ThickBrick: optimal event selection and categorization in high energy physics

Prasanth Shyamsundar, Fermilab Quantum Institute

IML Machine Learning Working Group Meeting

July 13, 2021

based on work with

Prof. Konstantin T. Matchev

ThickBrick: optimal event selection and categorization in high energy physics

Part 1: Signal discovery [arXiv:1911.12299]

Part 2: Parameter measurement [future]

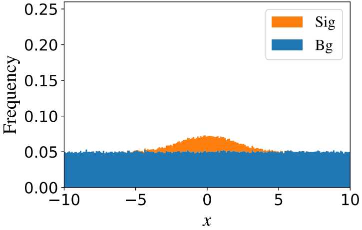
Part 3: Systematic uncertainties [future]

Also... ThickBrick package

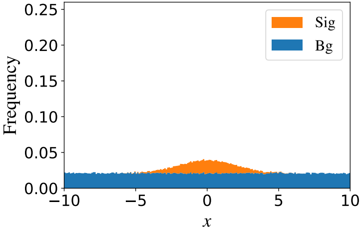
https://prasanthcakewalk.gitlab.io/thickbrick/

Introduction

- Event selection and/or categorization - an important step in any collider data analysis.
- Improves sensitivity by reducing the amount of “background” and makes data more “signal” rich.
- The “signal is better than background” heuristic has paved the way for ML techniques in event selection.



↓



Introduction

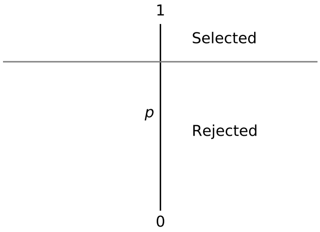
A straight forward ML approach to event selection:

- Train a classifier to distinguish between signal and background events.
- Use an appropriately chosen threshold on classifier output.

$$p(\boldsymbol{e}) \sim \frac{s(\boldsymbol{e})}{s(\boldsymbol{e}) + b(\boldsymbol{e})}$$

$$0 \leq p \leq 1$$

$\boldsymbol{e}$ is the feature vector



Introduction

- This approach is not perfectly aligned with the physics goals, namely
  - improve significance of a potential excess
  - improve the precision in parameter measurement

    (taking into account systematic uncertainties, in both cases)
- The presence of such a misalignment is well established.

  If you're not training to optimize physics goals directly, there's no reason to believe physics goals will be optimized.
- The source of misalignment is not well understood.

Introduction

Rectifying the misalignment

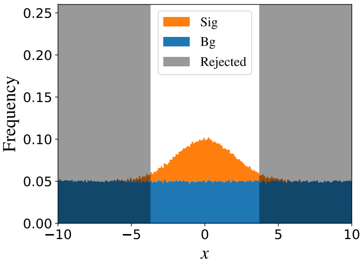
Previous attempts	Our approach
▶ Classify a (mini) batch of training data $\rightarrow$ perform analysis. ▶ Use the sensitivity of the analysis (signal significance or measurement uncertainty) as measure of performance of the classifier used. ▶ Train classifier based on this performance measure. This approach has its difficulties. S. Whiteson and D. Whiteson, 2009 A. Elwood and D. Krücker [1806.00322]	▶ Understand the sources of misalignment at an information-theoretic level. ▶ Rectify them and make training possible within the traditional ML techniques on an event-by-event basis.

Part 1:
Signal discovery

Task: Over all possible event selectors, find the one that maximizes the expected signal significance (statistical for now)

Source of misalignment: Intuitive outlook

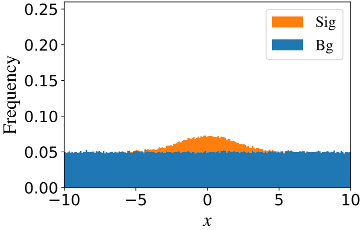
- Cutting based on the event variable $\boldsymbol{x}$ doesn't help. If anything, we lose sensitivity by losing bins.
- Background needs to be removed "from below", using information in $\boldsymbol{e}$ complementary to $\boldsymbol{x}$.
- $p(\boldsymbol{e})$ and $\boldsymbol{x}(\boldsymbol{e})$ have overlapping information. Especially if $\boldsymbol{x}$ is a "good" event variable. The result...



Not useful

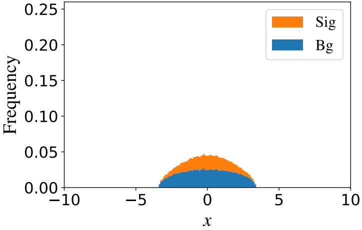
Source of misalignment: Intuitive outlook

- Cutting based on the event variable x doesn't help. If anything, we lose sensitivity by losing bins.
- Background needs to be removed "from below", using information in e complementary to x.
- $p(e)$ and $x(e)$ have overlapping information. Especially if x is a "good" event variable. The result...
- **Compromise between gain** in sensitivity from using complementary information in e **and loss** from using non-complementary information.
- Additional effect: Background shaping. Doesn't introduce bias, but could worsen the impact of systematic uncertainties.



↓

Cuts from below **and from the sides**



Source of misalignment: Information-theoretic outlook

How/why does event selection/categorization help?

- Consider two boxes of phase space, with (S_1, B_1) and (S_2, B_2) expected sig and bg events respectively.
- The only information we are provided is how many events were observed in each box.
- Some measures of sensitivity of the experiment to the presence of signal:

$$\sum_{i=1}^{2} \frac{S_i^2}{B_i}, \qquad \sum_{i=1}^{2} \frac{S_i^2}{N_i}, \qquad \sum_{i=1}^{2} \left[-S_i + N_i \ln\left(\frac{N_i}{B_i}\right)\right]$$

- Let the two boxes be mixed and analyzed together... information loss...
$S_{\text{tot}} = S_1 + S_2,\ B_{\text{tot}} = B_1 + B_2$
- $\frac{S_{\text{tot}}^2}{B_{\text{tot}}} \leq \sum_{i=1}^{2} \frac{S_i^2}{B_i}, \quad \frac{S_{\text{tot}}^2}{N_{\text{tot}}} \leq \sum_{i=1}^{2} \frac{S_i^2}{N_i}, \quad \sum_{i} \left[-S_{\text{tot}} + N_{\text{tot}} \ln\left(\frac{N_{\text{tot}}}{B_{\text{tot}}}\right)\right] \leq \sum_{i=1}^{2} \left[-S_i + N_i \ln\left(\frac{N_i}{B_i}\right)\right]$

Source of misalignment: Information-theoretic outlook

How/why does event selection/categorization help?

- Mixing regions of phase-space with different S/B (or S/N) values causes loss of sensitivity.
- Mixing regions of $\boldsymbol{e}$ with different values of $p(\boldsymbol{e})$ causes loss of sensitivity.
- Reducing $\boldsymbol{e} \rightarrow \boldsymbol{x}$ causes such a mixing.
- Event categorization helps by separating regions of phase-space **that would otherwise be mixed**.

$$\frac{S_{\text{tot}}^2}{B_{\text{tot}}} \longrightarrow \frac{S_1^2}{B_1} + \frac{S_2^2}{B_2}$$

Event selection helps by removing some regions of phase-space **that would otherwise mix** with other regions and worsen the sensitivity.

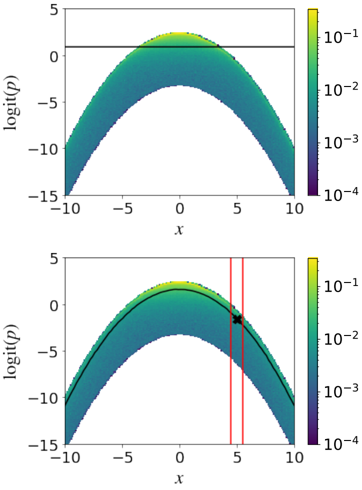
$$\frac{S_{\text{tot}}^2}{B_{\text{tot}}} \longrightarrow \frac{S_1^2}{B_1}$$

- Why separate/remove regions that aren't going to mix in the first place?

The fix: Bin dependent cut on $p(\boldsymbol{e})$

- A cut on $p(\boldsymbol{e})$ can be used to maximize $\frac{S^2}{B}\left(\text{or } \frac{S^2}{N}, \text{ etc}\right)$. (Neyman–Pearson Lemma)
- An $\boldsymbol{x}$ dependent cut on $p(\boldsymbol{e})$ can be used to maximize $\frac{s^2(\boldsymbol{x})}{b(\boldsymbol{x})}$ at each value of $\boldsymbol{x}$.
- Sensitivity $\sim \int d\boldsymbol{x} \frac{s^2(\boldsymbol{x})}{b(\boldsymbol{x})} \sim \sum_{i \in \boldsymbol{x} \text{ bins}} \frac{s_i^2}{b_i}$
- The cut at a given value of $\boldsymbol{x}$ only depends on the distribution at that value of $\boldsymbol{x}$, ensuring complementarity.
- Guiding principle: "Make the most out of each bin."
- How to derive these optimal $\boldsymbol{x}$ dependent cuts?

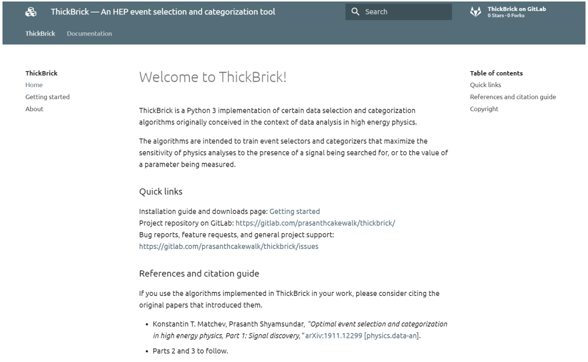
  Subject of a longer talk. Short answer...



ThickBrick https://prasanthcakewalk.gitlab.io/thickbrick/

- Input: Training data with $p(\boldsymbol{e})$ and $x(\boldsymbol{e})$ for each event. $p(\boldsymbol{e})$ could be learned using current ML techniques.
- Output: Optimal $\boldsymbol{x}$ dependent thresholds on $p(\boldsymbol{e})$ to maximize any of the following performance measures.

  Note: None of these can be written as sum of event-wise profit functions.


ThickBrick — An HEP event selection and categorization tool
Search
ThickBrick on GitLab
0 Stars · 0 Forks
ThickBrick Documentation

ThickBrick
Home
Getting started
About

Welcome to ThickBrick!

ThickBrick is a Python 3 implementation of certain data selection and categorization algorithms originally conceived in the context of data analysis in high energy physics.

The algorithms are intended to train event selectors and categorizers that maximize the sensitivity of physics analyses to the presence of a signal being searched for, or to the value of a parameter being measured.

Quick links

Installation guide and downloads page: Getting started
Project repository on GitLab: https://gitlab.com/prasanthcakewalk/thickbrick/
Bug reports, feature requests, and general project support: https://gitlab.com/prasanthcakewalk/thickbrick/issues

References and citation guide

If you use the algorithms implemented in ThickBrick in your work, please consider citing the original papers that introduced them.

- Konstantin T. Matchev, Prasanth Shyamsundar, *"Optimal event selection and categorization in high energy physics, Part 1: Signal discovery,"* arXiv:1911.12299 [physics.data-an].
- Parts 2 and 3 to follow.

Table of contents
Quick links
References and citation guide
Copyright

$$D_{\mathrm{Neym}\chi^2} = \sum_{c=1}^{C} \int d\boldsymbol{x}\, \frac{s_c^2(\boldsymbol{x})}{n_c(\boldsymbol{x})}$$

$$D_{\mathrm{Pear}\chi^2} = \sum_{c=1}^{C} \int d\boldsymbol{x}\, \frac{s_c^2(\boldsymbol{x})}{b_c(\boldsymbol{x})}$$

$$D_{\mathrm{KL}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[-s_c(\boldsymbol{x}) - n_c(\boldsymbol{x}) \ln\left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})}\right]\right]$$

$$D_{\mathrm{revKL}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[s_c(\boldsymbol{x}) + b_c(\boldsymbol{x}) \ln\left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})}\right]\right]$$

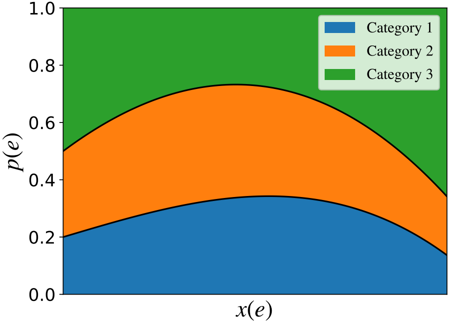
$$D_{\mathrm{J}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[-s_c(\boldsymbol{x}) \ln\left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})}\right]\right]$$

$$D_{\mathrm{B}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[n_c(\boldsymbol{x}) - \frac{s_c(\boldsymbol{x})}{2} - n_c(\boldsymbol{x})\sqrt{1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})}}\right]$$

ThickBrick https://prasanthcakewalk.gitlab.io/thickbrick/

- Input: Training data with $p(\boldsymbol{e})$ and $x(\boldsymbol{e})$ for each event. $p(\boldsymbol{e})$ could be learned using current ML techniques.
- Output: Optimal $\boldsymbol{x}$ dependent thresholds on $p(\boldsymbol{e})$ to maximize any of the following performance measures.

  Note: None of these can be written as sum of event-wise profit functions.



$$D_{\text{Neym}\chi^2} = \sum_{c=1}^{C} \int d\boldsymbol{x} \, \frac{s_c^2(\boldsymbol{x})}{n_c(\boldsymbol{x})}$$

$$D_{\text{Pear}\chi^2} = \sum_{c=1}^{C} \int d\boldsymbol{x} \, \frac{s_c^2(\boldsymbol{x})}{b_c(\boldsymbol{x})}$$

$$D_{\text{KL}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[-s_c(\boldsymbol{x}) - n_c(\boldsymbol{x}) \ln \left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})} \right] \right]$$

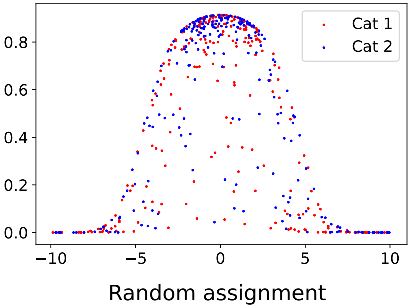
$$D_{\text{revKL}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[s_c(\boldsymbol{x}) + b_c(\boldsymbol{x}) \ln \left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})} \right] \right]$$

$$D_{\text{J}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[-s_c(\boldsymbol{x}) \ln \left[1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})} \right] \right]$$

$$D_{\text{B}} = \sum_{c=1}^{C} \int d\boldsymbol{x} \left[n_c(\boldsymbol{x}) - \frac{s_c(\boldsymbol{x})}{2} - n_c(\boldsymbol{x}) \sqrt{1 - \frac{s_c(\boldsymbol{x})}{n_c(\boldsymbol{x})}} \right]$$

ThickBrick working

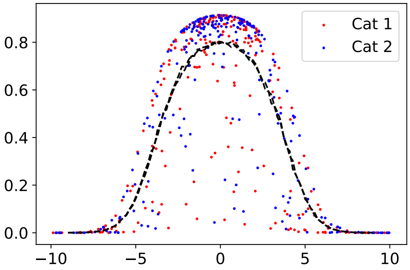
- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.



Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

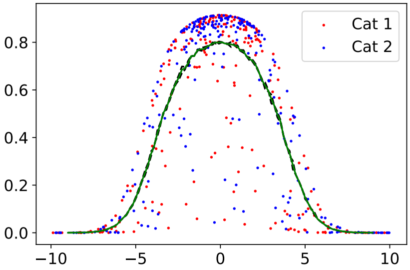


Iter 0: Compute category means

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

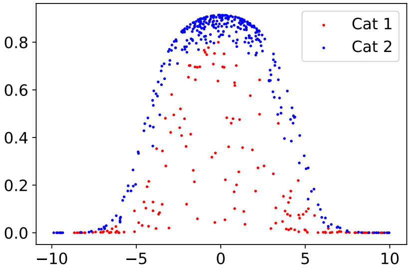


Iter 1: Compute next boundary

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

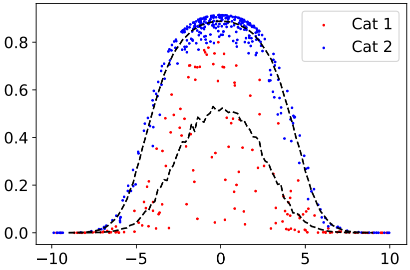


Iter 1: Re-categorize

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

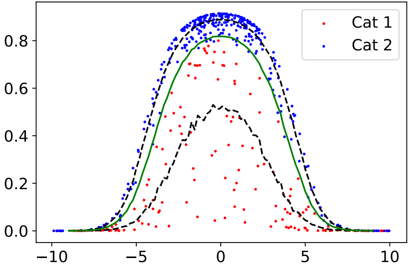


Iter 1: Compute category means

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that "clusters" data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

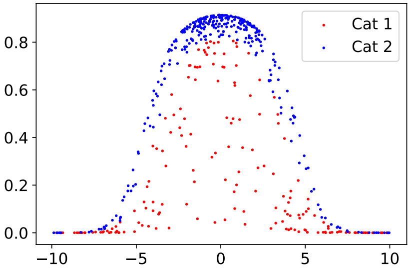


Iter 2: Compute next boundary

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

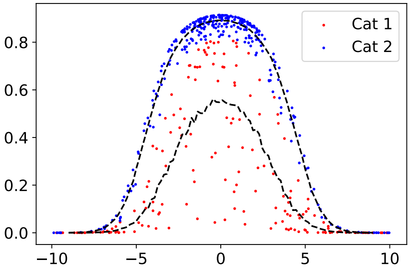


Iter 2: Re-categorize

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that "clusters" data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

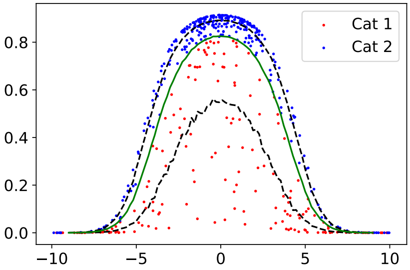


Iter 2: Compute category means

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

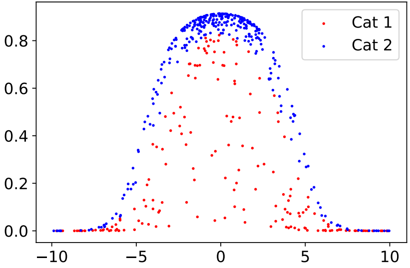


Iter 3: Compute next boundary

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

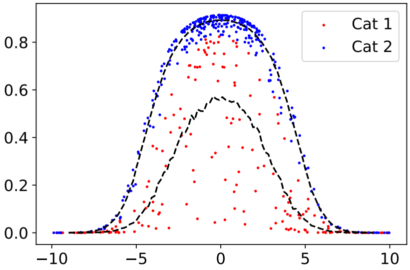


Iter 3: Re-categorize

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that "clusters" data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

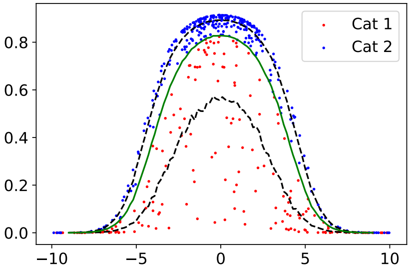


Iter 3: Compute category means

Actual clustering done with 1,000,000 data points, only 500 shown.

ThickBrick working

- Uses a modified k-means clustering algorithm that “clusters” data into different categories.
- Uses a (kernel) regression-based approach to avoid having to work in discrete x-bins.

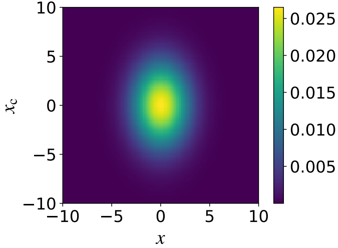


Iter 4: Compute next boundary

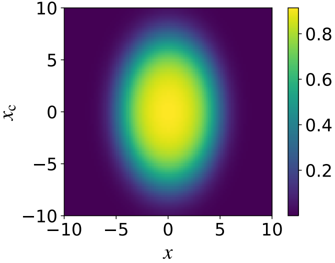
Actual clustering done with 1,000,000 data points, only 500 shown.

Converges too fast to see the clustering in action :/

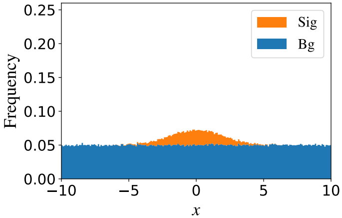
The toy data we've been looking at



signal distribution (flat bg)

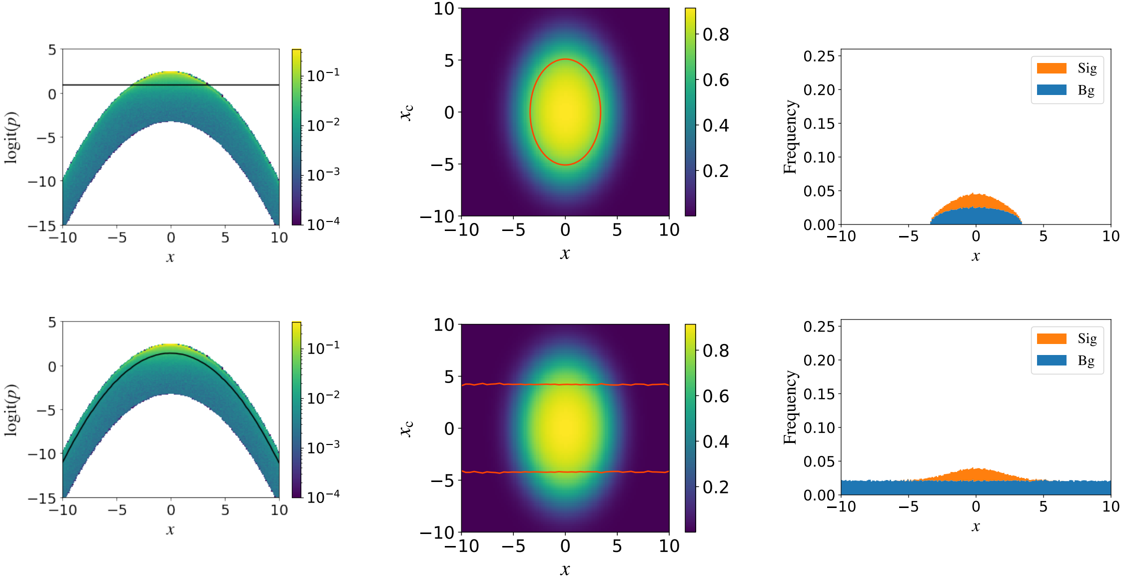


p calculated for a bg:sig = 1:1 sample

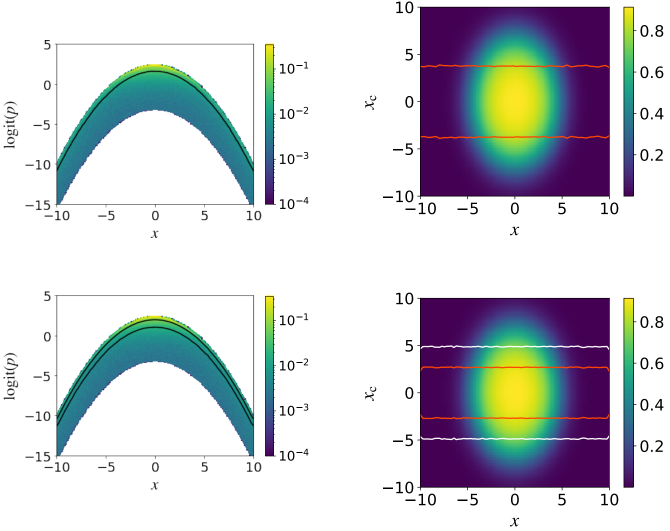


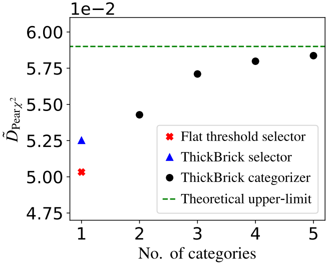
10% signal

Results: Flat cut selector vs ThickBrick selector using $D_{\mathrm{Pear}\chi^2}$



Results: Categorizers using $D_{\text{Pear}\chi^2}$





- Flat cut in x_c wasn't forced—the algorithm never saw x_c.
- Diminishing returns for increasing C... approach the performance of "direct inference from ML output" with just event categorization.

A teaser for

Part 2:
Parameter measurement

Some signal events can be worse than background events...

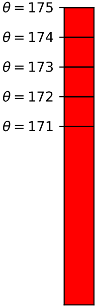
Sensitivity to parameter

In one bin:

- Variation due to parameter value change $\sim \dfrac{dn}{d\theta}$
- Statistical uncertainty in $n \sim \sqrt{n}$
- Measurement uncertainty (inverse) $\sim \dfrac{1}{n}\left(\dfrac{dn}{d\theta}\right)^2$

  $\left(\text{Think } \dfrac{s^2}{n}\right)$

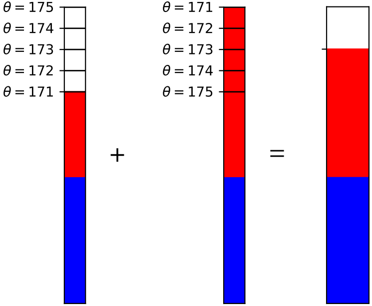
  (Sum or integrate over bins to get Fisher information.)



- Background is insensitive to θ. So background is bad.

Phase-space mixing

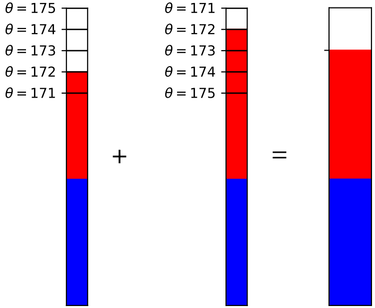
- Signal in red. Bg in blue.
- θ dependence in different parts of phase space **being mixed** could have opposite signs.
- These signal events are worse for sensitivity than background events!
  an extreme example of the misalignment in parameter measurement case.



- Event selection should be based on “score” — sensitivity of an event’s weight to parameter value.
- Estimating score...
  - MadMiner [J. Brehmer, K. Cranmer, I. Espejo, F. Kling, G. Louppe, J. Pavez [1906.01578, 1907.10621]]
  - DCTR [A. Andreassen, B. Nachman [1907.08209]]
  - + our own hat in the ring in part 2
- Event selection using the score to maximize Fisher information — subject of part 2

Phase-space mixing

- Signal in red. Bg in blue.
- θ dependence in different parts of phase space **being mixed** could have opposite signs.
- These signal events are worse for sensitivity than background events!
  an extreme example of the misalignment in parameter measurement case.

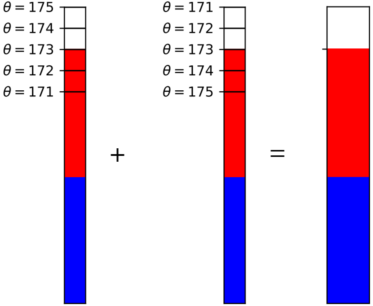


- Event selection should be based on “score” — sensitivity of an event’s weight to parameter value.
- Estimating score...
  - MadMiner [J. Brehmer, K. Cranmer, I. Espejo, F. Kling, G. Louppe, J. Pavez [1906.01578, 1907.10621]]
  - DCTR [A. Andreassen, B. Nachman [1907.08209]]
  - + our own hat in the ring in part 2
- Event selection using the score to maximize Fisher information — subject of part 2

Phase-space mixing

- Signal in red. Bg in blue.
- θ dependence in different parts of phase space **being mixed** could have opposite signs.
- These signal events are worse for sensitivity than background events!

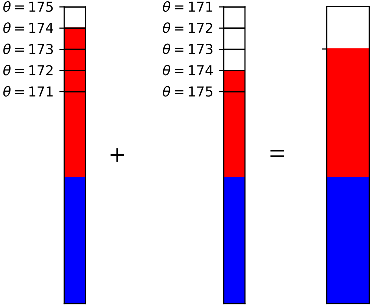
  an extreme example of the misalignment in parameter measurement case.



- Event selection should be based on "score" — sensitivity of an event's weight to parameter value.
- Estimating score...
  - MadMiner [J. Brehmer, K. Cranmer, I. Espejo, F. Kling, G. Louppe, J. Pavez [1906.01578, 1907.10621]]
  - DCTR [A. Andreassen, B. Nachman [1907.08209]]
  - + our own hat in the ring in part 2
- Event selection using the score to maximize Fisher information — subject of part 2

Phase-space mixing

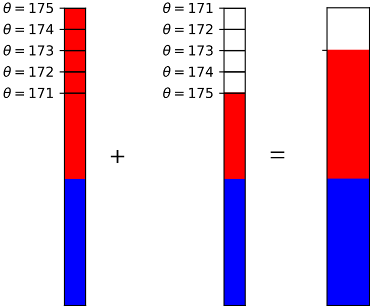
- Signal in red. Bg in blue.
- θ dependence in different parts of phase space **being mixed** could have opposite signs.
- These signal events are worse for sensitivity than background events!
  an extreme example of the misalignment in parameter measurement case.



- Event selection should be based on “score” — sensitivity of an event’s weight to parameter value.
- Estimating score...
  - MadMiner [J. Brehmer, K. Cranmer, I. Espejo, F. Kling, G. Louppe, J. Pavez [1906.01578, 1907.10621]]
  - DCTR [A. Andreassen, B. Nachman [1907.08209]]
  - + our own hat in the ring in part 2
- Event selection using the score to maximize Fisher information — subject of part 2

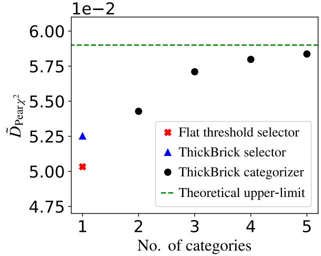
Phase-space mixing

- Signal in red. Bg in blue.
- θ dependence in different parts of phase space **being mixed** could have opposite signs.
- These signal events are worse for sensitivity than background events!
  an extreme example of the misalignment in parameter measurement case.



- Event selection should be based on "score" — sensitivity of an event's weight to parameter value.
- Estimating score...
  - MadMiner [J. Brehmer, K. Cranmer, I. Espejo, F. Kling, G. Louppe, J. Pavez [1906.01578, 1907.10621]]
  - DCTR [A. Andreassen, B. Nachman [1907.08209]]
  - + our own hat in the ring in part 2
- Event selection using the score to maximize Fisher information — subject of part 2

Summary and Upcoming



Summary:

- We have optimized event selection and categorization for signal discovery (statistical significance, exactly specified signal)

Upcoming:

- Part 2: Optimal for parameter measurement
  (pessimistic: By Apr 2020)
- Part 3: "Optimal" over a range of signal parameter values
  (pessimistic: By Jul 2020)
  - Advantage of "event selection followed by event variable based search": Sensitivity over a range of signal param, say mass of new particle.
- Part 3: "Optimal" incorporating systematic uncertainties!!!
  (pessimistic: By Jul 2020)
  - Using sensitivity of events to nuisance parameter value

Bonus 1: Decorrelation

The decorrelation properties can have applications in

- Mass decorrelation in jet taggers
- Decorrelating classifier trained on "naturally mixed samples" [LLP, CWoLa] from, say, differing underlying kinematics.
- Can do things other than s^2/b, like $-\sqrt{sb}$.

$$\sum_{c=1}^{C} \int d\boldsymbol{x} \; \frac{s_c^2(\boldsymbol{x})}{b_c(\boldsymbol{x})}$$

$$-\sum_{c=1}^{C} \int d\boldsymbol{x} \; \sqrt{s_c(\boldsymbol{x}) b_c(\boldsymbol{x})}$$

Bonus 2: A broader ML implication

- Training was done in two phases
  1. Learn $p(\boldsymbol{e})$ using ML
  2. Get optimal thresholds on $p(\boldsymbol{e})$ iteratively.
- But the two steps can be combined.
- Original idea did event selection directly based on $\boldsymbol{e}$ (iteratively or stochastically) — temporarily shelved in favor of the two phase approach for easy adoptability.

Takeaway:

- It is possible to train neural networks event-by-event to optimize cost functions that cannot be written as a sum of an event-wise loss function.
- Clues lie in the construction of our method in part 1, for those interested.
  (Long, but an easy read)

Jump to
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Bonus 2: A broader ML implication

- Training was done in two phases
  1. Learn $p(\boldsymbol{e})$ using ML
  2. Get optimal thresholds on $p(\boldsymbol{e})$ iteratively.
- But the two steps can be combined.
- Original idea did event selection directly based on $\boldsymbol{e}$ (iteratively or stochastically) — temporarily shelved in favor of the two phase approach for easy adoptability.

Takeaway:

- It is possible to train neural networks event-by-event to optimize cost functions that cannot be written as a sum of an event-wise loss function.
- Clues lie in the construction of our method in part 1, for those interested.
  (Long, but an easy read)

Questions?