New approaches for boosting to uniformity

Alex Rogozhnikov[a,b], Aleksandar Bukva[c], Vladimir Gligorov[d], Andrey Ustyuzhanin[b,e,f] and Mike Williams[g]

[a] Lomonosov Moscow State University, Moscow
[b] Yandex School of Data Analysis, Moscow
[c] Faculty of Physics, Belgrade
[d] Organisation Européenne pour la Recherche Nucléaire (CERN), Geneva
[e] Moscow Institute of Physics and Technology, Moscow
[f] Imperial College, London
[g] Massachusetts Institute of Technology, Cambridge

alex.rogozhnikov@yandex.ru

11 November, 2014

Outline

- What is uniformity (of predictions)?
- How to measure it? (metric functions)
- How to achieve it? (classifiers proposed)

Uniformity

In particle physics, apart from optimizing some FOM of classifier (BDT, ANN), there are cases when we need to have uniformity of predictions

- Dalitz-plot analysis (or any angular or amplitude analysis)
- search for a new particle (not to get fake peak)
- sensitivity for new signal in wide range of vars (mass, lifetime ...)

Uniform variables — variables, along which uniformity of selection is desired (Dalitz variables, mass variable).

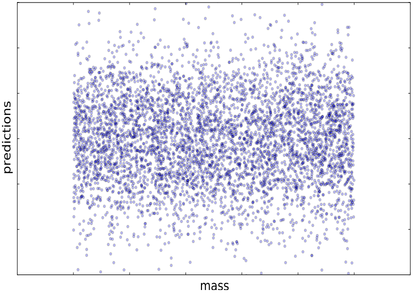
Typical solution: choose such features which don't give an ability to reconstruct 'mass' (or other selected 'uniform variables').

What is uniformity?

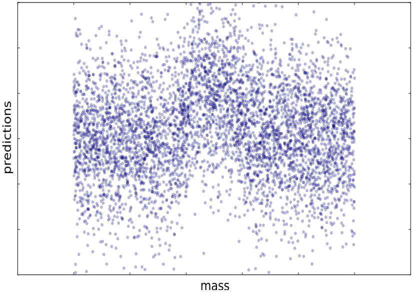
Predictions of some classifier are called *uniform* in variables $var_1, \ldots, var_n$ if prediction and set of this variables is *statistically independent*.
This (and only this) guarantees that any cut of prediction of classifier will produce the same efficiency in every region over $var_1, \ldots, var_n$

What is uniformity?

Predictions of some classifier are called *uniform* in variables $var_1, \ldots, var_n$ if prediction and set of this variables is *statistically independent*.
This (and only this) guarantees that any cut of prediction of classifier will produce the same efficiency in every region over $var_1, \ldots, var_n$



(a) Uniform predictions



(b) Non-uniform

Desirable properties of metrics

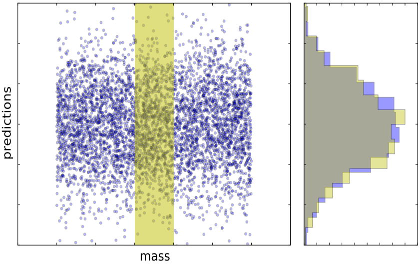
The metric should ...

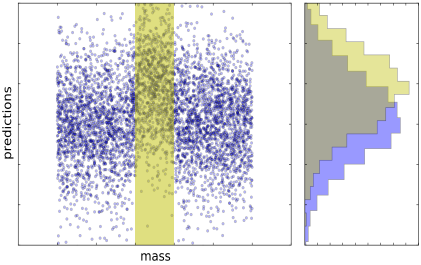
- not depend strongly on the number of events used to test uniformity
- not depend on the total weight
- depend on order of predictions, not the exact values of predictions (example: Pearson correlation does not satisfy this property)
- be stable against free parameters (number of bins, k in knn)

Similarity-based approach

Idea: uniformity means that distribution of predictions in every bin is equal.

Let's compare the global distribution (blue hist) with distibution in one bin (yellow hist). Yellow rectangle shows the events in selected bin over mass.





Similarity-based approach

Let $F(x)$ – cdf of all predictions, $F_{\text{bin}}(x)$ — cdf of predictions in bin over mass. Hereinfter $\text{weight}_{\text{bin}} = \frac{\text{weight of events in bin}}{\text{weight of all events}}$

Kolmogorov-Smirnov measure (uninformative)

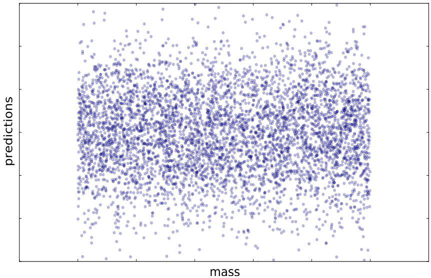
$$\sum_{\text{bin}} \text{weight}_{\text{bin}} \max_x |F_{\text{bin}}(x) - F(x)|,$$

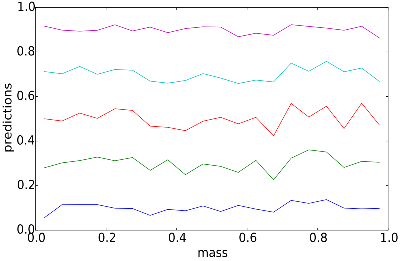
Cramér–von Mises similarity

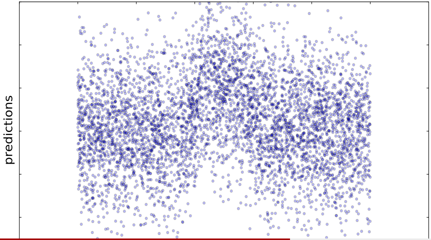
$$\sum_{\text{bin}} \text{weight}_{\text{bin}} \int |F_{\text{bin}}(x) - F(x)|^p \, dF(x)$$

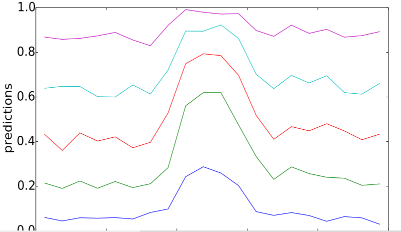
Cut-based approach (1/2)

Select some set of efficiencies (in examples: 0.1, 0.3, 0.5, 0.7, 0.9), for each one can compute global cut and look at efficincies in each bin:









Cut-based approach (2/2)

Standard deviation of efficiency

$$\text{SDE}^2(\text{eff}) = \sum_{\text{bin}} \text{weight}_{\text{bin}} \times (\text{eff}_{\text{bin}} - \text{eff})^2$$

$$\text{SDE}^2 = \frac{1}{k} \sum_{\text{eff} \in [\text{eff}_1 \dots \text{eff}_k]} \text{SDE}^2(\text{eff})$$

Theil index of $x_1, \dots, x_n$

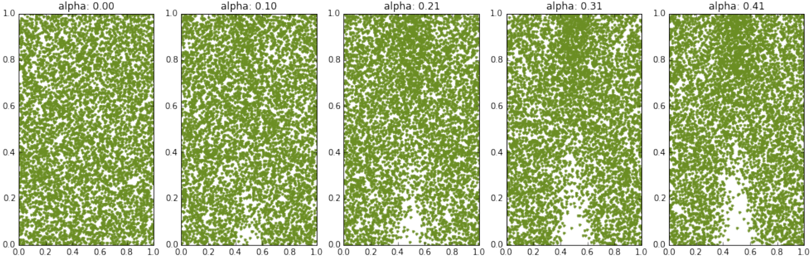
$$\text{Theil} = \frac{1}{N} \sum_i \frac{x_i}{< x >} \ln \frac{x_i}{< x >},$$

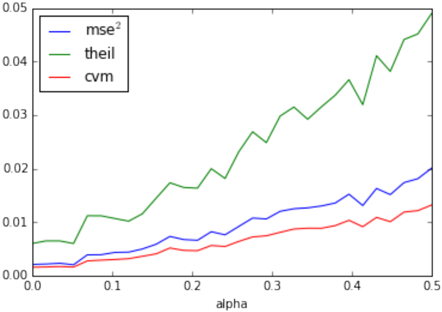
Theil index of efficiency

$$\text{Theil(eff)} = \sum_{\text{bin}} \text{weight}_{\text{bin}} \frac{\text{eff}_{\text{bin}}}{\text{eff}} \ln \frac{\text{eff}_{\text{bin}}}{\text{eff}}$$

$$\text{Theil} = \frac{1}{k} \sum_{\text{eff} \in [\text{eff}_1 \dots \text{eff}_k]} \text{Theil(eff)}.$$

Example


alpha: 0.00
alpha: 0.10
alpha: 0.21
alpha: 0.31
alpha: 0.41


mse²
theil
cvm
alpha

Summary on metrics

1. two basic approaches were introduced (distribution-based and cut-based)
2. despite their difference, the results obtained with metrics proposed are **similar**.
3. for higher dimensions: $k$nn modifications of metrics are available (instead of binning over uniform variables, we can compute nearest neighbours in the space of uniform variables).

Boosting to uniformity

Previuos work: uBoost

J. Stevens and M. Williams, *uBoost: A boosting method for producing uniform selection efficiencies from multivariate classifiers*, JINST **8**, P12013 (2013). [arXiv:1305.7248]

The classifiers we propose alter the **boosting** procedure as well

Boosting: *knn*AdaBoost

Usual AdaBoost reweighting procedure (p_i is prediction of last classifier):

$$w'_i = w_i \times \exp[-y_i \, p_i],$$

*knn*AdaBoost uses mean of predictions of neighbours

$$w_i = w_i \times \exp[-y_i \, \frac{1}{k} \sum_{j \in \mathrm{knn}(i)} p_j]$$

(neighbours are of the same class).
Thus boosting focuses not on the events that were poorly classified, but on the regions with poor classification.

Boosting: Gradient Boosting with *knn*AdaLoss (1/2)

Gradient boosting on trees is widely used algorithm, it's built upon decision tree regressors with usage of some loss function.
Usual AdaLoss:

$$L_{\text{ada}} = \sum_{i \in \text{events}} w_i \times \exp[-\text{score}_i \, y_i]$$

Pseudo-residual of AdaLoss:

$$-\frac{\partial L_{\text{ada}}}{\partial \, \text{score}_i} = w_i \, y_i \exp[-\text{score}_i \, y_i],$$

*knn*AdaLoss:

$$L_{\text{knn-ada}} = \sum_{i \in events} \exp[-y_i \times \sum_{j \in \text{knn}(i)} \text{score}_j],$$

Boosting: Gradient Boosting with *knn*AdaLoss (2/2)

*knn*AdaLoss:

$$L_{\text{knn-ada}} = \sum_{i \in events} \exp[-y_i \times \sum_{j \in \text{knn}(i)} \text{score}_j],$$

It can be written as particular case of:

$$L_{\text{general}} = \sum_i \exp[-y_i \sum_j a_{ij} \, \text{score}_j],$$

$$a_{ij} = \begin{cases} 1, & j \in \text{knn}(i), \text{ events } i \text{ and } j \text{ belong to the same class} \\ 0, & \text{otherwise,} \end{cases}$$

This is one particular choice of a_{ij};
in general case matrix a_{ij} even may be non-square.

Boosting: Gradient Boosting with FlatnessLoss (uGBFL)

CvM measure of non-uniformity:

$$\sum_{\text{bin}} \text{weight}_{\text{bin}} \int |F_{\text{bin}}(x) - F(x)|^p \, dF(x),$$

Let's modify this function:

$$\text{FL} = \sum_{\text{bin}} \text{weight}_{\text{bin}} \int |F_{\text{bin}}(x) - F(x)|^p \, dx$$

so that it becomes differentiable

$$\frac{\partial}{\partial \, \text{score}_i} \text{FL} \cong w_i \, p \, \left| F_{\text{bin}(i)}(x) - F(x) \right|^{p-1} \text{sgn}[F_{\text{bin}(i)}(x) - F(x)] \Big|_{x = \text{score}_i}$$

Boosting: Gradient Boosting with FlatnessLoss (uGBFL)

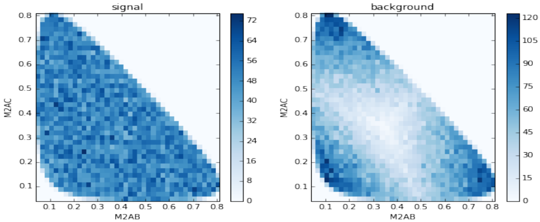
FL doesn't take into account the quality of predictions, only uniformity. So what we use in practice is linear combination of FlatnessLoss and AdaLoss:

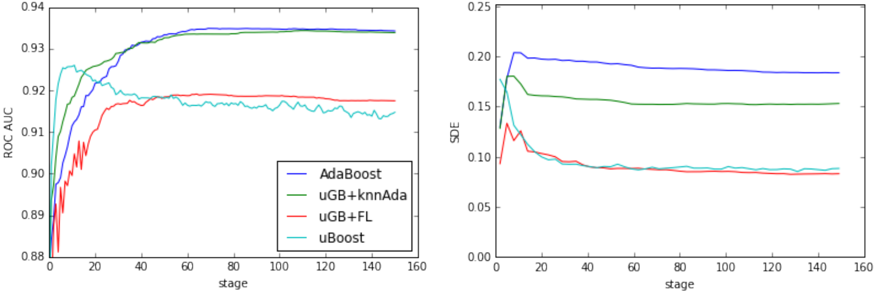
$$\text{loss} = \text{FL} + \alpha \, L_{\text{ada}}$$

First one penalizes non-uniformity, second one — poor predictions, α is usually taken small.

Tests on Dalitz data

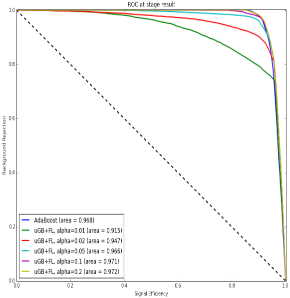
Testing on dataset from paper about uBoost



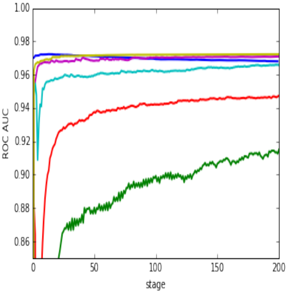


Tradeoff uniformity vs quality

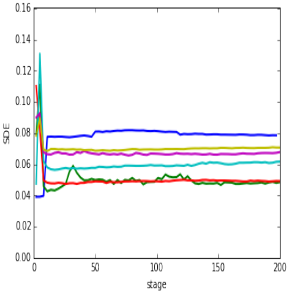
In uGBFL we can choose different values of *alpha* thus adjusting quality/uniformity.



(a) ROC curves



(b) ROC vs #trees (greater is better)



(c) SDE vs #trees (less is better)

Summary on classifiers

New classifiers

- Faster (than uBoost)
- Introduces classifiers can target at uniformity in *both* signal and bck
- *knn*AdaBoost and uGB + knnAdaLoss can be easily implemented, but don't seem to produce good uniformity
- uGBFL is highly tunable and proved to be able to fight severe correlation

Read:
http://arxiv.org/abs/1410.4140

Try out (python implementation):
https://github.com/anaderi/lhcb_trigger_ml

Q&A