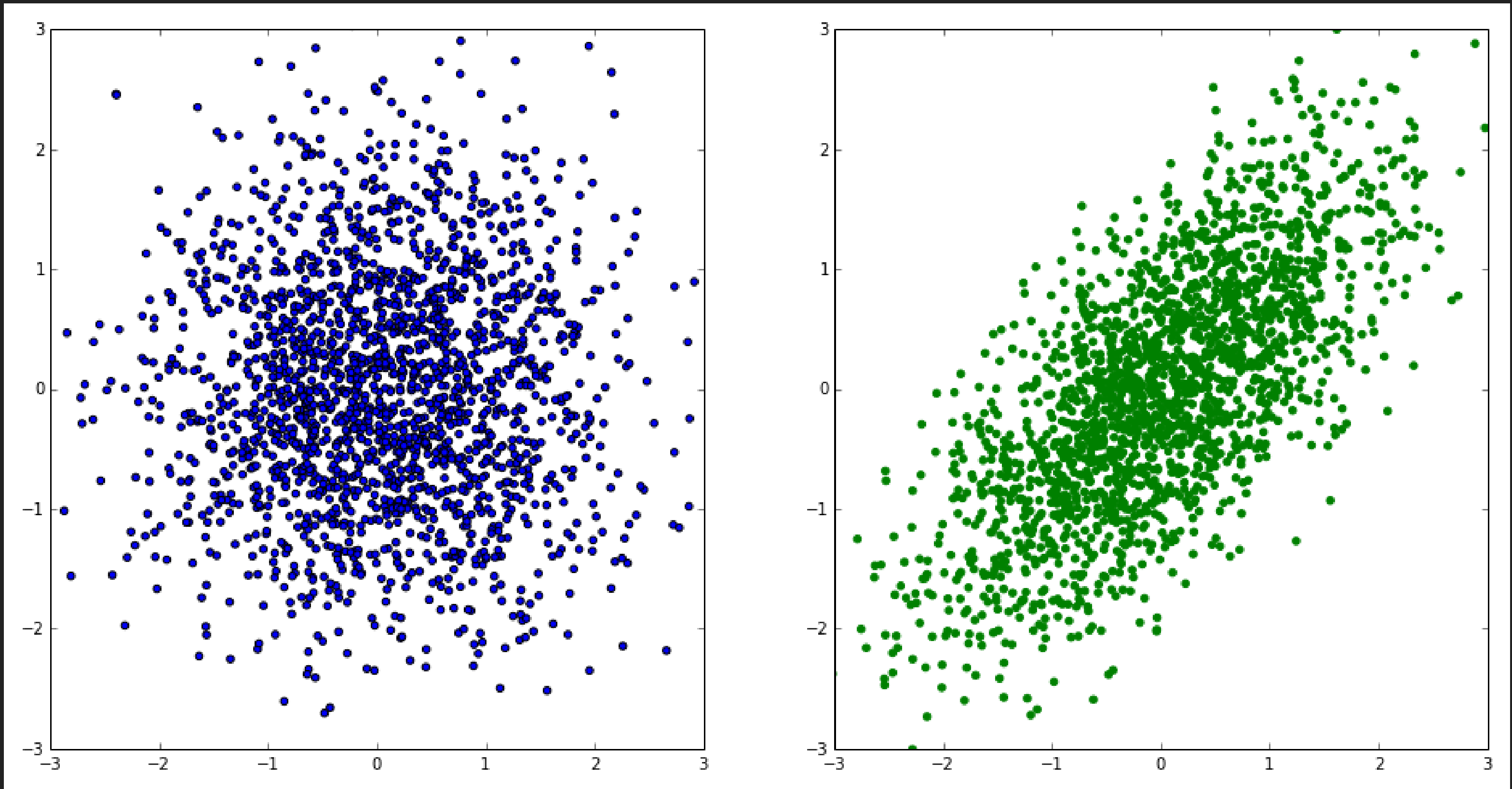
$w_{\text{target, bin}}, w_{\text{original, bin}}$ — total weight of events in bin for target and original distributions.

1. Simple and fast!
2. Very few (typically, one or two) variables
3. Reweighting one variable may bring disagreement in others
4. Which variable to use in reweighting?

A better approach is proposed in this presentation.

COMPARING DISTRIBUTIONS, MEASURING DIFFERENCE

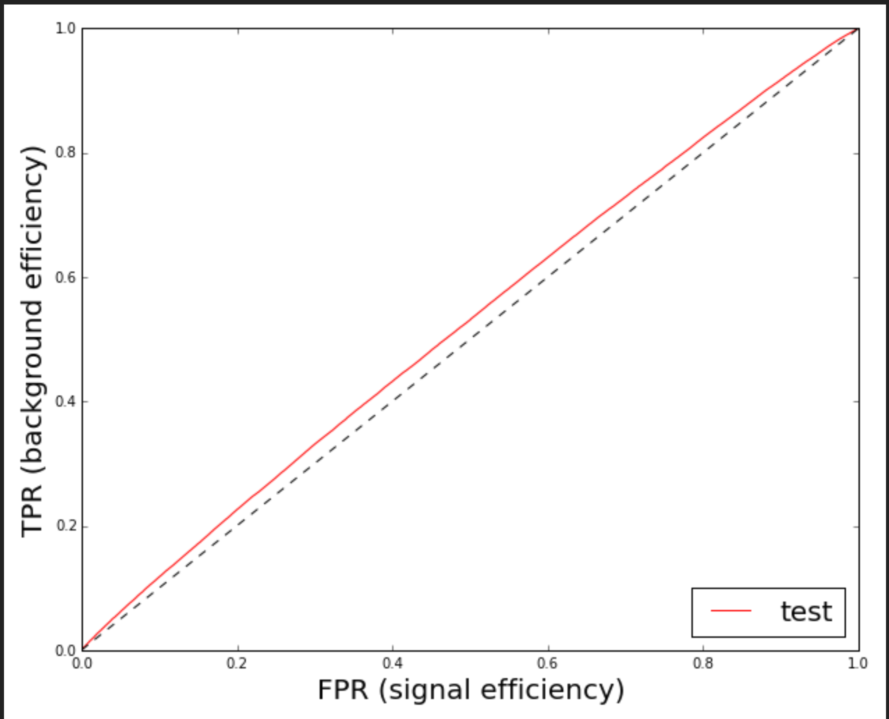
- one dimension: KS-test, CvM, Mann-Whitney (U-test)
- two or more dimensions?
- comparing 1d projections is not the way:



COMPARING DISTRIBUTIONS USING ML

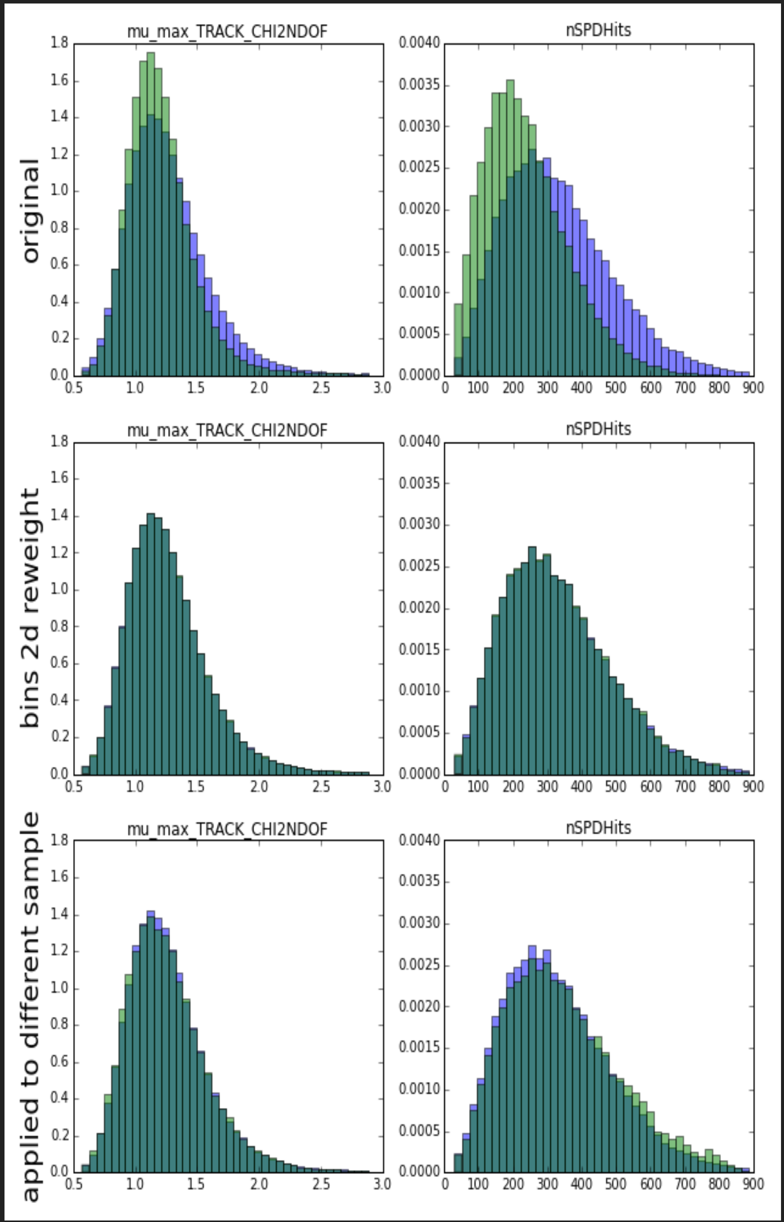
Final goal: define if the classifier discriminates RD and MC.

Comparison shall be done using ML, output of classifier is 1-dimensional. Looking at ROC curve on a holdout:



See also: J. Friedman, On Multivariate Goodness–of–Fit and Two–Sample Testing, 2003

REWEIGHTING WITH HISTOGRAMS: EXAMPLE



Problems arise when there are too few events in bin.

But this may be checked on holdout.

Tradeoff:

1. few bins — rule is rough
2. many bins — rule is not reliable

REUSING ML CLASSIFIERS TO REWEIGHTING

We need to estimate density ratio $\dfrac{f_1(x)}{f_0(x)}$

Classifier trained to discriminate MC and RD should reconstruct probabilities $p_0(x)$ and $p_1(x)$.

So, for reweghting we can use $\dfrac{p_1(x)}{p_0(x)} \sim \dfrac{f_1(x)}{f_0(x)}$

- able to reweight in many variables
- successfully tried in HEP, see D. Martschei et al, "Advanced event reweighting using multivariate analysis", 2012
- poor reconstruction when ratio is too small / high
- slow

BETTER IDEA:

Write ML algorithm to solve directly reweighting problem

1. Split space of variables in several large regions
2. Find this regions 'intellectually'!

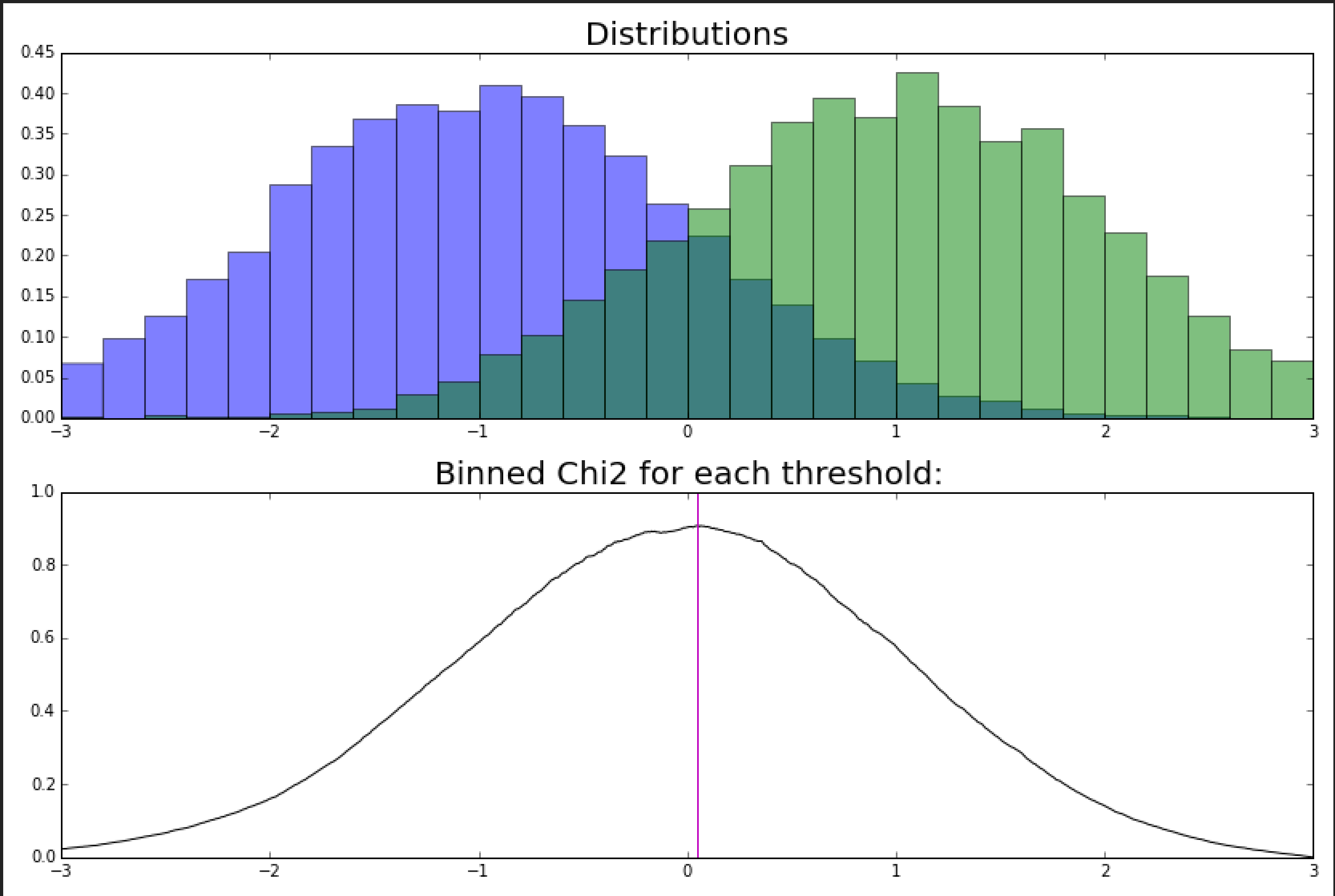
USING DECISION TREE TO FIND REGIONS

1. Tree splits the space of variables by orthogonal splits
2. Finding regions with high difference by maximizing symmetrized χ^2:

$$\chi^2 = \sum_{\text{bin}} \frac{(w_{\text{bin, original}} - w_{\text{bin, target}})^2}{w_{\text{bin, original}} + w_{\text{bin, target}}}$$

SYMMETRIZED BINNED χ^2

Finding optimal threshold to split variable into two bins:



BDT REWEIGHTER

To train reweighter many times repeat following steps:

1. build a shallow tree to maximize symmetrized χ^2
2. compute predictions in leaves:

$$\text{leaf\_pred} = \log \frac{w_{\text{leaf, target}}}{w_{\text{leaf, original}}}$$

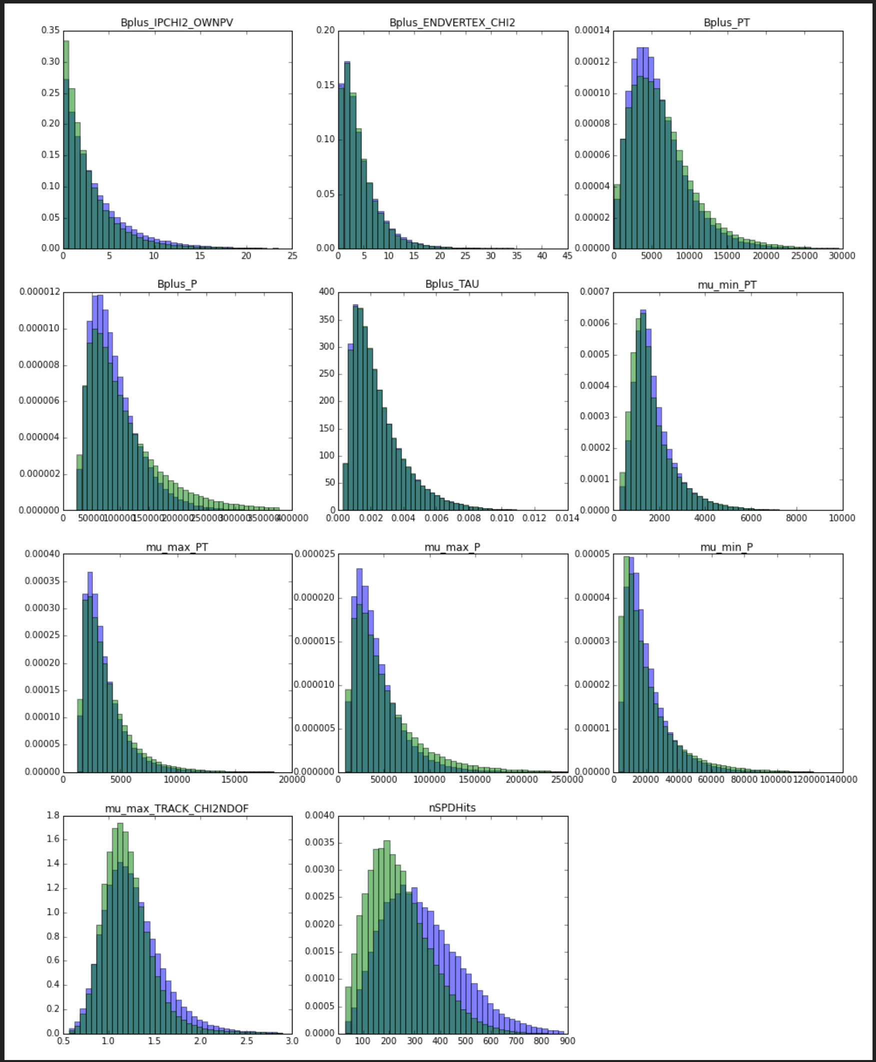
3. reweight distributions (compare with AdaBoost):

$$w = \begin{cases} w, & \text{if event from target (RD) distribution} \\ w \times e^{\text{pred}}, & \text{if event from original (MC) distribution} \end{cases}$$

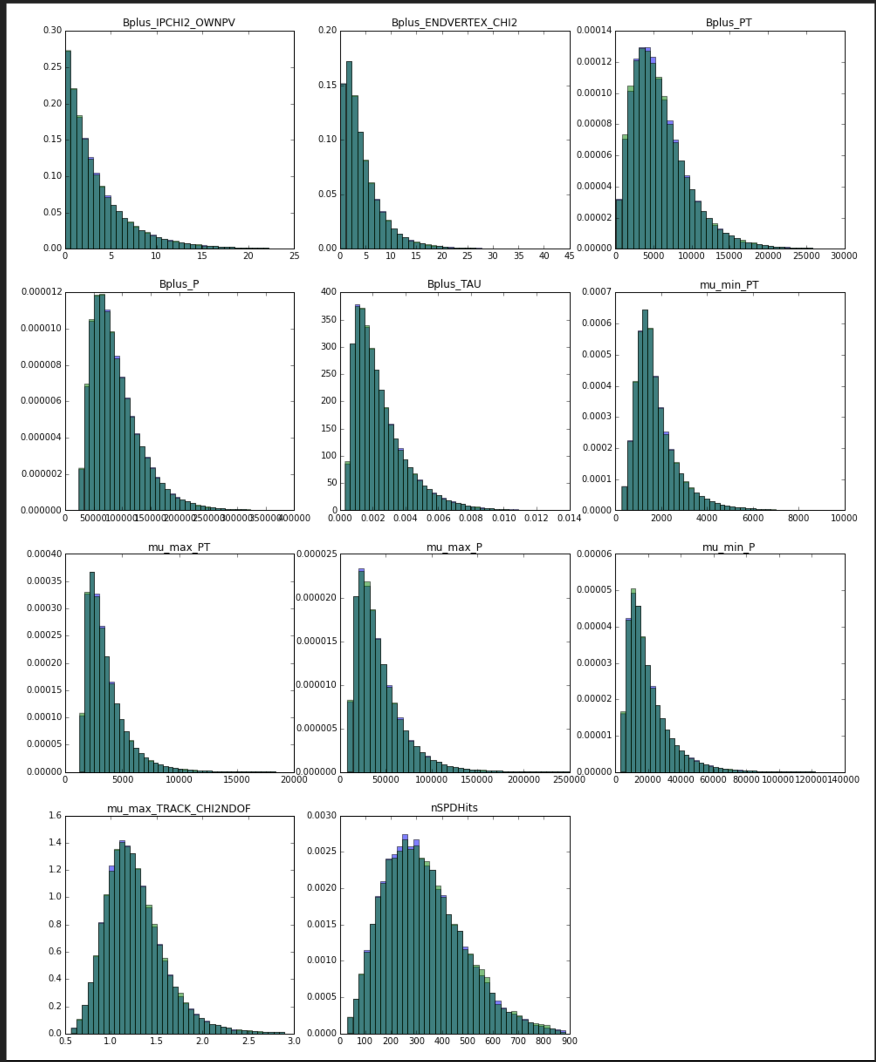
CHANGES COMPARED TO GBDT

- tree splitting criterion (MSE $\to \chi^2$)
- different boosting procedure

DEMONSTRATION



original



BDT reweighted

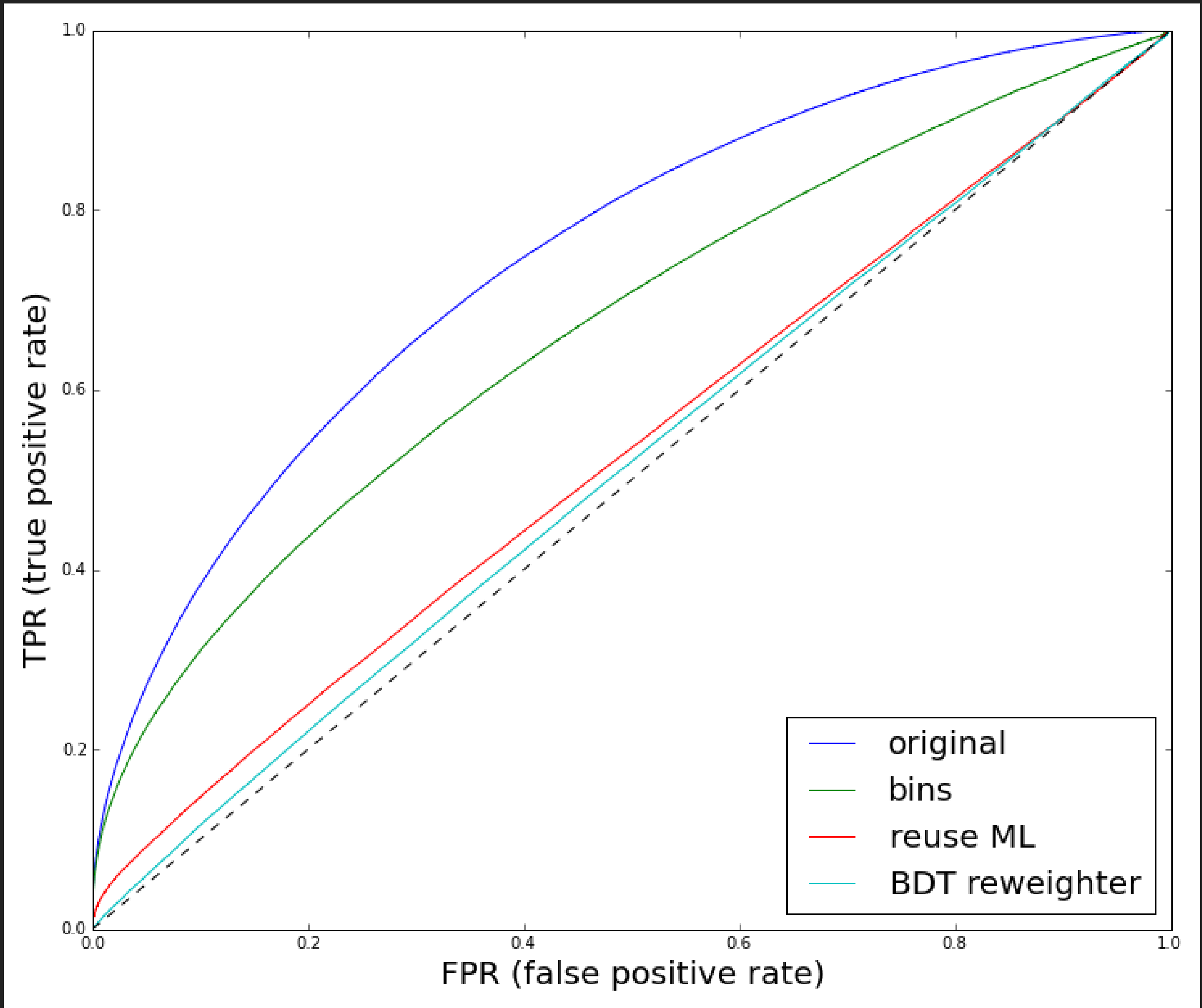
KS DISTANCES AFTER REWEIGHTING

Bins reweighter uses only 2 last variables (60×60 bins); ML and BDT use all variables

Feature	original	bins	reuse ML	BDT reweighter
Bplus_IPCHI2_OWNPV	0.0796	0.0642	0.0463	0.0028
Bplus_ENDVERTEX_CHI2	0.0094	0.0175	0.0490	0.0021
Bplus_PT	0.0586	0.0679	0.0126	0.0053
Bplus_P	0.1093	0.1126	0.0044	0.0047
Bplus_TAU	0.0037	0.0060	0.0324	0.0044
mu_min_PT	0.0623	0.0604	0.0017	0.0036
mu_max_PT	0.0483	0.0561	0.0053	0.0035
mu_max_P	0.0906	0.0941	0.0084	0.0036
mu_min_P	0.0845	0.0858	0.0058	0.0043
mu_max_TRACK_CHI2NDOF	0.0956	0.0042	0.0128	0.0043
nSPDHits	0.2478	0.0098	0.0180	0.0075

COMPARING RESULTS WITH ML

Using approach to distribution comparison described earlier.
Reweighting two variables wasn't enough:



BOOSTED REWEIGHTING

- uses each time few large bins
- is able to handle many variables
- requires less data (for same performance)
- ... but slow

Implemented in hep_ml library:

```
from hep_ml.reweight import GBReweighter
reweighter = GBReweighter()
reweighter.fit(mc_data, real_data, target_weight=real_data_sweights)
reweighter.predict_weights(other_mc_data)
```

FEATURE IMPORTANCES

Being a variation of GBDT, BDT reweighter is able to calculate feature importances. Two features used in reweighting with bins are indeed most important

	importance
feature	
mu_max_TRACK_CHI2NDOF	0.240272
nSPDHits	0.209090
Bplus_P	0.122314
mu_min_P	0.115245
Bplus_PT	0.080641
Bplus_IPCHI2_OWNPV	0.068209
mu_max_P	0.060518
mu_max_PT	0.037863
mu_min_PT	0.037761
Bplus_ENDVERTEX_CHI2	0.026598
Bplus_TAU	0.001489

CONCLUSION

1. Comparison of multidimensional distributions is ML problem
2. Reweighting of distributions is ML problem

BDT reweighter was proposed

1. works with many variables
2. provides stable and accurate results
3. doesn't require much tuning

Poster about flavour tagging of B-mesons on Friday!

Inclusive Flavour Tagging Algorithm

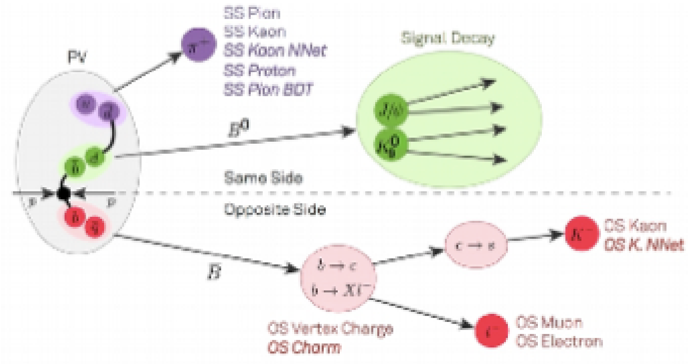
Tatiana Likhomanenko, Alexey Rogozhnikov, Denis Derkach

NRC "Kurchatov Institute", Yandex School of Data Analysis, Higher School of Economics
E-mail: antares@yandex-team.ru

17th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Chile, 18-22 January, 2016

Introduction and problem formulation

The Flavour Tagging (FT) algorithm determines the flavour of each reconstructed signal B meson at the production point. The B meson consists either of a b or a $\bar{b}$ quarks, which defines its flavour.



- The production of a B meson is often accompanied by the production of another b hadron and other particles, like kaon, pion, and proton. The FT algorithms are usually divided into two groups (Eur. Phys. J. C72 2022):
  - same side (SS) taggers exploit light particles that evolve from the production process of the signal B meson, like kaons, pions, and protons
  - opposite side (OS) taggers use decay products of b hadrons that are produced together with the signal B
- SS and OS outputs are combined to a single answer.

For most *CP* measurements the signal B decay products do not carry information on the production flavour.

The FT algorithm should predict probabilities $P(b)$ and $P(\bar{b})$.

Symme

Restric

- Cla
  pro
- Cor
- Cal
- Dis