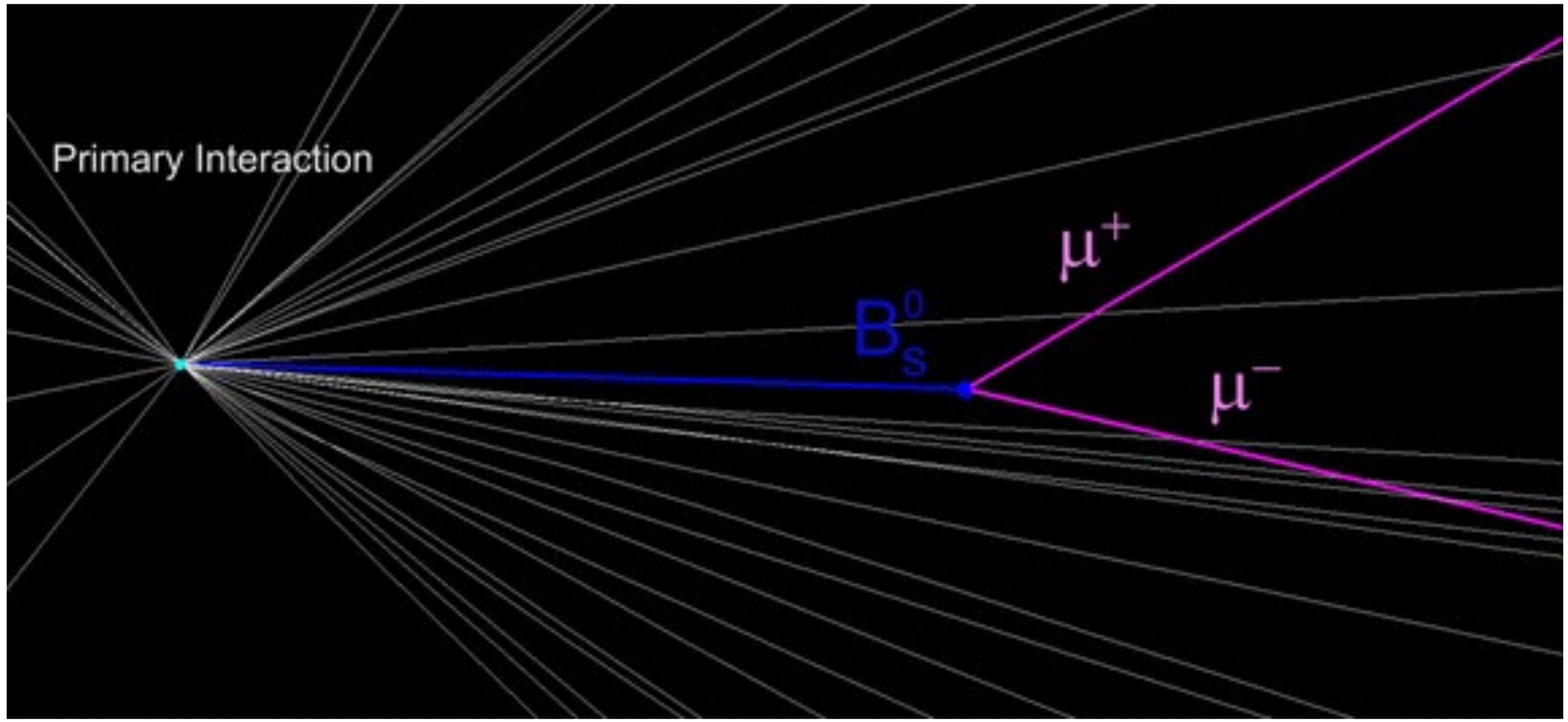
Machine learning in LHCb: triggering and flavour tagging systems

Tatiana Likhomanenko

Machine learning in LHCb: triggering and flavour tagging systems

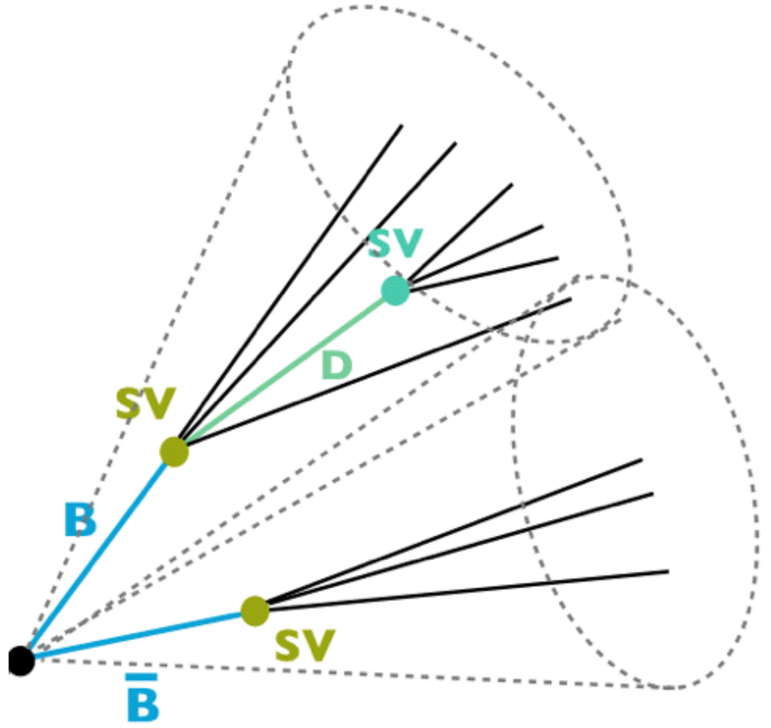
LHC data structure

LHC event

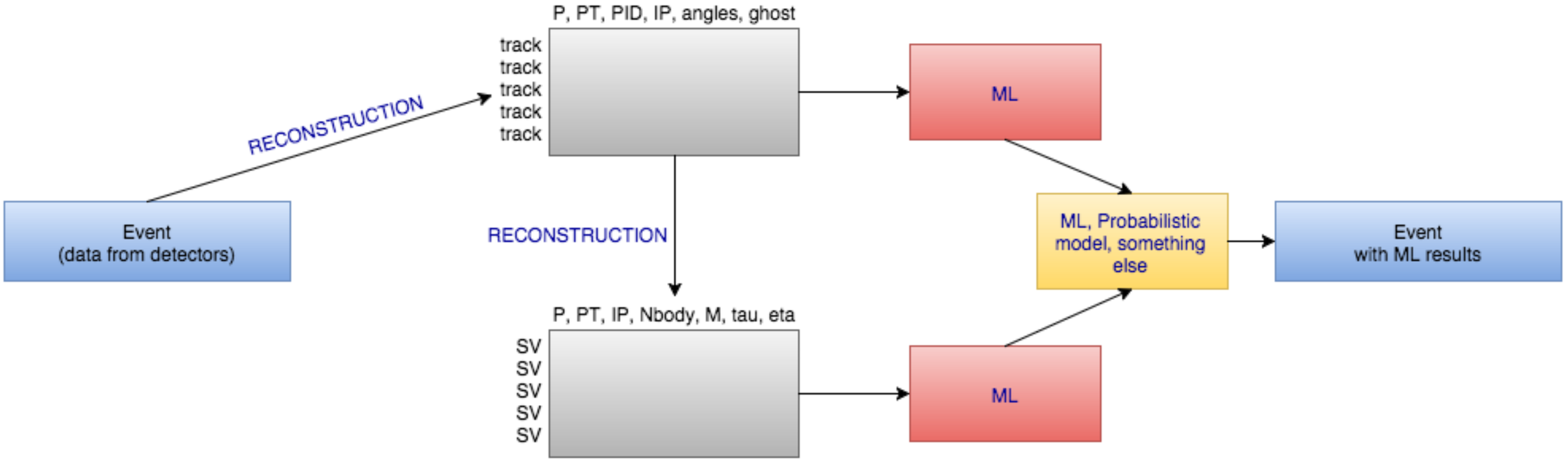


Event definition

- Proton-proton bunches collide
- Sample: one proton-proton bunches collision, called Event
- Event consists of the tracks and secondary vertices (SV), where particles are produced
- Features: a track, SV and its products physical characteristics reconstructed from the detectors (momentum, mass, angles, impact parameter)



Event processing

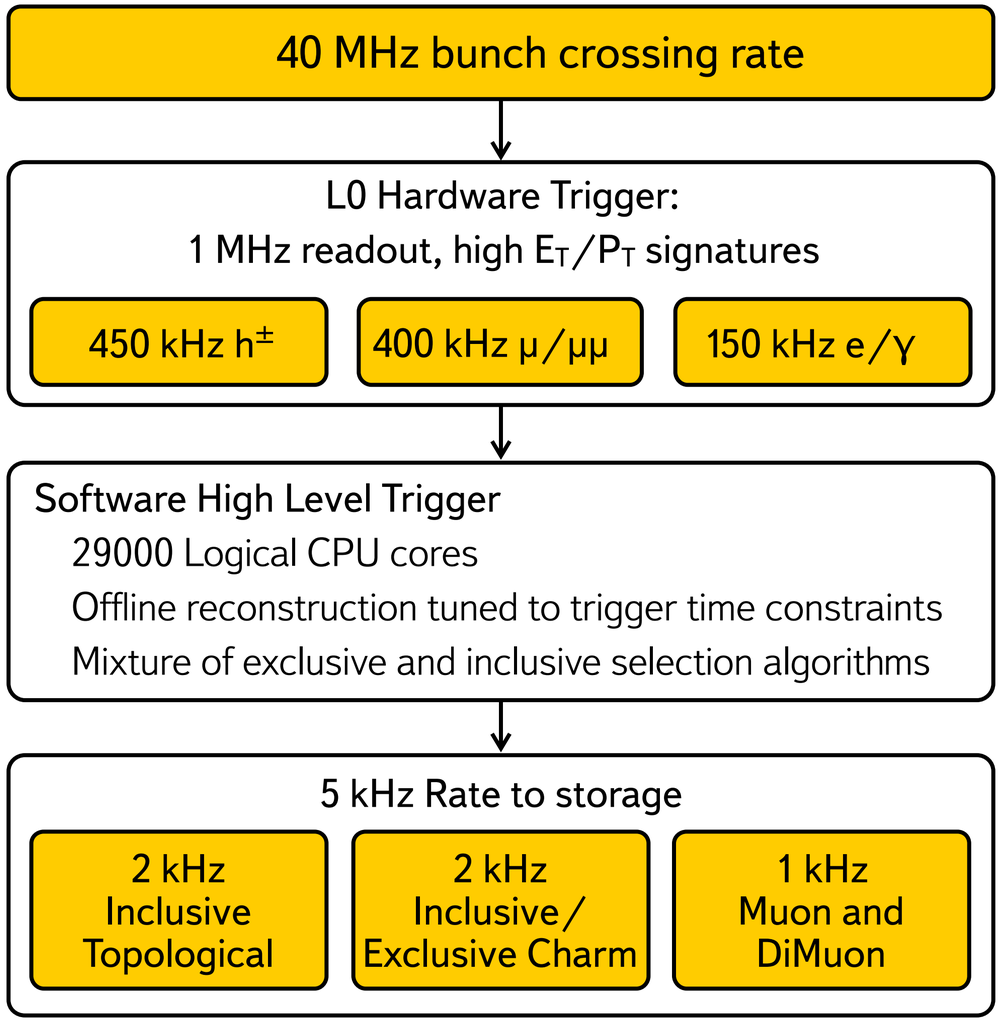


Machine learning in LHCb: triggering and flavour tagging systems

Trigger system

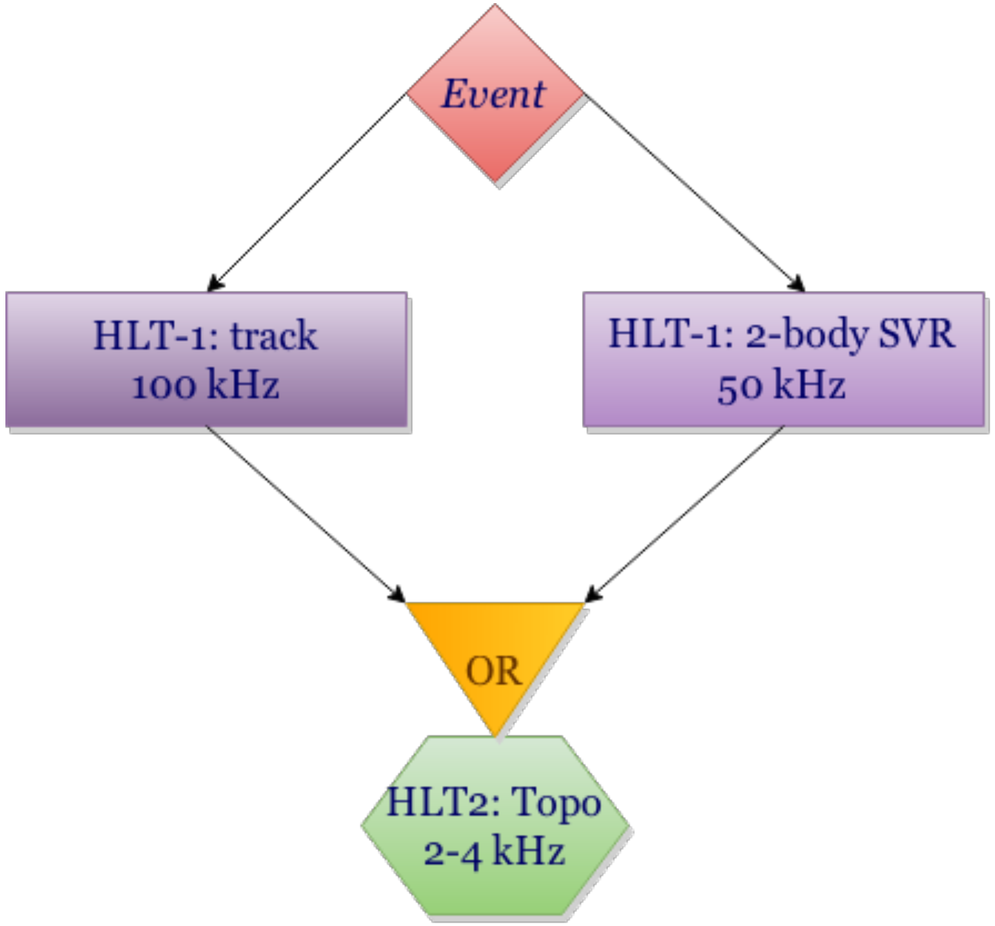
What is it?

- Select events to store them for offline processing
- Should efficiently select interesting events
- Interesting event is an event that contains at least one SV where necessary particle and products are produced
- Output rate for trigger system is limited



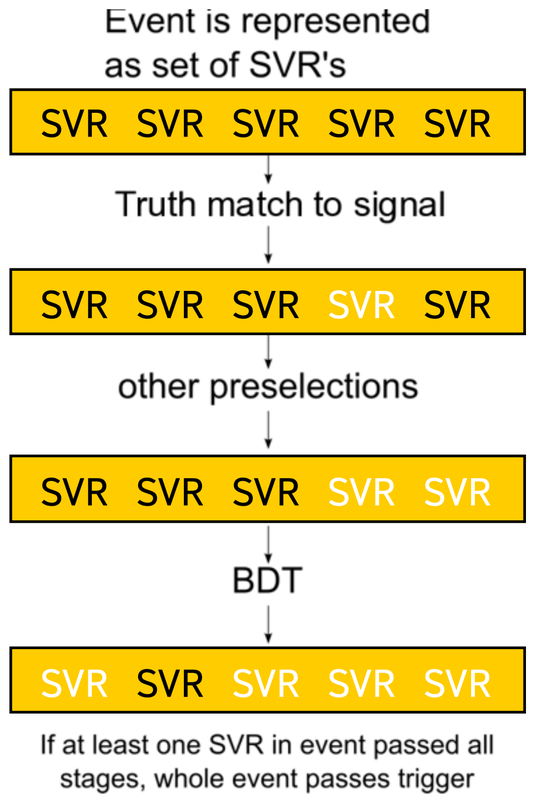
Run-II topological trigger

- HLT-1 track is looking for either one super high PT or high displacement track
- HLT-1 2-body SV classifier is looking for two tracks making a vertex
- HLT-2 improved topo classifier uses full reconstructed event to look for 2, 3, 4 and more tracks making a vertex



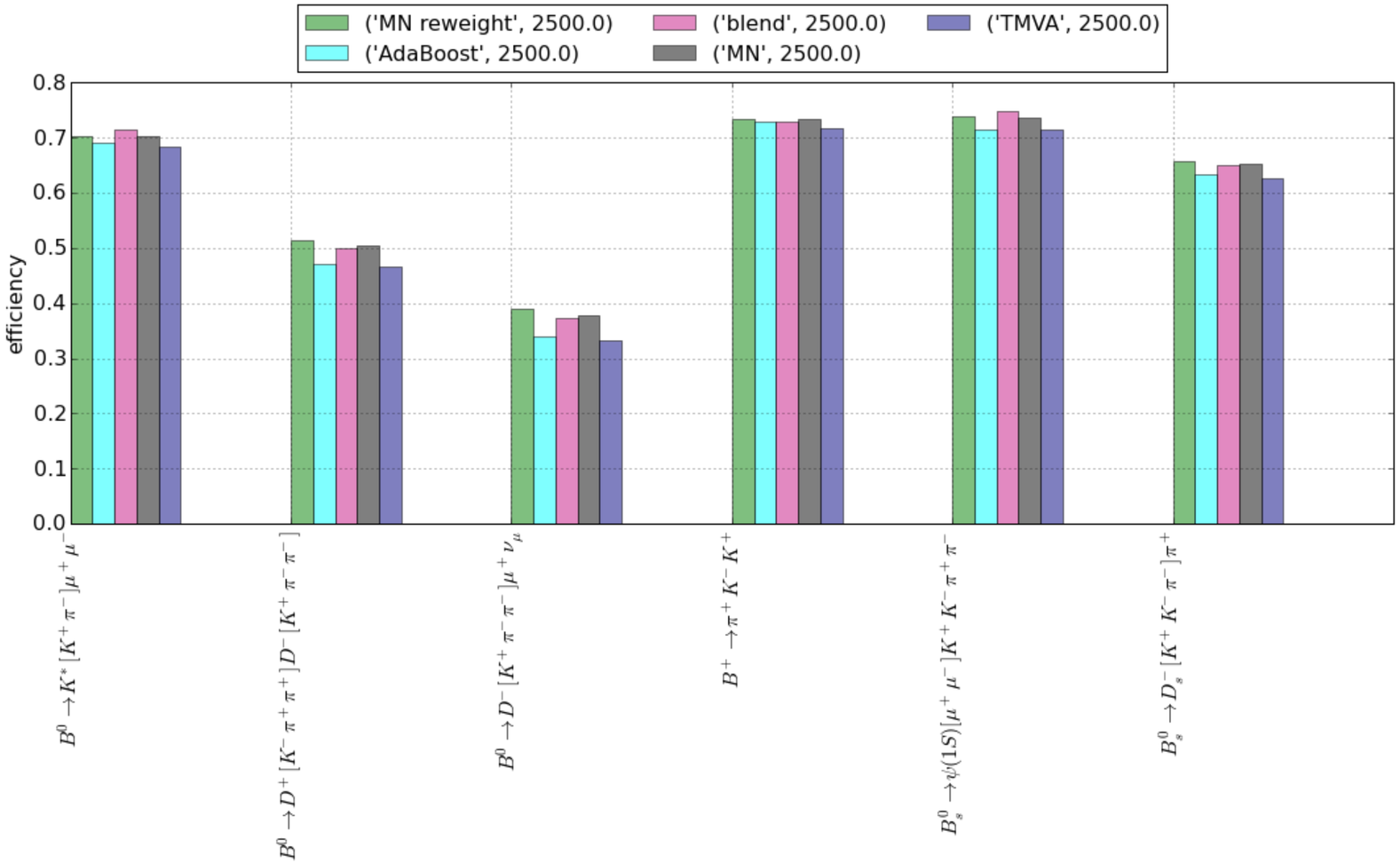
Machine learning problem

- Training data are set of SVs for all events
- Monte Carlo sample (used as signal-like) were simulated for various types of interesting events (different decays)
- Minimum bias data (real data for a small period of time) are used as background-like
- Output rate is fixed, thus, false positive rate (FPR) for events is fixed
- Goal is to improve efficiency for each type of signal events.



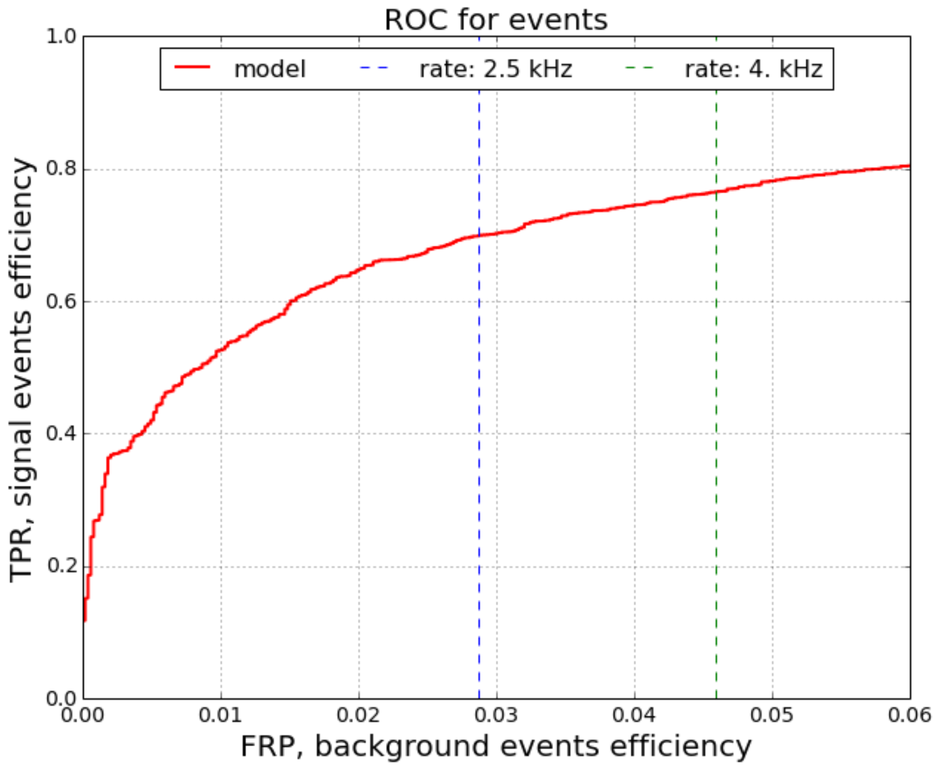
Event representation

How to measure quality?



ROC curve, computed for events

- Optimize true positive rate (TPR) for fixed FPR for events
- Weight signal events in such way that channels have the same amount of events.
- Optimize ROC curve in a small FPR region



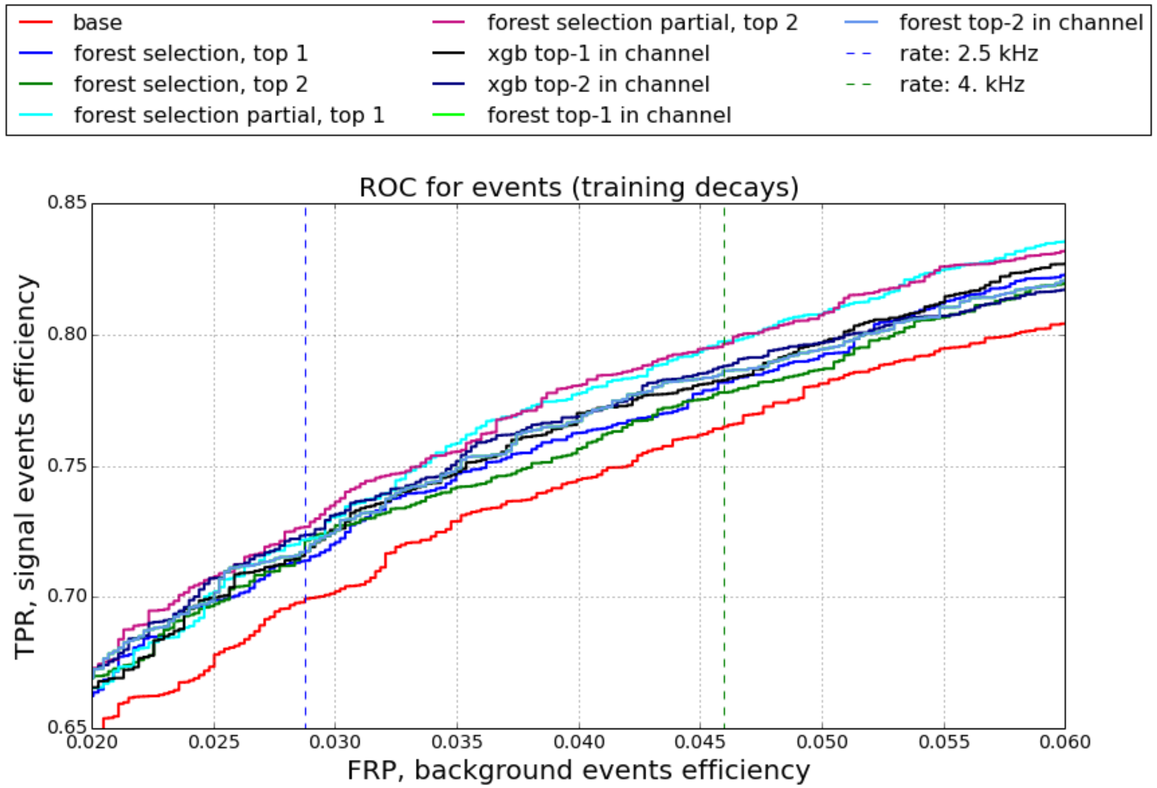
ROC curve interpretation

Trigger system: random forest trick

Simulated signal event contains at least one interesting SV, but not each SV should be interesting

Random forest for SVRs selection

- Train random forest (RF) on SVRs
- Select top-1, top-2 SVs by
  RF predictions for each signal event
- Train classifier on selected SVs



Machine learning in LHCb: triggering and flavour tagging systems

Trigger system: real-time

Online processing

There are two possibilities to speed up prediction operation:

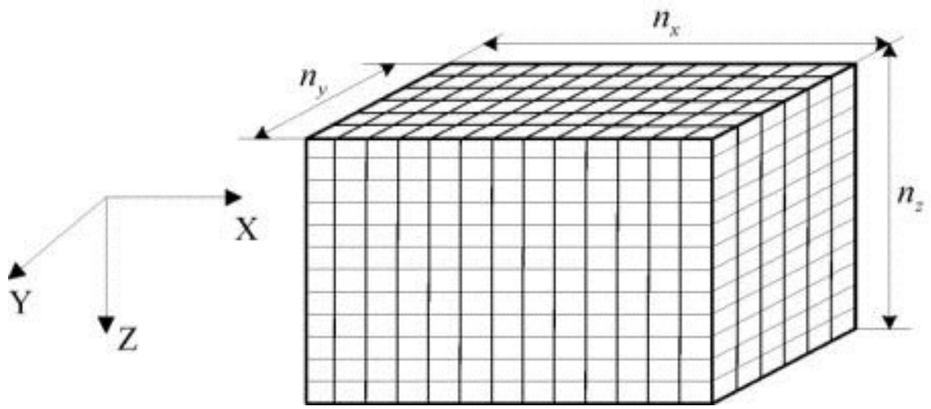
- Bonsai boosted decision tree format (BBDT)
- Post-pruning

What is MatrixNet

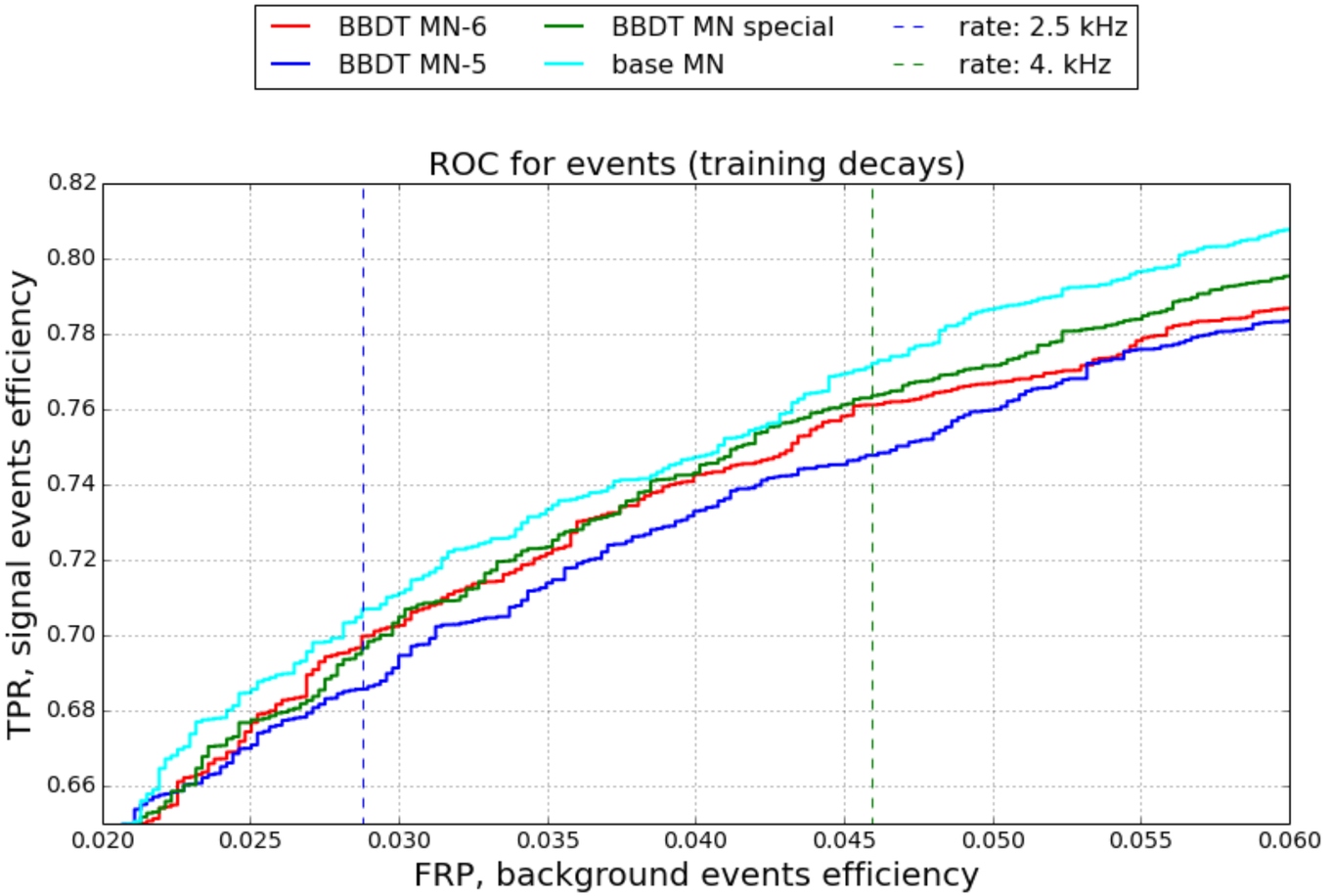
- Yandex machine leaning algorithm
- Gradient Boosting over oblivious Decision Trees
- Feature binarization (like feature hashing)
- Classification, Regression, Ranking

BBDT

- Features hashing using bins before training
- Converting decision trees to n-dimensional table (lookup table)
- Table size is limited in RAM (1Gb), thus count of bins for each features should be small (5 bins for each of 12 features)
- Discretization reduces the quality
- Prediction operation takes one reading from the table



BBDT, results



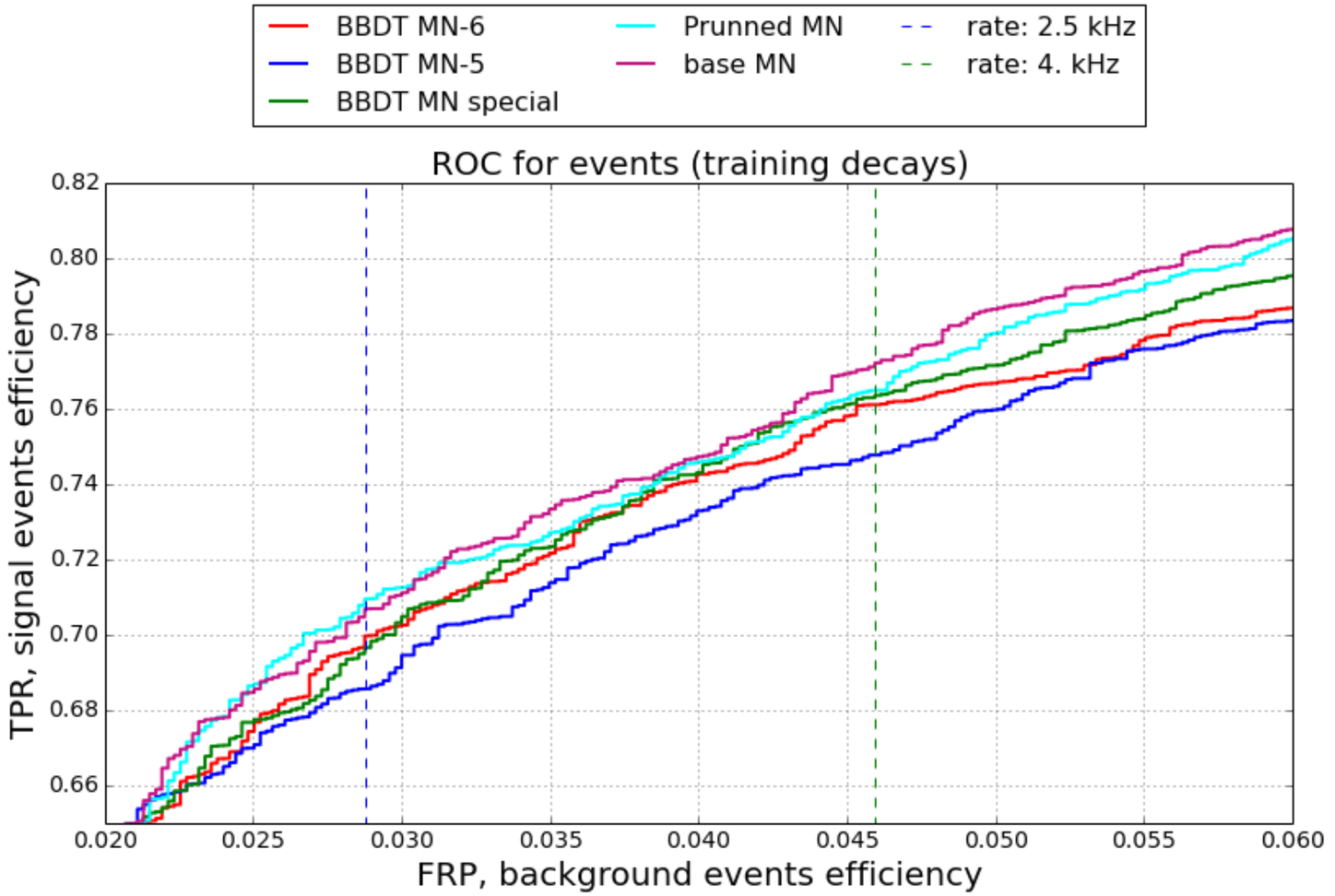
Post-pruning

- Train MatrixNet (MN) with several thousands trees
- Reduce this amount of trees to a hundred
- Greedily choose trees in a sequence from the initial ensemble to minimize a modified loss function:

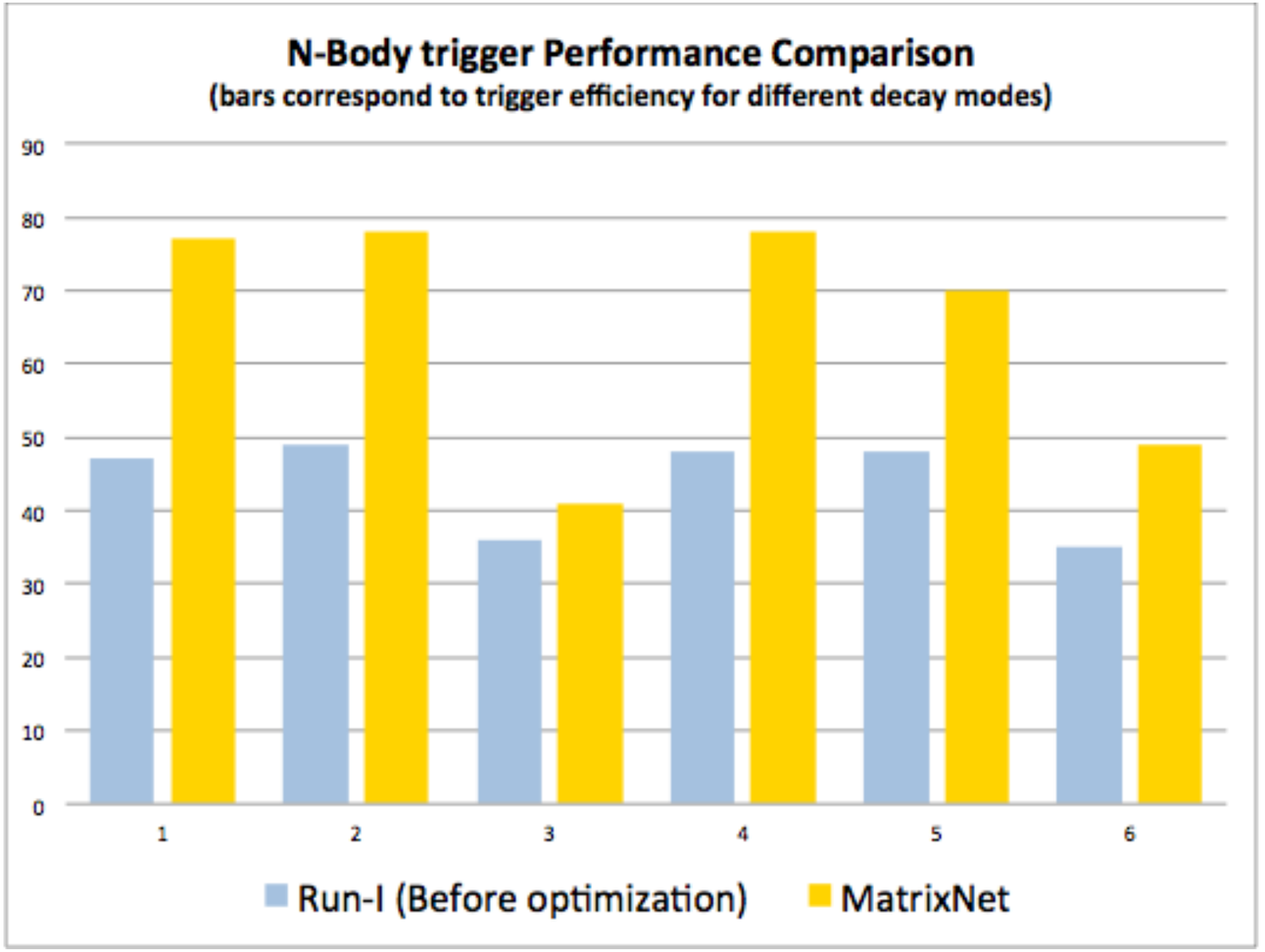
$$\sum_{signal} \log\left(1 + e^{-F(x)}\right) + \sum_{background} e^{F(x)}$$

- At the same time change values in leaves (tree structure is preserved)

Post-pruning, results



Trigger results



https://github.com/yandexdataschool/LHCb-topo-trigger

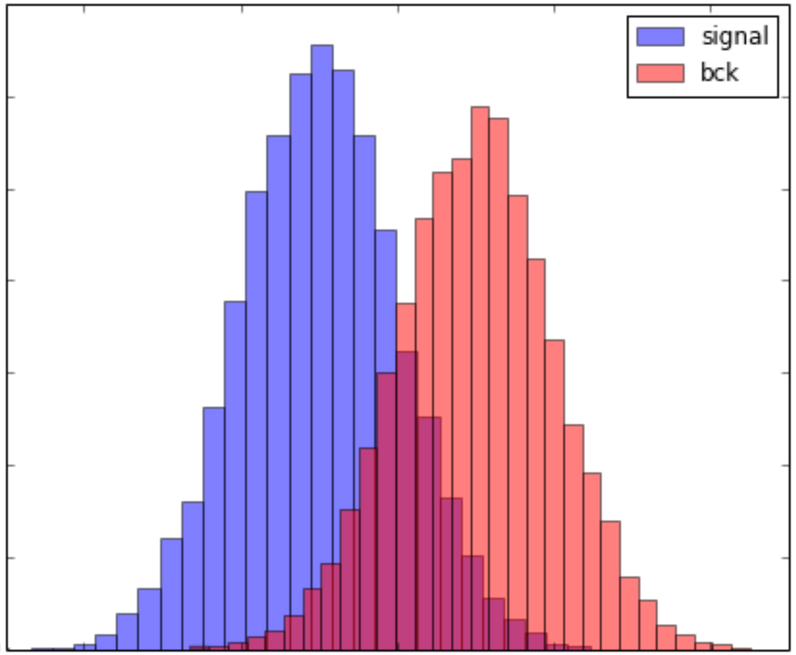
Machine learning in LHCb: triggering and flavour tagging systems

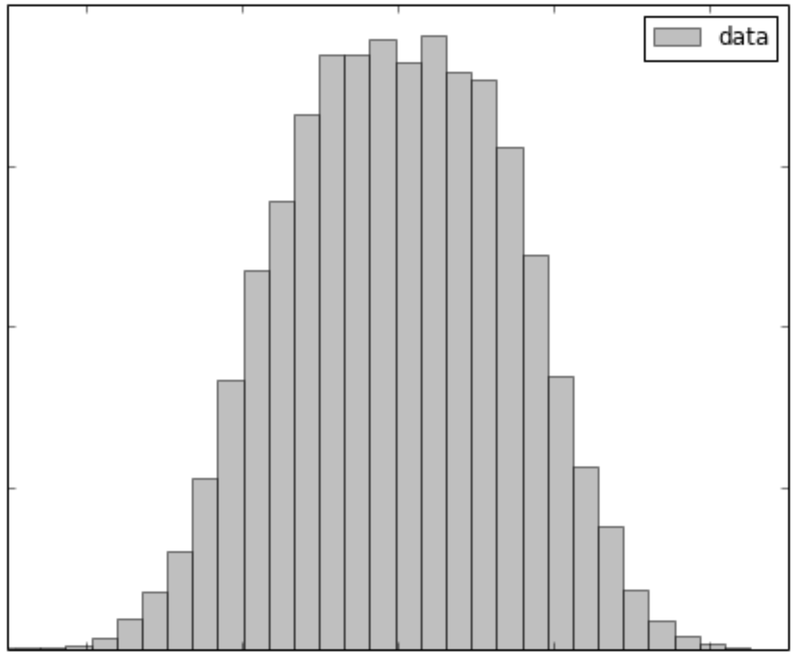
sPlot technique

Solution for what?

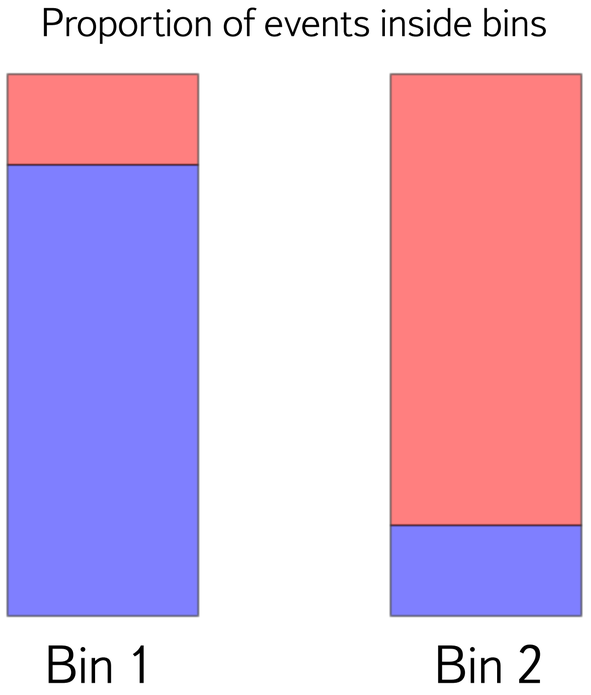
- Monte Carlo is not-well simulated
- Need to work with real unlabeled data
- Need somehow to label real data: want to restore for features their distributions for the signal and background data
- Our main knowledge is the mass distribution for real data from which we can extract (using some physics) the mass pdfs for signal and background.
- How to restore signal/bck pdfs for other features?

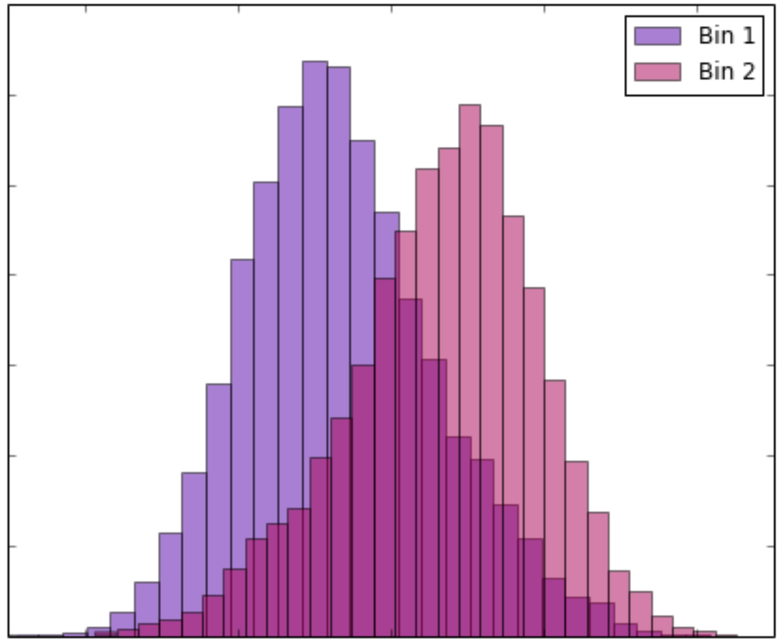
Feature initial distributions





Two mass bins



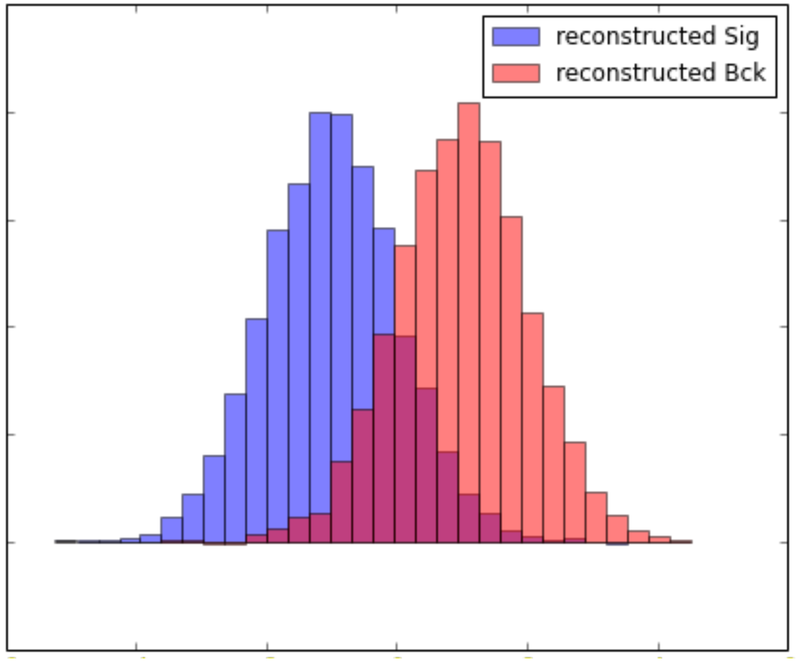


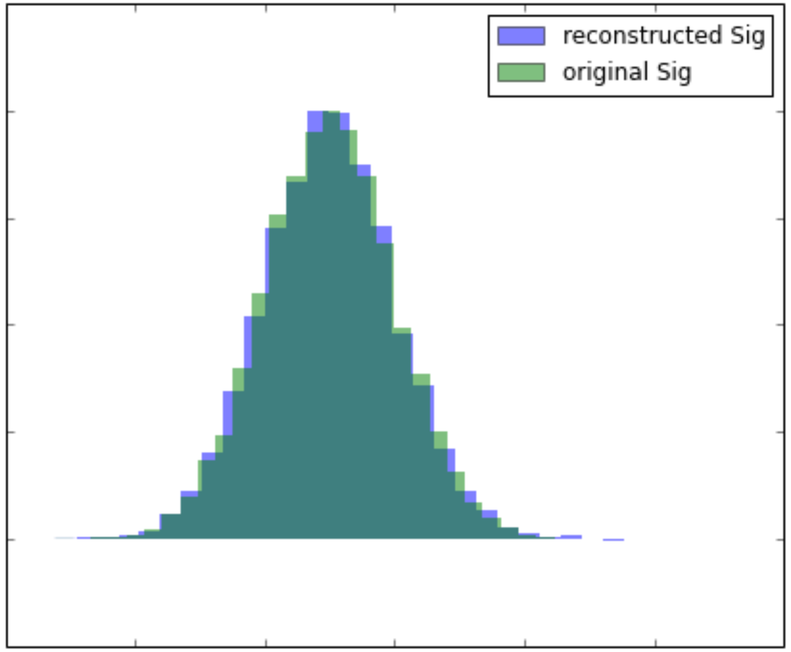
$$Bin1 : w_{b_1} f_b + w_{s_1} f_s \quad *w_{b_2}$$
$$Bin2 : w_{b_2} f_b + w_{s_2} f_s \quad *(-w_{b_1})$$

+

will obtain initial signal distribution

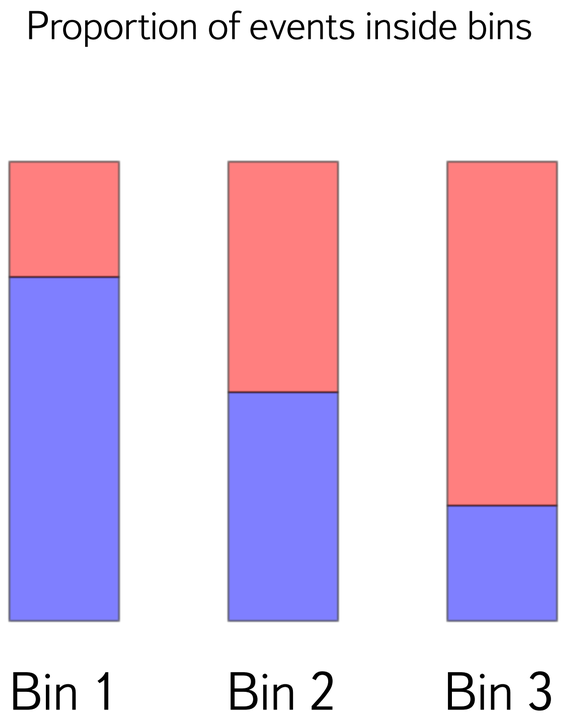
Reconstruction





More bins: sWeight

- Equivalent to some optimization problem
- Have explicit solution
- Produce weights (sWeight) for each event
- Feature pdf with sWeight will be signal pdf



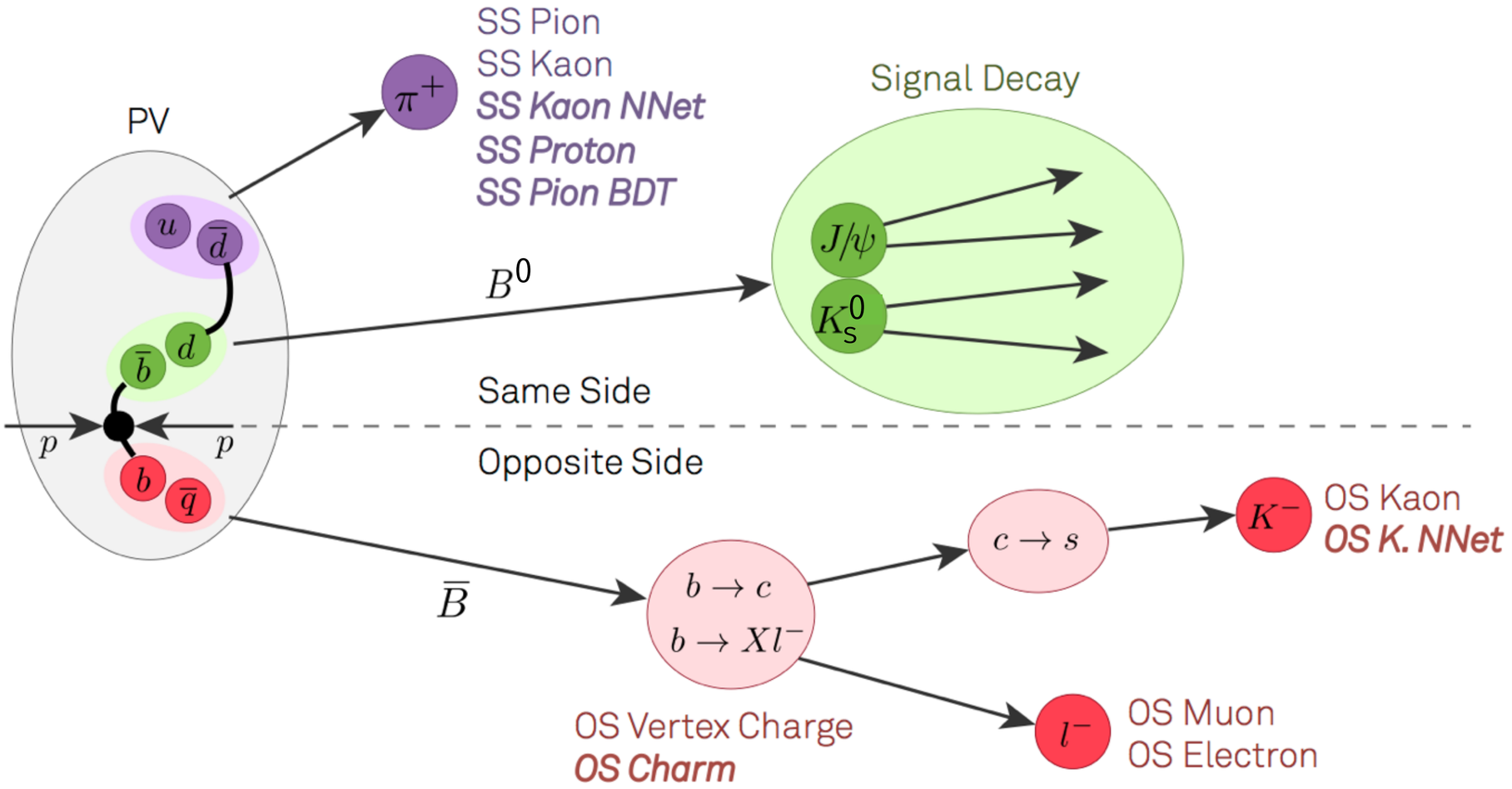
How to reweight?

Tagging system (not official)

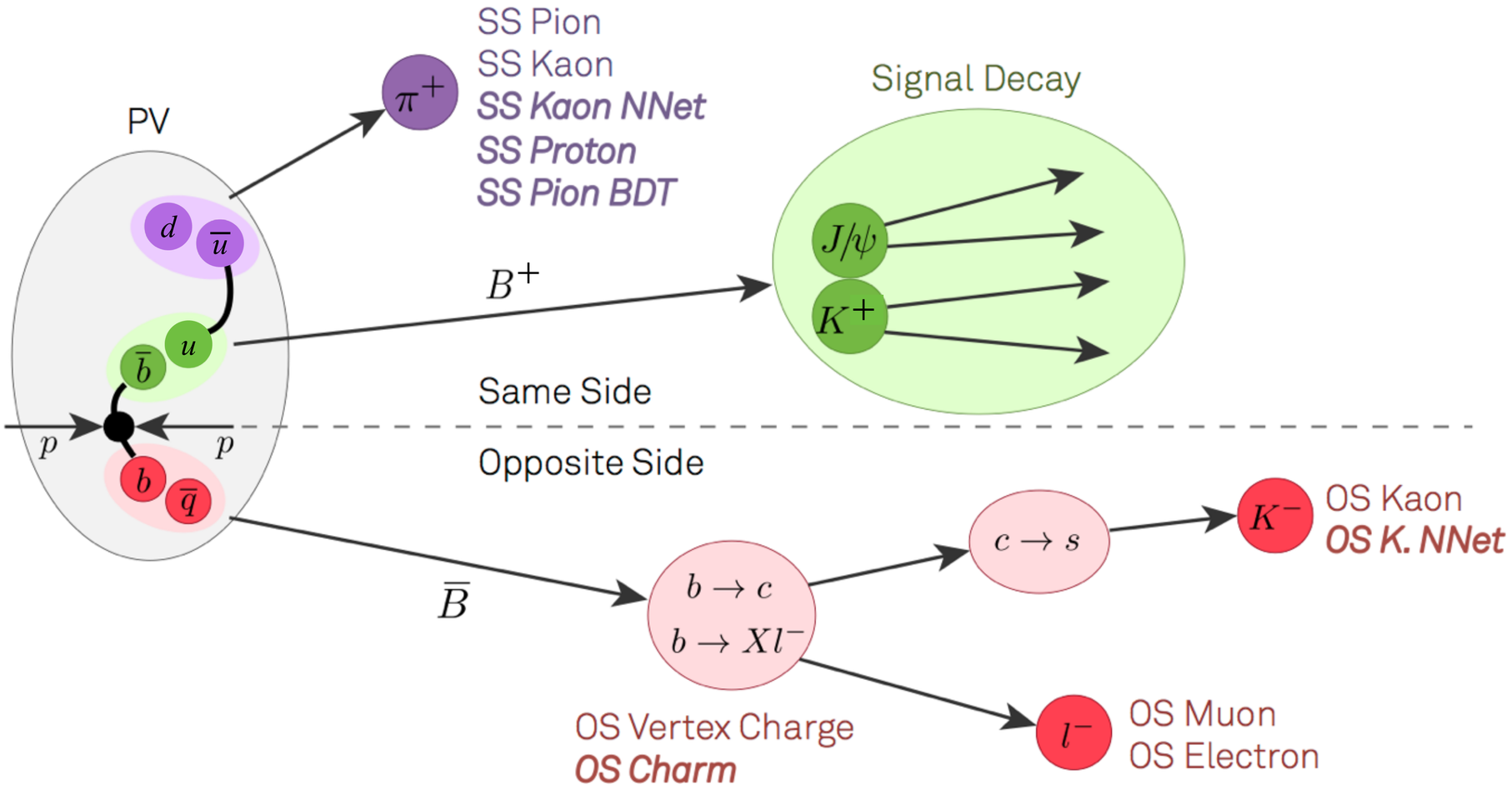
What is it?

- Event has a signal decay part
- The signal decay part can be produced from b quark or anti-b quark
- The system should effectively predict the source of the signal decay (b quark or anti-b quark)
- An intermediate B-meson in the signal decay part can oscillate
- The tagging system prediction P(anti-b quark) will allow to measure the oscillation effects

Tagging particles (goal)



Tagging particles (training)



Confidence interval to asymmetry

- Construct probabilities for b→final state and anti-b→final state using P(anti-b quark)
- Take relation of probabilities, called asymmetry
- Confidence interval for parameter of interest (A_m - measured):

$$\sigma_A = \frac{\sigma_{A_m}}{1-2\omega} = \frac{\sqrt{1-A_m^2}}{\sqrt{\varepsilon_{tag}N}(1-2\omega)}$$

- Tagging system should maximize effective efficiency (ω - mistag probability):

$$\varepsilon_{eff} = \varepsilon_{tag}D^2 = \varepsilon_{tag}(1-2\omega)^2$$

Current tagging system

- Take $B^{\pm} \rightarrow J/\psi\, K^{\pm}$
- Apply sPlot technique to real data to extract signal-like data (sWeights)
- Choose one tagging track (physical selections, like PID and max PT track) for each event
- Train exclusive taggers for SS (same side) tagging particles and OS (opposite side)
- Target for classifier is ‘right tagged’ label
- Combine all taggers to one (probabilistic model) to obtain P(anti-b quark)

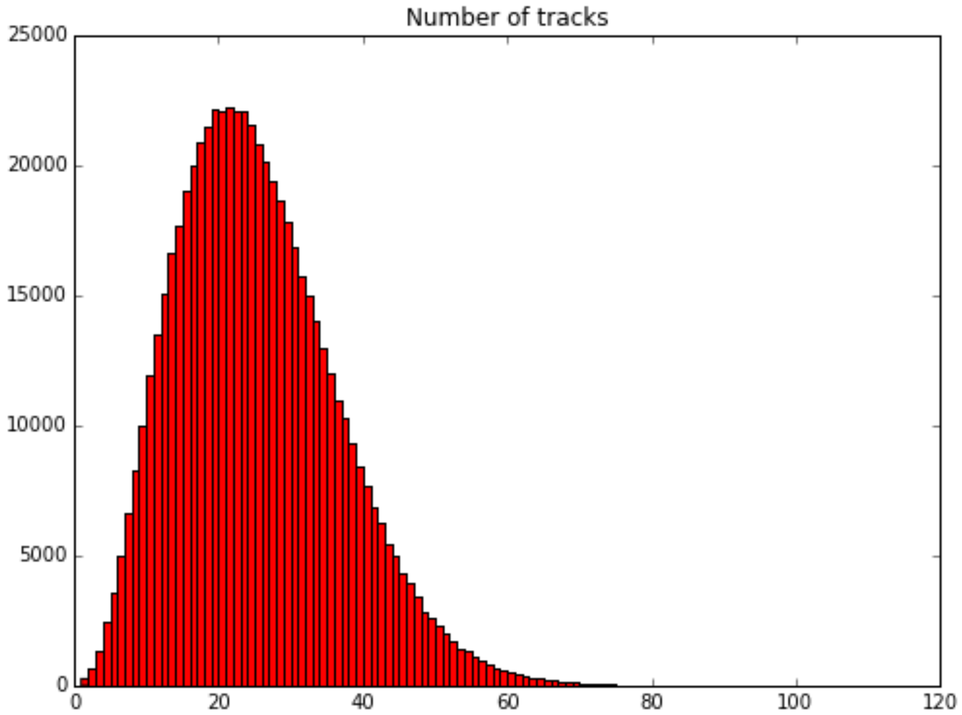
Inclusive tagging system (new ideas)

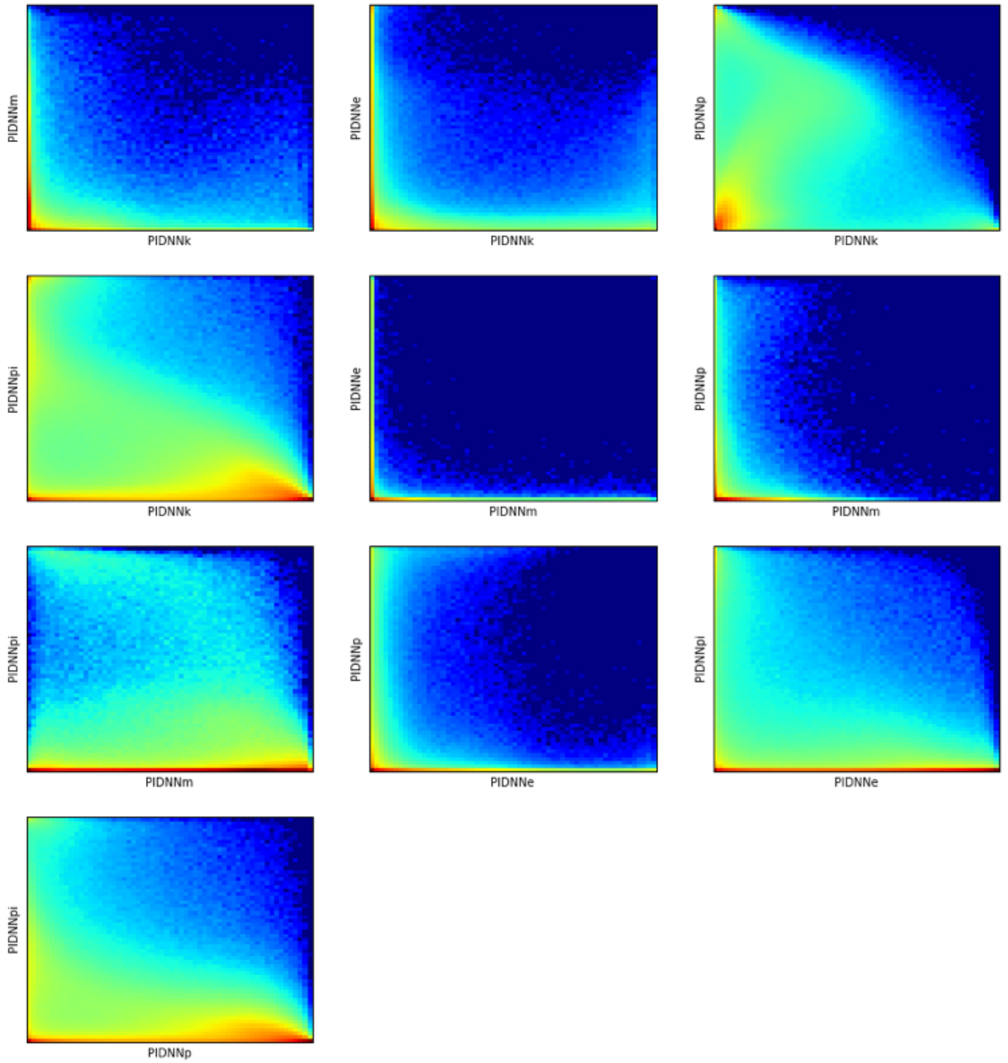
- Don’t apply physical selections, except loose cuts on PID and track ghost
- For each event use full topological information: all tracks and possible vertices
- Use probabilistic model for tracks and vertices to obtain P(anti-b quark)
- Maximize ROC AUC score

Restrictions

- Maximize effective efficiency
- Necessary to calibrate output to probability
- Distribution should be symmetry for b-quark (B^-) and anti-b quark (B^+)
- Flatness for B-mass, lifetime, momentum (to simplify further analysis)

Track-based inclusive tagging





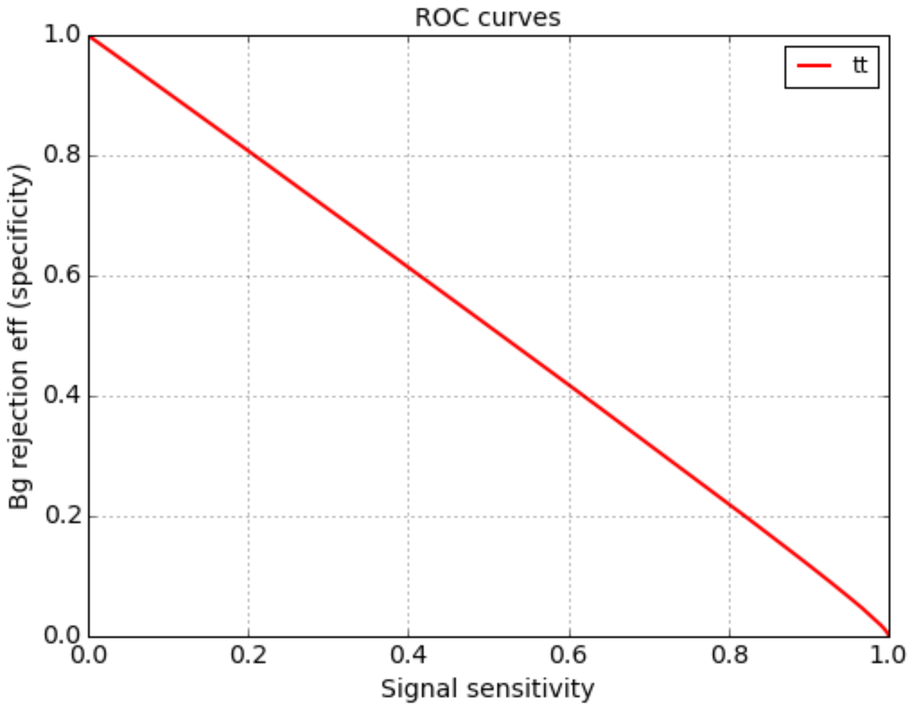
Track-based inclusive tagging

- Target is BSign*trackSign > 0 (avoid to define tagging particle)
- Classifier will return P(track has same sign as B | B sign)
- Suppose:

$$P(\text{track has same sign as B} \mid \text{B sign}) =$$

$$P(\text{B has same sign as track} \mid \text{track sign})$$

- AUC 0.5134
- Calibrate output to probability (isotonic calibration)



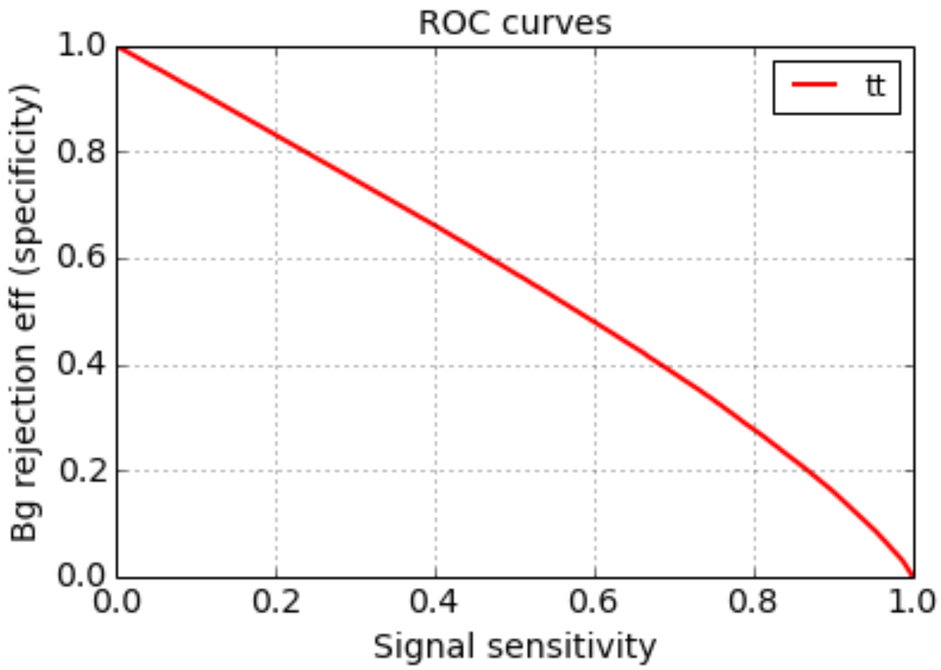
Vertex-based inclusive tagging

- Target is BSign*svSign > 0
- Classifier will return P(vertex has same sign as B | B sign)
- Suppose:

$$P(\text{vertex has same sign as B} \mid \text{B sign}) =$$

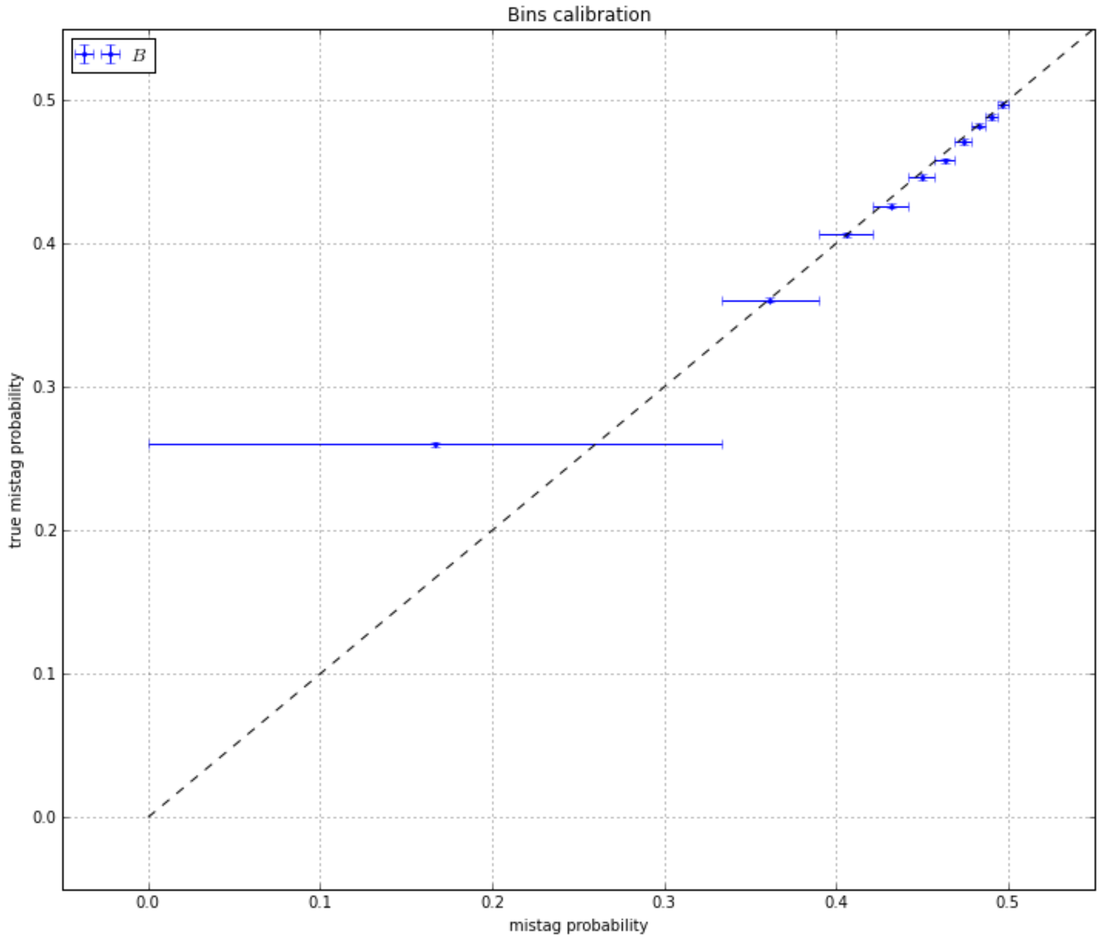
$$P(\text{B has same sign as vertex} \mid \text{vertex sign})$$

- AUC 0.5544
- Calibrate output to probability (logistic calibration)

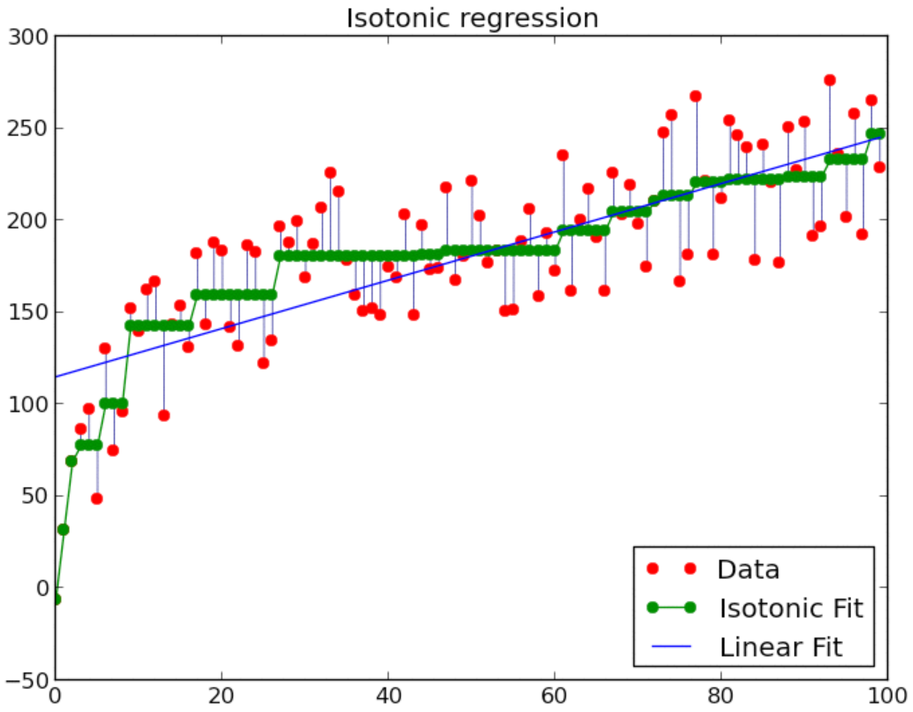


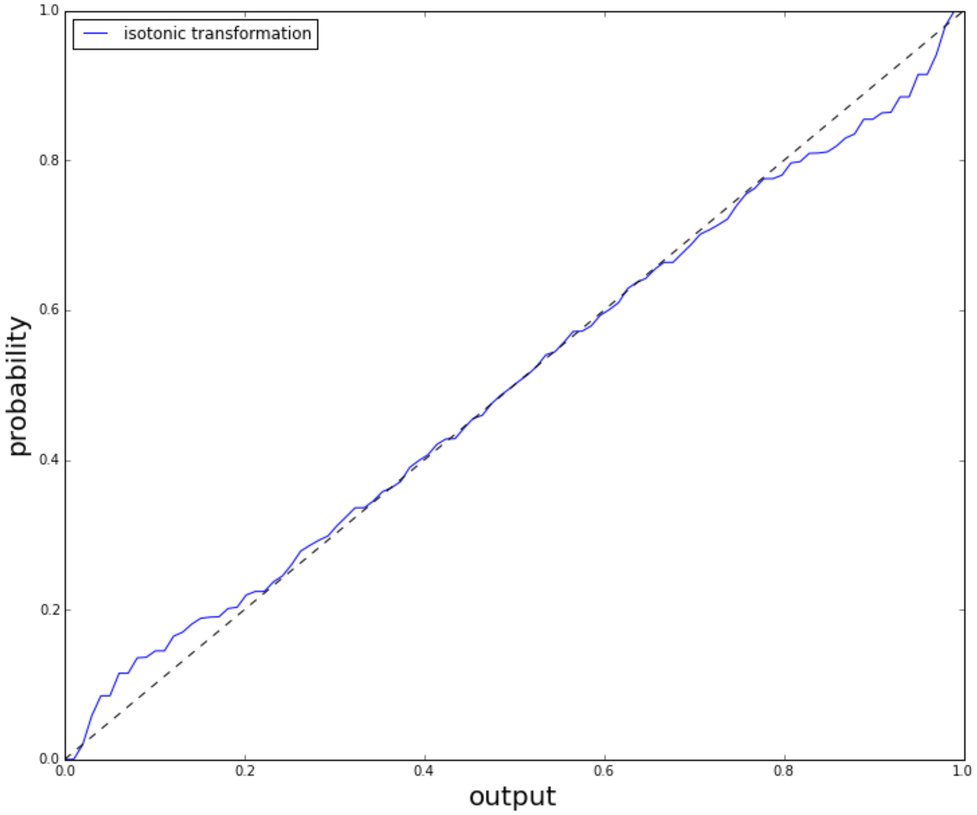
Calibration

- Platt's calibration: logistic regression over the classifier output
- Bin method:
  - #(same sign in bin) / #(in bin)
  - fit by linear function
- Isotonic regression: monotonic function (extend the bin method)



Isotonic calibration





Probabilistic model for events

$$\frac{P(B^+)}{P(B^-)} = \prod_{track,vertex} \frac{P(\text{track/vertex}|B^+)}{P(\text{track/vertex}|B^-)} = \alpha$$

$$\Rightarrow \qquad P(B^+) = \frac{\alpha}{1+\alpha}, \qquad [1]$$

where

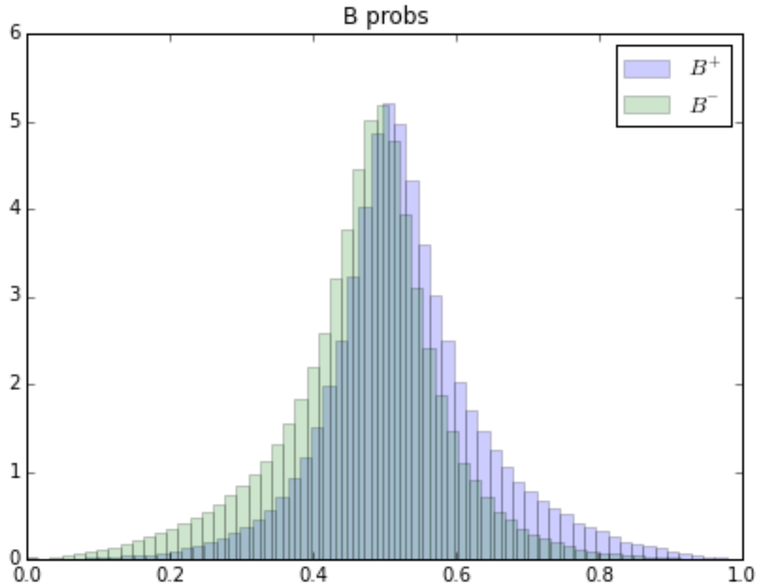
$$\frac{P(B^+)}{P(B^-)} = \prod_{track,vertex} \begin{cases} \dfrac{P(\text{track/vertex same sign as } B|B)}{P(\text{track/vertex opposite sign as } B|B)}, \text{ if track/vertex}^+ \\ \dfrac{P(\text{track/vertex opposite sign as } B|B)}{P(\text{track/vertex same sign as } B|B)}, \text{ if track/vertex}^- \end{cases}$$

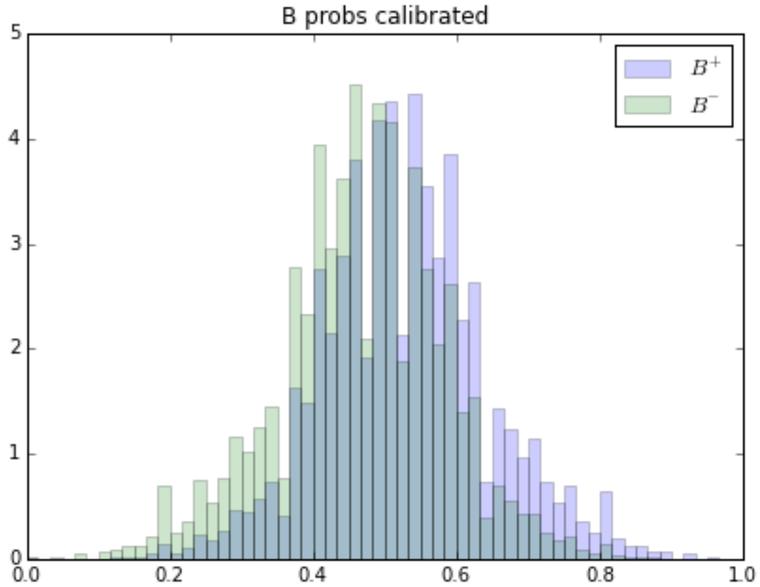
$$p_{mistag} = min(p(B^+), p(B^-))$$

anti-b quark probability

- Isotonic calibration
- Produced tied probabilities
- Add small normal noise

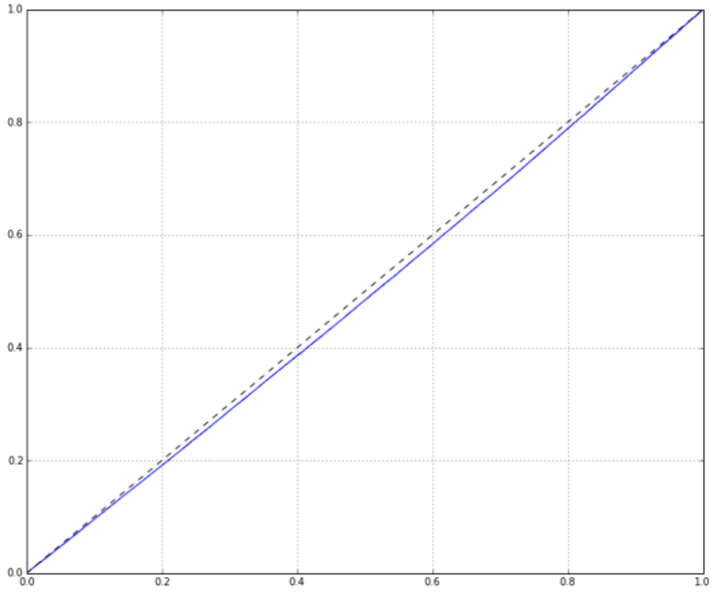
  0.001 * normal(0, 1)

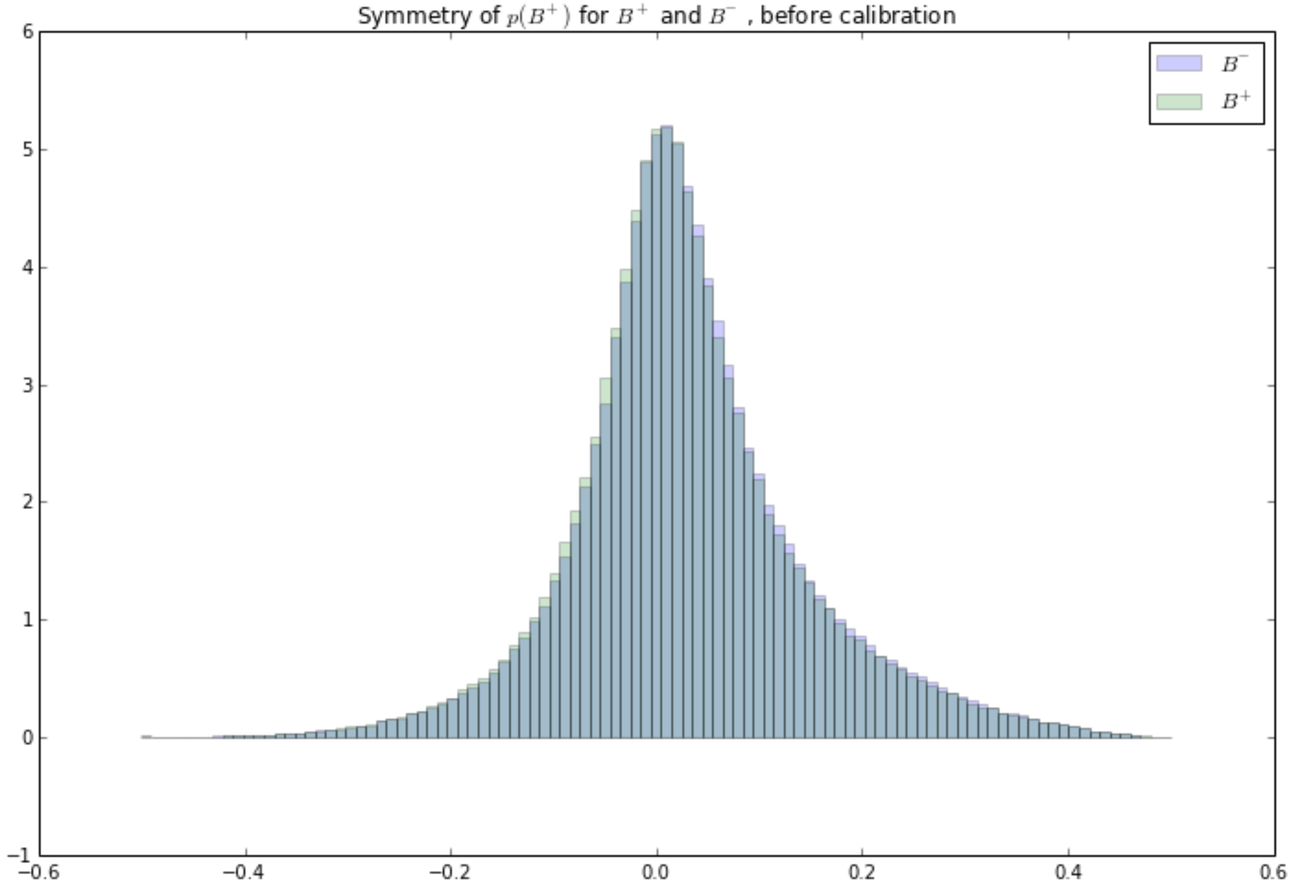




Check B symmetry (before calibration)

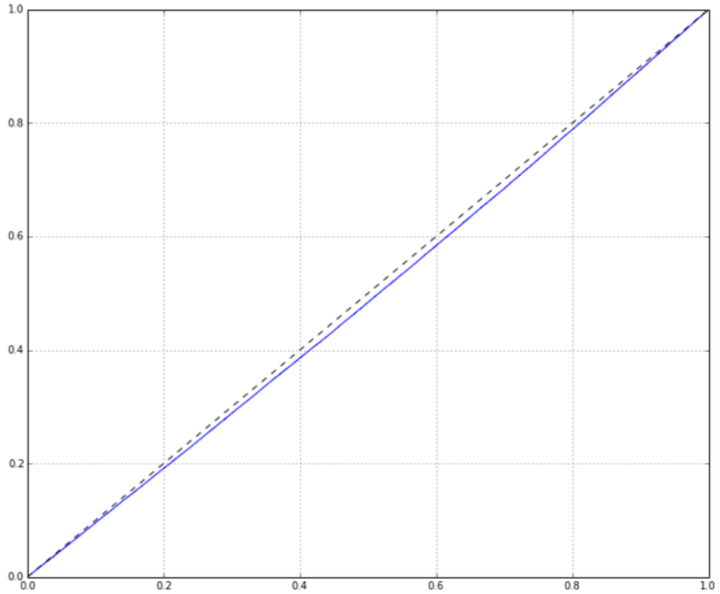
- KS: 0.0168
- compare PDFs using ROC curve
- AUC: 0.48949

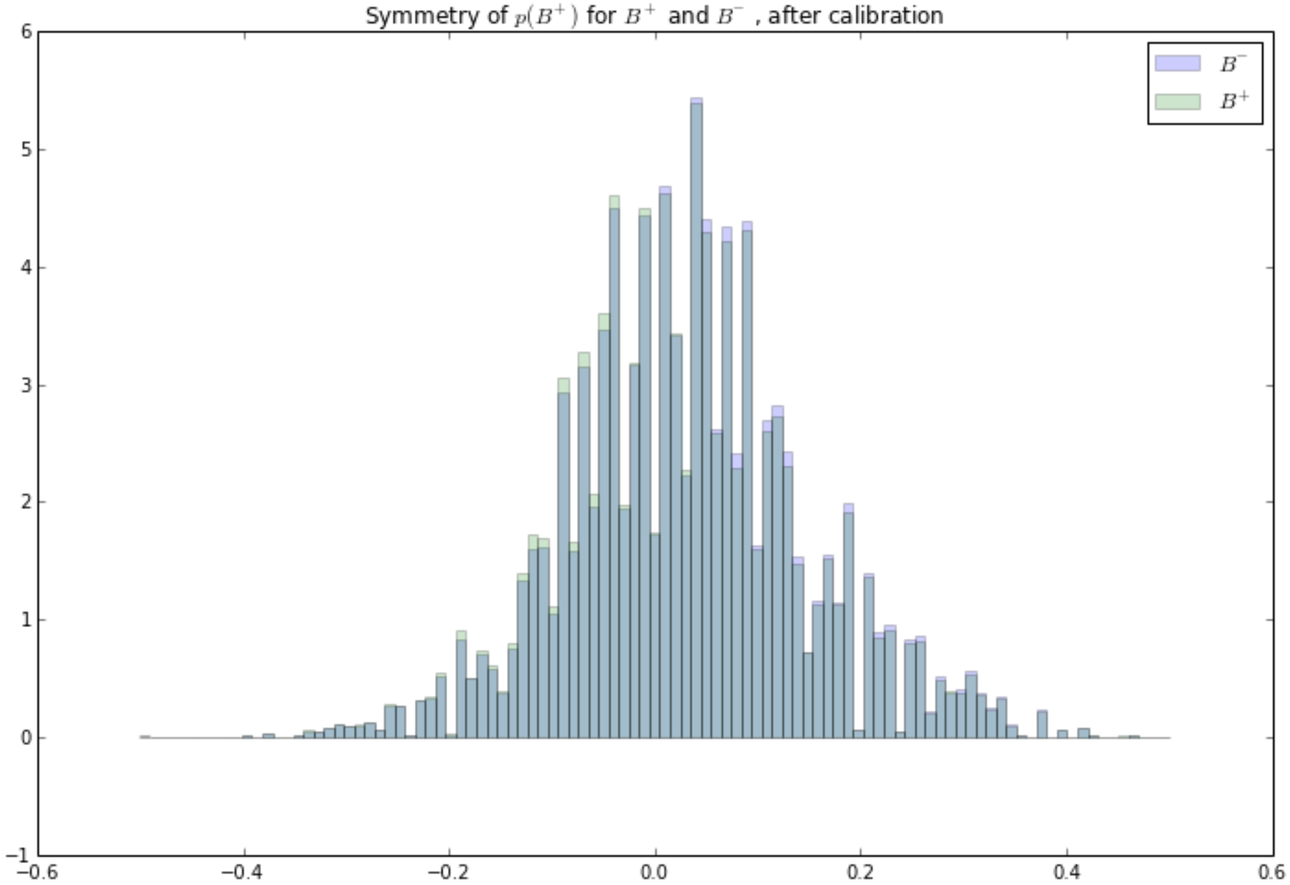




Check B symmetry (after calibration)

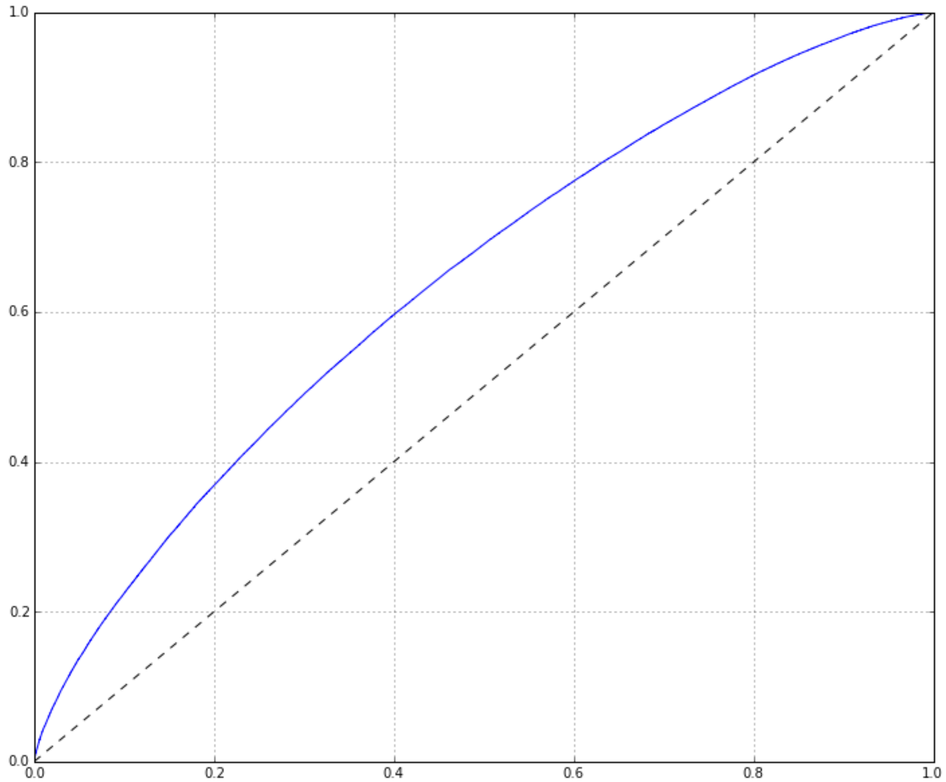
- KS: 0.01683
- compare PDFs using ROC curve
- AUC: 0.4895



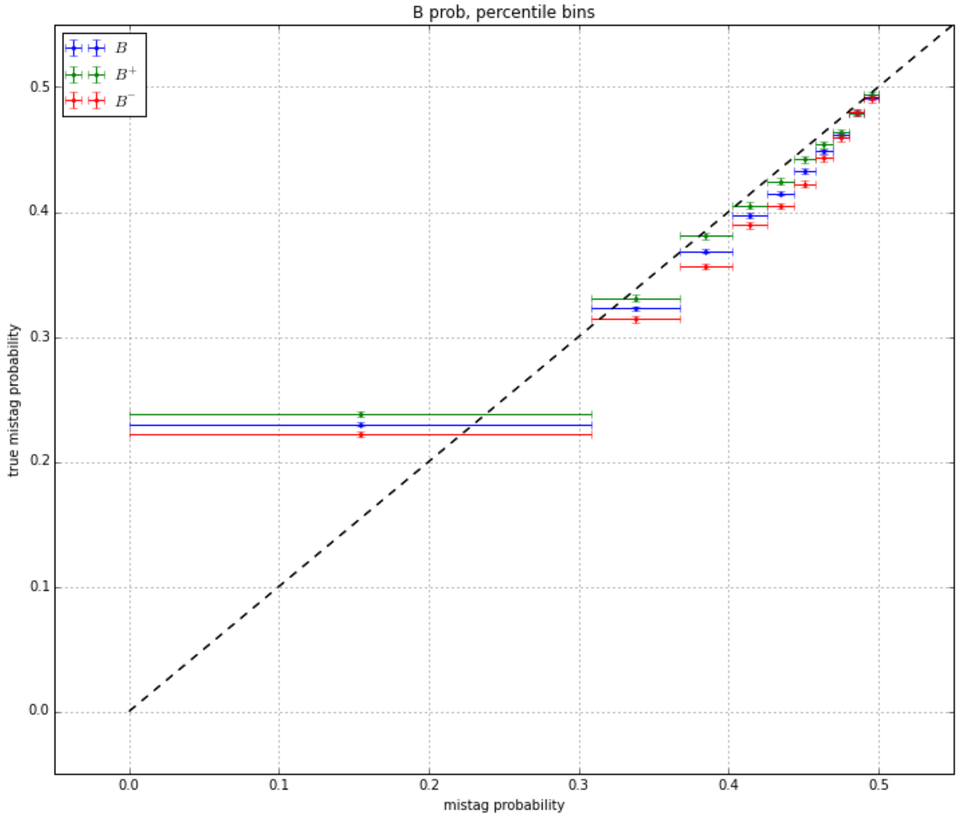


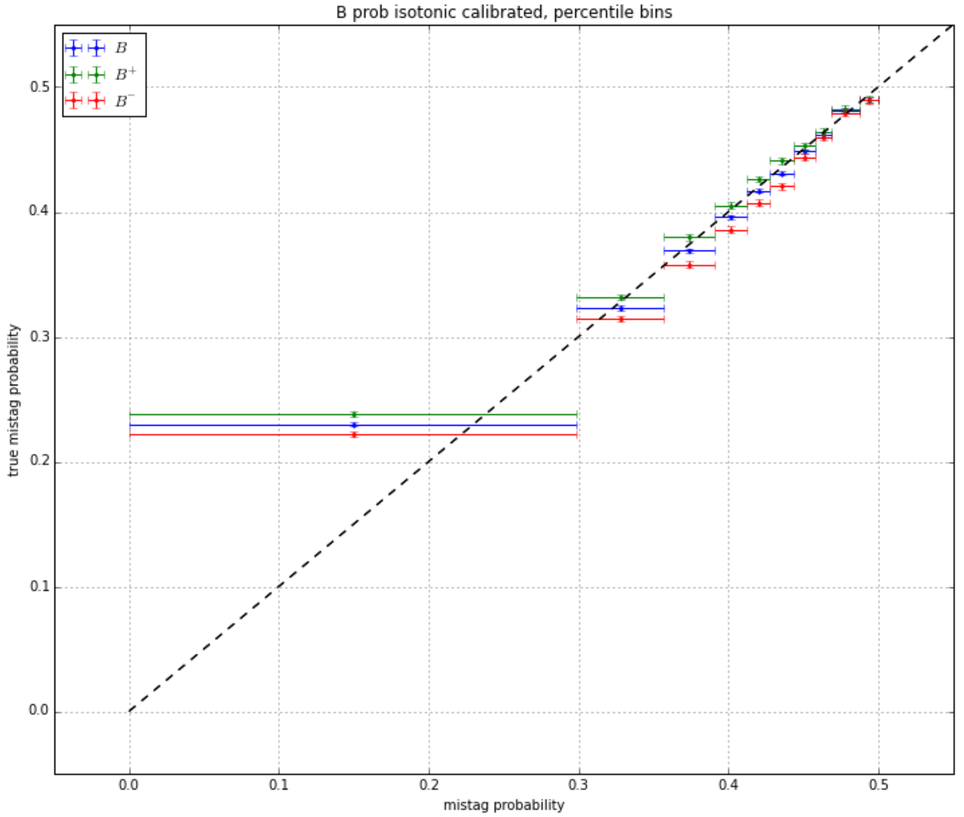
ROC for events (not official)

- AUC score 0.64 (for current tagging system 0.566)
- What about calibration?



Check calibration (with B-symmetry)





Thanks for attention

Contacts

Likhomanenko Tatiana
researcher-developer

antares@yandex-team.ru

References

- Trigger system: https://github.com/yandexdataschool/LHCb-topo-trigger, https://cdsweb.cern.ch/record/1384380/files/LHCb-PUB-2011-016.pdf, http://arxiv.org/abs/1510.00572
- Tagging system: https://github.com/tata-antares/tagging_LHCb, https://inspirehep.net/record/1381330/files/CERN-THESIS-2015-040.pdf
- Calibration: http://fastml.com/classifier-calibration-with-platts-scaling-and-isotonic-regression/, http://arxiv.org/pdf/1211.0025.pdf, http://arxiv.org/pdf/1511.00213
- sPlot technique: http://arxiv.org/pdf/physics/0402083

sPlot theory

After the mass maximum likelihood fitting we know for each event y

$$p_s(y), p_b(y)$$

which are probabilities of an event to be signal and background.

Will reconstruct number of *signal* events in a *particular* histogram bin for the reconstructed feature. Introduce unknown probability that the signal/bck event will be in a particular bin:

$$p_s, p_b$$

The total amount of signal/bck events obtained from the fit:

$$N_s, N_b$$

sPlot theory

The random variable, number of signal events in the bin:

$$X = \sum_y w_s(y) \mathbb{I}_{y \in bin}$$

where $w_s(y)$ are sPlot weights and are a subject to find.

Property: an estimation should be unbiased:

$$\mathbb{E}X = p_s N_s$$

Corollary:

$$p_s N_s = \mathbb{E}X = \sum_y w_s(y) \mathbb{E}\mathbb{I}_{y \in bin} =$$

$$= \sum_y w_s(y)(p_s p_s(y) + p_b p_b(y))$$

sPlot theory

Since the previous equation should hold for all possible p_s, p_b, we get two equalities:

$$p_s N_s = \sum_y w_s(y) p_s p_s(y)$$

$$0 = \sum_y w_s(y) p_b p_b(y)$$

after reduction:

$$N_s = \sum_y w_s(y) p_s(y)$$

$$0 = \sum_y w_s(y) p_b(y)$$

Then we can guarantee that mean input of background are 0 (the expectation is zero, but observed number will not be zero due to the deviation)

sPlot theory

Assumption of the linearity:

$$w_s(y) = a_1 p_b(y) + a_2 p_s(y)$$

Then:

$$\begin{pmatrix} V_{bb} & V_{bs} \\ V_{sb} & V_{ss} \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 \\ N_s \end{pmatrix}$$

where $V_{ij} = \sum_y p_i(y) * p_j(y)$

The assumption of the linearity is correct because apart from having correct mean, we should also minimize the variation of the reconstructed variable.

$$\mathbb{V}X = \sum_y w_s^2(y) \mathbb{V}\mathbb{I}_{y \in bin} \leq \sum_y w_s^2(y)$$

Minimization problem

$$\sum_y w_s^2(y) \to min$$

$$\sum_y w_s(y) p_b(y) = 0$$

$$\sum_y w_s(y) p_s(y) = N_s$$

Lagrangian:

$$\mathcal{L} = \sum_y w_s^2(y) + \lambda_1 \left(\sum_y w_s(y) p_b(y) \right) + \lambda_2 \left(\sum_y w_s(y) p_s(y) - N_s \right)$$

$$0 = \frac{\partial \mathcal{L}}{\partial w_s(y)} = 2 w_s(y) + \lambda_1 p_b(y) + \lambda_2 p_s(y)$$

It holds for each event, thus we are getting needed the linear dependence

Main assumption

The mass must be uncorrelated with the reconstructed feature:

$$p_s(mass, feature) = p_s(y, feature) = p_s(y)p_s(feature)$$

Then it holds for a particular bin for the reconstructed feature:

$$p_s(mass, bin) = p_s(y, feature) = p_s(y)p_s$$