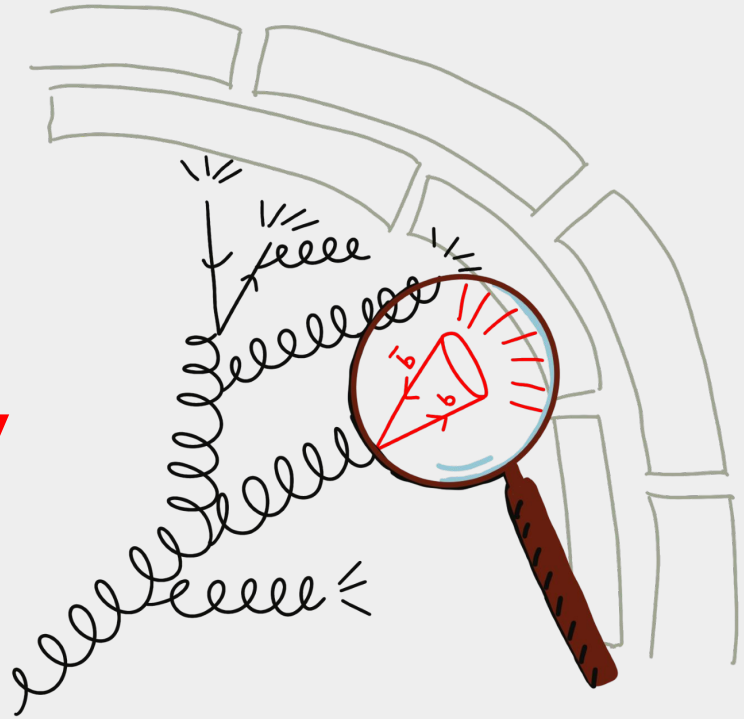
[**RG**, Mastandrea, Nachman, Thaler; 2502.14036]
[**RG**, Mastandrea, Nachman, Thaler; WIP]

Likelihood-based Reweighting for Improved sensitivity in Anomaly Detection



Rikab Gambhir

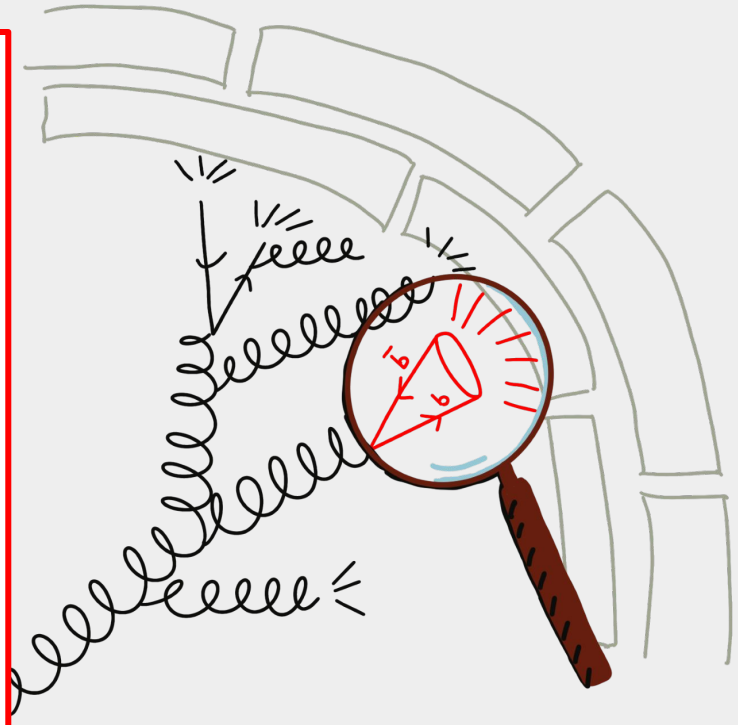
In Collaboration with: Radha Mastandrea, Ben Nachman, and Jesse Thaler



[**RG**, Mastandrea, Nachman, Thaler; 2502.14036]
[**RG**, Mastandrea, Nachman, Thaler; WIP]

Alternate Talk Title:

The Least Interesting Thing to do with a Classifier is Classify.



Rikab Gambhir

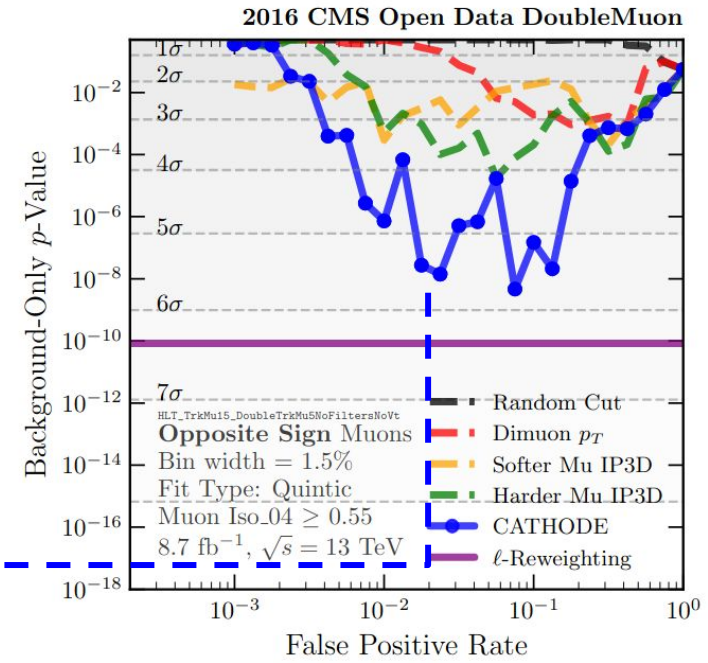
In Collaboration with: Radha Mastandrea, Ben Nachman, and Jesse Thaler



[RG, Mastandrea, Nachman, Thaler; 2502.14036]
[1][Hallin et. al.; 2109.00546]

(By Radha!)

Previous Talk: *Isolating UnIsolated Upsilons*



5.7σ Excess over a smooth background!

We saw that we can use **CATHODE**[1] to find *real signals* in *real data*!

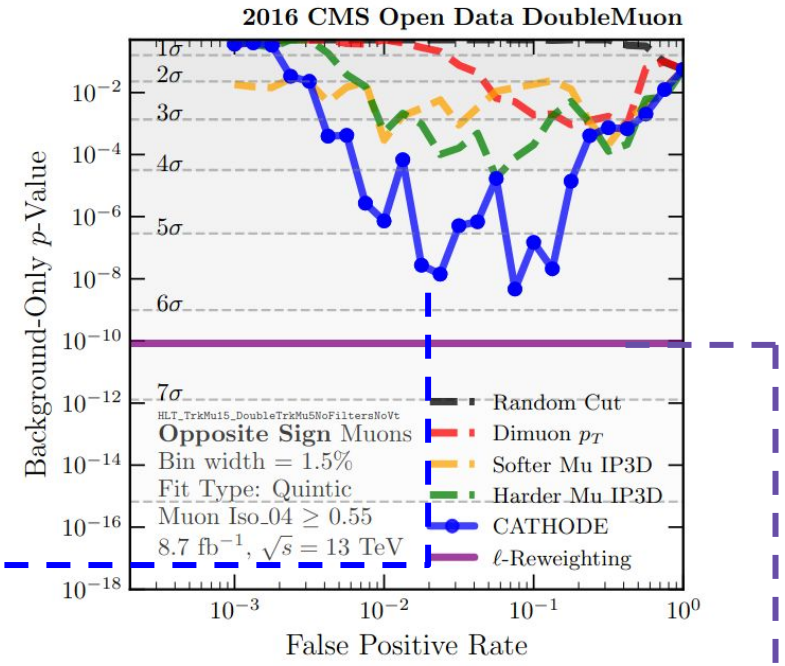
We found something > 5σ! We can declare victory.

← Try our curated Upsilon benchmark!

[**RG**, Mastandrea, Nachman, Thaler; 2502.14036]
[1][Hallin et. al.; 2109.00546]

(By Radha!)

Previous Talk: *Isolating UnIsolated Upsilons*



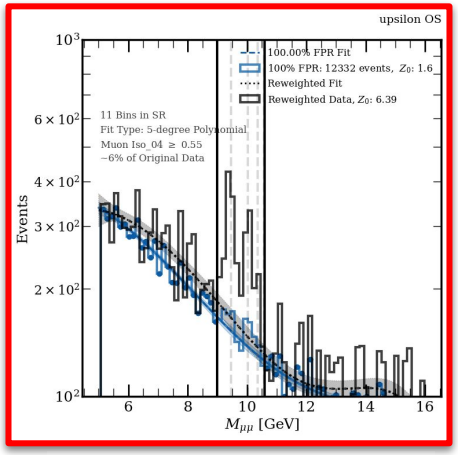
5.7σ Excess over a smooth background!

We saw that we can use **CATHODE**[1] to find *real signals* in *real data*!

We found something > 5σ! We can declare victory.

This Talk: We aren't finished winning! Let's improve sensitivity *further*, with *no additional work*, with **likelihood-based reweighting**! → 6.4σ ←

Likelihood-based Reweighting



1. **What did we do?**

   Explaining the weighted method

2. **Why did we do it?**

   Near-Optimality

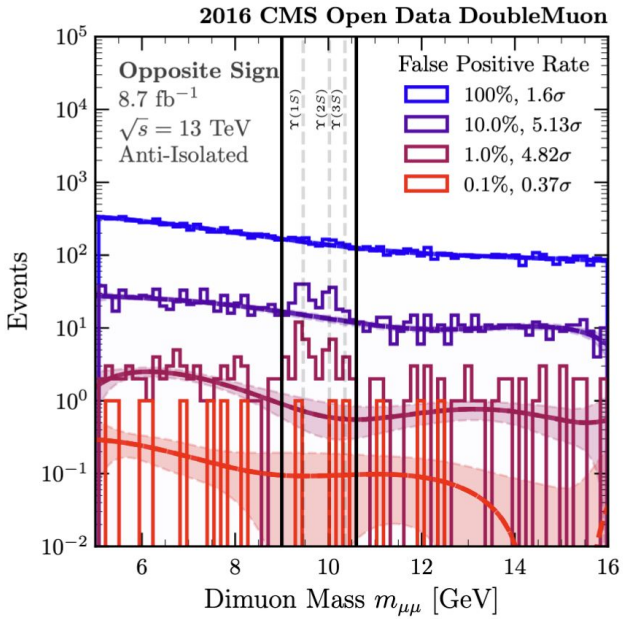
$$Z = \frac{\int dx\, S(x) w(x)}{\sqrt{\int dx\, B(x) w^2(x)}}$$

3. **Why is this OK to do?**

   Robustness and sanity checks

Cut-and-Count Anomaly Detection

[See Radha's talk for more details!]



$$\mathcal{L}(S;N) = N\log(B+S) - (B+S) - \log\Gamma(1+N)$$

$$B = \int_{\text{Signal Region}} [f(m;\theta)]\, dm$$

$f(m;\theta) =$ Poisson Likelihood fit to SB

Outline of CWoLa-Like Methods

1. Define data samples in a signal region and obtain background samples to compare against.

   CWoLa: Bkg samples = sideband
   CATHODE: Bkg samples = NF interpolation
   c.f. CURTAINS, SALAD,, …

2. Train a classifier between the data and background.

   Fact: A data-to-background classifier is monotonically related to a signal-to-background classifier!

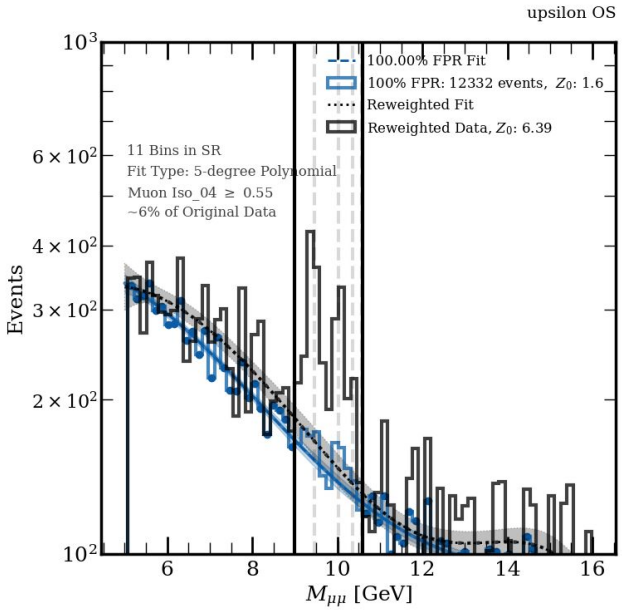
3. Cut on the classifier to enrich the signal fraction.

4. Perform an ordinary mass bump-hunt on the surviving samples, compute test statistics

   Our hypothesis test: Fit (with nuisance parameters) to sidebands to interpolate the bkg. Treat the SR as a single Poisson bin with the mean determined by the fit. Then check for $S = 0$ vs $S > 0$ in that bin.

Weight ~~Cut~~-and-Count Anomaly Detection

[The New Thing!]



$\mathcal{L}(S; N, w_i) =$ Scaled Poisson Likelihood with $\lambda = S + B$, moment $= \langle w^2 \rangle / \langle w \rangle$

$$B = \int_{\text{Signal Region}} [f(m;\theta)]\, dm$$

$f(m;\theta) =$ Scaled Poisson Likelihood fit to SB

Outline of CWoLa-Like Methods

1. Define data samples in a signal region and obtain background samples to compare against.
2. Train a classifier between the data and background.
3. ~~Cut on the classifier to enrich the signal fraction~~
4. Perform an ordinary mass bump-hunt on the surviving samples, compute test statistics

Weight each event *x* using some weight function *w(x)*.

The *best* weight function is the signal-to-background likelihood ratio, but any weight function is valid!

Weight ~~Cut~~-and-Count Anomaly Detection (Details)

The effect of weighting is to replace *Poissons* with *Weighted (Compound[1]) Poissons*

$$N_{\text{cut}} = \sum_i^N \Theta(f(x_i) - f_{\text{cut}}) \longrightarrow N_w = \sum_i^N w(x_i)$$

Observable: Bin Count *N*

Observable: Weighted Bin Count, depends on *N and* $\langle w \rangle$

$$\text{Pois}(\text{N}; \lambda) \longrightarrow \text{CPD}(N_w; \lambda, p(w))$$

Poissonian Bin Counts

Compound-Poissonian Bin Counts
Depends on the weight distribution – but not strongly!

Approx

$$\text{SPD}(N_w; \lambda, \langle w^2 \rangle)$$

Tractable approximation to CPD

Net effect is to alter the first and second moments of the Poisson: Nice in the asymptotic limit!
Weighted Bump Hunts!

Weight each event *x* using some weight function *w(x)*.

The *best* weight function is the signal-to-background likelihood ratio, but any weight function is valid!

Likelihood-based Reweighting

1. What did we do?

Explaining the weighted method

2. **Why did we do it?**

Near-Optimality

$$Z = \frac{\int dx\, S(x)w(x)}{\sqrt{\int dx\, B(x)w^2(x)}}$$

3. Why is this OK to do?

Robustness and sanity checks

*Anything reasonable — no weight distributions with infinite variance!
**Nice property: The SPD is invariant under weight rescaling. So, no free power just by doubling weights!

[Neyman, Pearson; On the problem of the most efficient tests of statistical hypotheses]
[**RG**, Mastandrea, Nachman, Thaler; 2502.14036]

The *Optimal* Weights

What weight function should we choose? Anything is valid![*]

If you use your classifier to classify, that corresponds to choosing weights of 0 or 1. But:

1. You have to choose a classification threshold (a *working point*).
2. Information about *how* signal or background like an event is gets lost.

Classifier scores are *more than just scores* — by Neyman-Pearson, they're **Likelihood Ratios**!

Claim: Choosing the weight function to be the signal-to-background per-event likelihood ratio provides (near)-optimal sensitivity!

Optimal in the single-bin counting experiment limit with known likelihood ratio

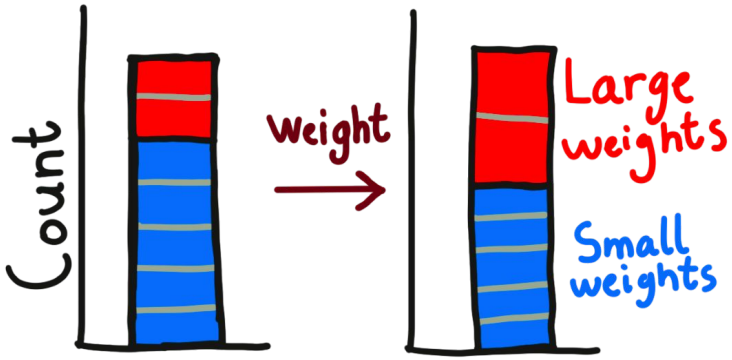
This is essentially a *free* increase in sensitivity just by using the fact that classifiers learn the likelihood ratio, plus we don't have to pick a cut working point!.
The worst thing to do with a classifier is classify!

[Neyman, Pearson; On the problem of the most efficient tests of statistical hypotheses]
[RG, Mastandrea, Nachman, Thaler; 2502.14036]

[Freytsis, Ovanesyan, Thaler; 0909.2862]

Sketch of Proof

For simplicity, assume we are doing a single-bin counting experiment in the asymptotic limit with known signal.* Let's allow every event to have a weight w**.



The total number of events is now:

$$S_{\text{eff}} = \int dx\, S(x)w(x), \quad B_{\text{eff}} = \int dx\, B(x)w(x)$$

The "significance" is:

$$Z = \frac{\int dx\, S(x)w(x)}{\sqrt{\int dx\, B(x)w^2(x)}} \quad \text{(Generalizes } Z = \frac{S}{\sqrt{B}}\text{)}$$

The Likelihood Ratio!

We can then calculate what choice of weights optimizes the significance:

$$w(x) = \frac{S(x)}{B(x)} = \frac{N_{\text{Sig}}}{N_{\text{Bkg}}} \frac{p_{\text{Sig}}(x)}{p_{\text{Bkg}}(x)}$$

Unimportant Constants

*In the full analysis, we do a complete Poisson likelihood treatment with nuisance parameters due to fits. When these are included, these weights may not be strictly optimal, but they are still better than nothing!
**Ordinary cutting can be seen as assigning a weight of 0.

The worst thing to do with a classifier is classify!

Last Ingredient: Estimating the Likelihood

We are *already* training BDT's anyways to perform the CWoLa-style cuts.
Neyman-Pearson tells us we learn the likelihood ratio out of this!

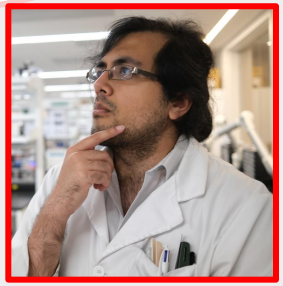
Classifier score — Naive signal estimate, obtained by first-pass bump-hunt

$$\ell(x) \equiv \frac{z(x) - (1-\mu)\,(1-z(x))}{\mu\,(1-z(x))} = \frac{p_{\mathrm{Sig}}(x)}{p_{\mathrm{Bkg}}(x)}$$

Technical detail: Both the classifier score z and the signal fraction μ are imperfect estimators, so the weights might go negative! But we can just cut those out — we were cutting things in the original analysis anyways! Reduces optimality, but not correctness!

[Aside]:

"Why bother with all this weighting stuff? If I have the likelihood ratio per-event, isn't that *already* the most powerful hypothesis test?"



[Me over the course of my first AD project, confused why we do cuts and not just the full likelihood]

It is true that the *full* information we have access to is the *set* of all per-event likelihoods

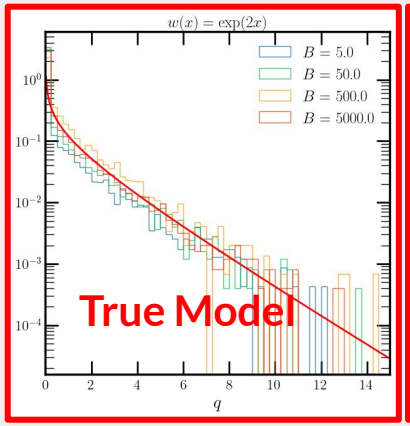
$$\mathcal{L}(\lambda; N, x_i) = \log \mathrm{Pois}(N; \lambda) + \sum_i^N \log(\ell(x))$$

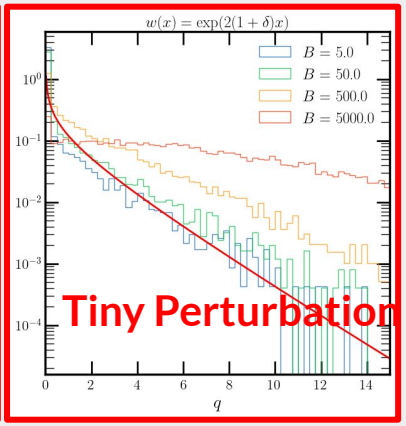
And that this leads to a powerful test statistic.

But it isn't the most ***stable*** — if ℓ is even slightly off, the entire distribution of the test statistic under the null hypothesis is ruined. Extremely hard to calibrate!

But the sum of weights is more robust!

Toy model: Gaussian Likelihoods





Test Statistic Distribution

Likelihood-based Reweighting

1. What did we do?

Explaining the weighted method

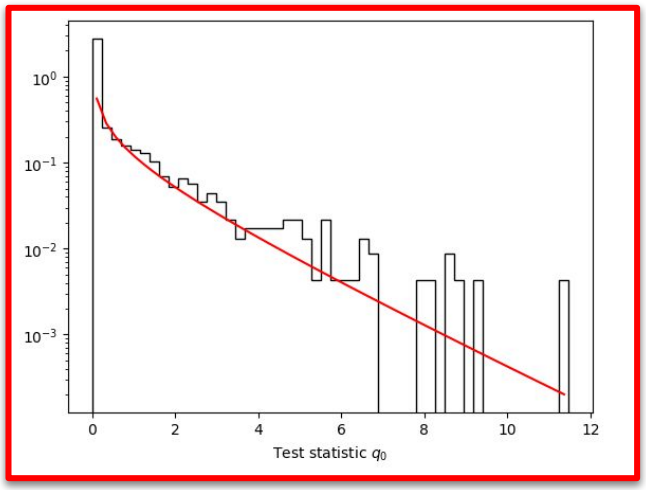
2. Why did we do it?

Near-Optimality

$$Z = \frac{\int dx\, S(x)w(x)}{\sqrt{\int dx\, B(x)w^2(x)}}$$

3. Why is this OK to do?

Robustness and sanity checks



[Cowan, Cranmer, Gross, Vitells; 1007.1727]

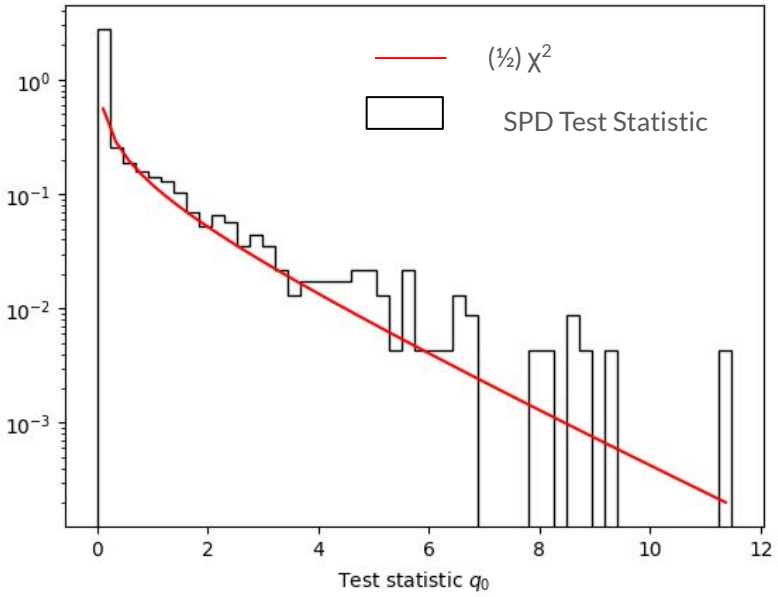
Distribution of the Test Statistic

It is already known that under the null hypothesis, in the asymptotic limit, the distribution of the test statistic of a Poissonian counting process follows a (half)-chi^2 distribution [Wilkes & Wald]

Weights are not expected to change this! They only change the effective Gaussian moments, but this is OK!**

Pictured: Distribution of the test statistic under the null hypothesis (Same-Sign data), with pseudoexperiments generated via NF

Upshot: it works!

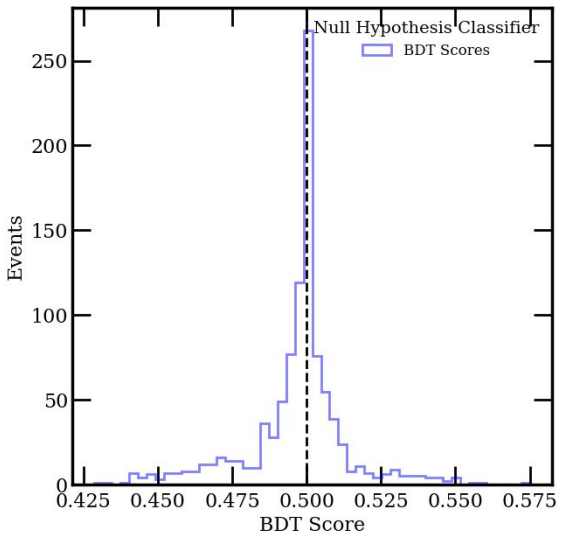


*Up to potential small third-moment effects
*What about small *N*? Ask me offline!

Objection 1: BDT's are Fallible

A real-life BDT does *not* learn the likelihood ratio. Even a BDT between two identical datasets will not learn a likelihood ratio of 1, it will be 1 ± noise.

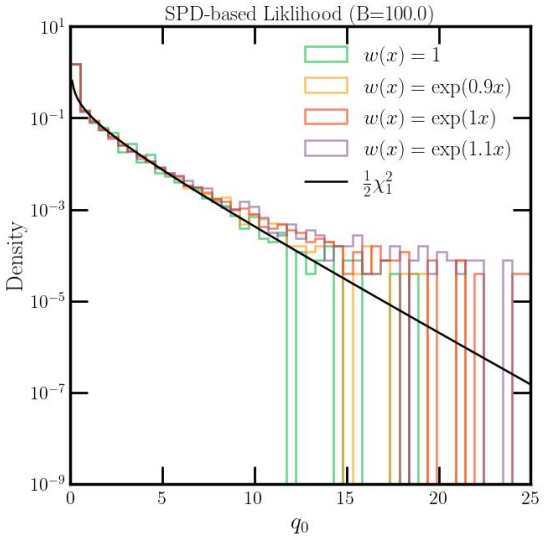
But this is OK! Any weights work, and produce the same test statistic distribution under the null. You may not have the *most powerful* weights anymore, but you will not get an incorrect result.



Example: Gaussian Signal on a Gaussian background

Constructed such that the true likelihood ratio is exp(x)

But mismodeling does not significantly change the distribution of the test statistic![*]



[*]At low significance values
[**]Extremely high variance can begin to ruin things

[c.f. Prasanth's Talk!]

Objection 2: Looking at Data?

We look at the data to determine the weights via training. Technically, this training procedure should be part of the full hypothesis likelihood.

However, this is not an issue with our method, but rather all CWoLa-style AD. As in CATHODE, we assume that the training "factorizes" and we do not have a strong signal-fraction dependence.

The only place the signal fraction enters is through (a) the distribution of weights and (b) the definition of the weight function, both of which have small dependence

Distribution of Weights

For μ << 1:

$$\langle w \rangle = e^{\frac{1}{2}\lambda^2 - \frac{1}{2}\lambda} + 2\mu e^{\frac{1}{2}\lambda^2} \sinh\left(\frac{1}{2}\lambda\right)$$

$$\langle w^2 \rangle = e^{2\lambda^2 - \lambda} + 2\mu e^{2\lambda^2} \sinh(\lambda)$$

Small!

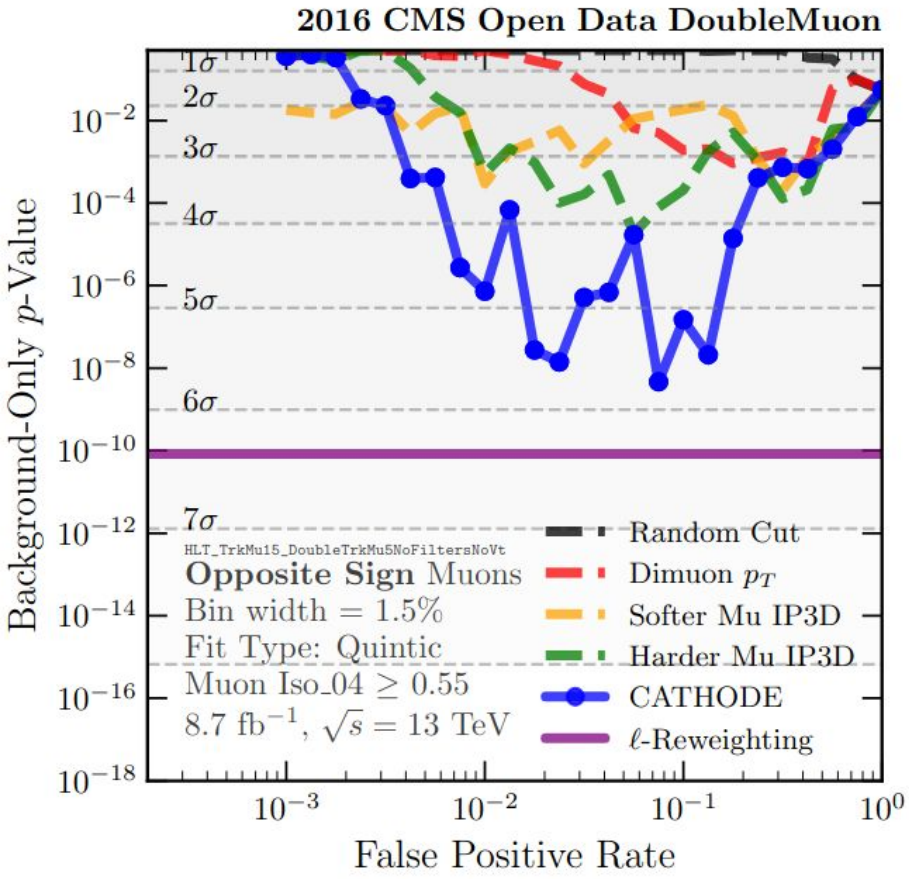
(And similar for all higher moments)

Two Gaussians model

Definition of Weights

Toy model: Fit a basic "logistic classifier" L = exp(λx) on data. For a signal fraction of exactly 0, will learn $\lambda = 0$. For any positive signal fraction, will learn $\lambda = 1$ in the infinite-statistics limit.
But we already know (previous slide) that changing λ is robust!

Conclusion



- AD can be used to find *real*, *nontrivial signals* in *real data*, like hard-to-find upsilons.
- Likelihood reweighting can improve sensitivity even further.
- Classifiers can do more than classify!

Try our curated Upsilon open data benchmark!

Appendices

Same-Sign Validation