

# Likelihood ratio tests constructed with discriminative classifiers and calibrated with generative models

**Kyle Cranmer**

KYLE.CRANMER@NYU.EDU

*Center for Cosmology and Particle Physics*

*Center for Data Science*

*New York University*

*New York, NY 10003, USA*

**Editor:**

## Abstract

words

## 1. Introduction

In many areas of science, likelihood ratio tests are established tools for statistical inference. Directly constructing the likelihood ratio for high-dimensional observations is often not possible or is computationally impractical. Here we demonstrate how discriminative classifiers can be used to construct equivalent likelihood ratio tests when a generative model for the data is available for calibration. We use the following notation

- $x$ : a vector of features
- $D$ : a dataset of  $D = \{x_1, \dots, x_n\}$ , where  $x_e$  are assumed to be i.i.d.
- $\theta$ : parameters of a statistical model
- $f(x|\theta)$ : probability density (statistical model) for  $x$
- $s(x; \theta_0, \theta_1)$ : real-valued discriminative classification score, parametrized by  $\theta_0$  and  $\theta_1$
- $g(s|\theta)$ : The probability density for  $s(x; \theta_0, \theta_1)$  implied by  $f(x|\theta_0, \theta_1)$

We will assume the  $x_e$  are i.i.d., so that  $f(D|\theta) = \prod_{e=1}^n f(x_e|\theta)$ .

In the setting where one is interested in simple hypothesis testing between a null  $\theta = \theta_0$  against an alternate  $\theta = \theta_1$ , the Neyman-Pearson lemma states that the likelihood ratio

$$T(D) = \prod_{e=1}^n \frac{f(x_e|\theta_0)}{f(x_e|\theta_1)} \quad (1)$$

15 is the most powerful test statistic. In order to evaluate  $T(D)$ , one must be able to evaluate  
 16 the probability density  $f(x|\theta)$  at any value  $x$ . However, it is increasingly common in science  
 17 that one has a complex simulation that can act as generative model for  $f(x|\theta)$ , but one  
 18 cannot evaluate the density directly. For instance, this is the case high energy physics  
 19 where the simulation of particle detectors can only be done in the ‘forward mode’.

Our main result is that one can form an equivalent test based on

$$T'(D) = \prod_{e=1}^n \frac{g(s_e|\theta_0)}{g(s_e|\theta_1)} \quad (2)$$

if

$$s_e = s(x_e; \theta_0, \theta_1) = \frac{f(x_e|\theta_0)}{f(x_e|\theta_1)} \quad (3)$$

20 or some monotonic function of that ratio. This will be proven below. This allows us to  
 21 recast the original likelihood ratio test into an alternate form in which a discriminative  
 22 classifier is used to learn  $s(x; \theta_0, \theta_1)$ . The discriminative classifier can be trained with data  
 23  $(x, y = 0)$  generated from  $f(x|\theta_0)$  and  $(x, y = 1)$  generated from  $f(x|\theta_1)$ . In Section 4 we  
 24 extend this result to generalized likelihood ratio tests, where it will be useful to have the  
 25 discriminative classifier explicitly parametrized in terms of  $(\theta_0, \theta_1)$ .

26 While the original goal for frequentist hypothesis testing is to make a decision to accept  
 27 or reject the null hypothesis based on the entire dataset  $D$ , the machine learning problem  
 28 is an event-by-event classification problem. This follows from the fact that we assume the  
 29  $x_e$  to be i.i.d.

## 30 2. Comments on classification and frequentist hypothesis tests

31 Significant literature exists around generative classifiers and discriminative classifiers. Typ-  
 32 ically, generative classifiers learn a model for the joint probability  $p(x, y)$ , of the inputs  $x$   
 33 and the classification label  $y$ , and predict  $p(y|x)$  via Bayes rule. In contrast, discriminative  
 34 classifiers model the posterior  $p(y|x)$  directly. For classification tasks, one then thresholds  
 35 on  $p(y|x)$ . In both cases this description in terms of a posterior requires a prior distribu-  
 36 tion for  $p(y)$ , which is either modeled explicitly or learned from the training data. This  
 37 formulation of classification is somewhat confusing in the setting of the current work.

38 First of all, the generative model  $f(x|\theta)$  here is a generative statistical model, not a  
 39 generative classifier. We think of the  $f(x|\theta)$  along the lines of a traditional scientific theory,  
 40 able to make predictions about  $x$  and being motivated by domain-specific considerations.  
 41 For example, in the context of high energy particle physics  $f(x|\theta)$  is based on quantum field  
 42 theory and a detailed simulation of the particle detector and data processing algorithms  
 43 that transform raw sensor data into the feature vector  $x$ . The point here is to make  
 44 the distinction of  $f(x|\theta)$  in this context from a generic generative classifier like normal  
 45 discriminate analysis or naive Bayes.

Secondly, we are not interested in thresholded classification on individual events  $x_e$ , but are interested in a thresholded hypothesis test on an entire data set based on the likelihood ratio test statistic  $T(D)$ . Additionally, we know that both discriminative and generative classifier scores are often poorly calibrated. We wish to have well calibrated statements, but not about  $p(y|x)$  but p-values associated to  $P(T(D) > k|\theta)$ .

Lastly, in the setting of frequentist hypothesis we do not have a prior  $\pi(\theta)$ . While we can use the generative models to produce training data  $(x, y = 0)$  generated from  $f(x|\theta_0)$  and  $(x, y = 1)$  generated from  $f(x|\theta_1)$ , the relative mix  $p(y)$  is arbitrary. When  $p(y = 0) = p(y = 1) = 1/2$ , then

$$p(y = 1|x) = \frac{p(x|y = 1)}{p(x|y = 0) + p(x|y = 1)} = \frac{f(x|\theta_1)}{f(x|\theta_0) + f(x|\theta_1)}, \quad (4)$$

which is monotone with the desired likelihood ratio  $f(x|\theta_1)/f(x|\theta_0)$  and free from the complications of the prior  $\pi(\theta)$  that is undefined in the frequentist context.

Since the prior  $p(y)$  has no clear meaning for the test at hand and because the classifier score  $p(y|x)$  may not be well calibrated, we choose to denote the classifier score  $s(x)$  and simply think of it as a deterministic dimensionality reduction map  $s : X \rightarrow \mathbb{R}$ .

### 3. Dimensionality reduction

The target hypothesis test is based on

$$\ln T = \sum_{e=1}^n \log \underbrace{\left[ \frac{f(x_e|\theta_0)}{f(x_e|\theta_1)} \right]}_{q(x_e)}. \quad (5)$$

Here we see that the optimal  $T$  for the experiment is composed of a sum over events of a linear function of the per-event function  $q(x)$ . A monotonic, but non-linear function of  $q(x)$  would not lead to an equivalent hypothesis test.

The important part of the per-event function  $q(x)$  is that it defines iso-contours in the feature space  $x$ . As we will show, our goal is to learn a monotonic function of  $f(x|\theta_0)/f(x|\theta_1)$  that shares the same iso-contours. Then the remaining challenge is to find the appropriate rescaling that gives back linear function of  $q(x)$ . Our claim is that the generative model  $f(x|\theta)$  can be used to calibrate  $g(s|\theta)$  and that

$$\ln T' = \sum_{e=1}^n \log \underbrace{\left[ \frac{g(s_e|\theta_0)}{g(s_e|\theta_1)} \right]}_{q(s_e)}, \quad (6)$$

leads to an equivalent test. In particular, we need to show the density

$$f(q_x|\theta) = \int dx \delta(q_x - q_x(x)) f(x|\theta) / |\hat{n} \cdot \nabla q_x| \quad (7)$$

is the same as

$$f(q_s|\theta) = \int dx \delta(q_s - q_s(s(x))) f(x|\theta) / |\hat{n} \cdot \nabla q_s|. \quad (8)$$

60 It is sufficient to show that  $q(x_e) = q(s(x_e)) \forall x \in \Omega_c$ .

For notational simplicity, let  $f_0(x) = f(x|\theta_0)$ ,  $f_1(x) = f(x|\theta_1)$ , and  $s(x) = s(x; \theta_1, \theta_0)$ . The distribution of  $x$  totally determines the distribution of  $s$  via the change of variables  $x \rightarrow s$ . In the application at hand, the function  $s$  maps a high-dimensional feature vector  $x$  to  $\mathbb{R}^+$ . Let  $\Omega_c$  be the level set  $\{x \mid s(x) = c\}$  and  $\hat{n} = \nabla s(x)/|\nabla s(x)|$  be the orthonormal vector to  $\Omega_c$  at the point  $x$ . The induced density  $g_1(c)$  is given by

$$g_1(c) = \int dx \delta(c - s(x)) f_1(x) = \int d\Omega_c f_1(x) / |\hat{n} \cdot \nabla s| \quad (9)$$

61 and a similar equation for  $g_0(c)$ .

62

**Theorem 1:** We have the following equalities

$$\frac{g_1(c)}{g_0(c)} = s(x) = \frac{f_1(x)}{f_0(x)} \quad \forall x \in \Omega_c. \quad (10)$$

**Proof** We can factor out of the integral  $s(x) = f_1(x)/f_0(x)$  since it is constant over  $\Omega_c$ . Thus

$$g_1(c) = \int dx \delta(c - s(x)) f_1(x) = \int d\Omega_c f_1(x) / |\hat{n} \cdot \nabla s| = s(x) \int d\Omega_c f_0(x) / |\hat{n} \cdot \nabla s|, \quad (11)$$

and the integrals cancel in the likelihood ratio

$$\frac{g_1(c)}{g_0(c)} = \frac{s(x) \int d\Omega_c f_0(x) / |\hat{n} \cdot \nabla s|}{\int d\Omega_c f_0(x) / |\hat{n} \cdot \nabla s|} = s(x) = \frac{f_1(x)}{f_0(x)} \quad \forall x \in \Omega_c. \quad (12)$$

In the case of simple hypothesis testing,  $\theta_0$  and  $\theta_1$  are specified and there is a unique map  $s(x) = s(x_e; \theta_0, \theta_1)$ . In that case, the equivalent likelihood ratio test can be performed by first transforming the data to  $D_s = \{s_1, \dots, s_e\}$ , constructing the likelihoods

$$g(D_s | \theta) = \prod_{e=1}^n g(s_e | \theta) \quad (13)$$

63 for  $\theta = \{\theta_0, \theta_1\}$ , and constructing the likelihood ratio based on  $g(D_s|\theta_0)/g(D_s|\theta_1)$ .

## 4. Composite hypotheses and the generalized likelihood ratio

In the case of composite hypotheses  $\theta \in \Theta_0$  against an alternative  $\theta \in \Theta_0^C$ , the generalized likelihood ratio<sup>1</sup> test is commonly used

$$\lambda(x) = \frac{\sup_{\theta \in \Theta_0} f(D|\theta)}{\sup_{\theta \in \Theta} f(D|\theta)} . \quad (14)$$

This generalized likelihood ratio can be used both for hypothesis tests in the presence of nuisance parameters or to create confidence intervals with or without nuisance parameters. Often, the parameter vector is broken into two components  $\theta = (\mu, \nu)$ , where the  $\mu$  components are considered parameters of interest while the  $\nu$  components are considered nuisance parameters. In that case  $\Theta_0$  corresponds to all values of  $\nu$  with  $\mu$  fixed.

Denote the maximum likelihood estimator

$$\hat{\theta} = \arg \max_{\theta} f(D|\theta) \quad (15)$$

and the conditional maximum likelihood estimator

$$\hat{\hat{\theta}} = \arg \max_{\theta \in \Theta_0} f(D|\theta) . \quad (16)$$

It is not obvious that if we are working with the distributions  $g(s|\theta)$  (for some particular  $s(x; \theta_0, \theta_1)$  comparison) that we can find the same estimators. Fortunately, there is a construction based on  $g(s|\theta)$  that works. The maximum likelihood estimate is the same as the value that maximizes the ratio with respect to  $f(D|\theta_1)$  for some fixed value of  $\theta_1$ . This allows us to use Theorem 1 to find

$$\hat{\theta} = \arg \max_{\theta} \frac{f(D|\theta)}{f(D|\theta_1)} = \arg \max_{\theta} \sum \ln \frac{f(x_e|\theta)}{f(x_e|\theta_1)} = \arg \max_{\theta} \sum \ln \frac{g(s(x_e; \theta, \theta_1)|\theta)}{g(s(x_e; \theta, \theta_1)|\theta_1)} . \quad (17)$$

It is important that we include the denominator  $g(s(x_e; \theta, \theta_1)|\theta_1)$  because this cancels Jacobian factors that change as we vary  $\theta$ .

## 5. Learning the correct mapping and its distribution

Thus far we have shown that likelihood ratio tests based on  $f(x|\theta_0)/f(x|\theta_1)$  with high dimensional features  $x$  can be reproduced via hypothesis tests based on the univariate densities  $g(s|\theta)$  for the very special dimensionality reduction map  $s(x|\theta_0, \theta_1)$ . The motivation for this is that often it is not possible to evaluate the density  $f(x|\theta)$  at a given point  $x$ . This approach is not useful if it is not possible to approximate  $s(x|\theta_0, \theta_1)$  and  $g(s|\theta)$  without evaluating the density  $f(x|\theta)$ . In order for this approach to be useful, we need to be able to approximate both based on samples  $\{(x, \theta)\}$  drawn from the generative model  $f(x|\theta)$ .

---

1. Also known as the profile likelihood ratio.

80 Denote the approximate dimensionality reduction map  $\hat{s}(x; \theta_0, \theta_1)$  and its distribution  
 81  $\hat{g}(\hat{s}|\theta)$ . In general we will be interested in the machine learning problem that approximates  
 82 these distributions based on samples  $\{x_i\}$  drawn from the generative model  $f(x|\theta)$ . In  
 83 particular, is there a loss function such that the function  $\hat{s}(x)$  that minimizes the expected  
 84 loss leads to a function that is one-to-one with  $f(x|\theta_0)/f(x|\theta_1)$ ?

## 85 5.1 The standard discriminative classification setting

For fixed  $\theta_0$  and  $\theta_1$  we can generate large samples from each model and train a classifier. To be concrete, let's use  $f(x|\theta_0)$  to generate training data ( $x_i, y_i = 0$ ) and  $f(x|\theta_1)$  to generate training data ( $x_i, y_i = 1$ ). If we use the squared-loss function, then the expected loss is

$$\mathbb{E}[L] = \int dx f(x|\theta_0)(\hat{s}(x))^2 + \int dx f(x|\theta_1)(1 - \hat{s}(x))^2 . \quad (18)$$

86 The function  $\hat{s}(x) = f(x|\theta_1)/(f(x|\theta_0) + f(x|\theta_1))$  minimizes this expected loss.

Proof (Sketch): Consider a variation about  $\hat{s}(x) = f(x|\theta_1)/(f(x|\theta_0) + f(x|\theta_1))$  given by  $\hat{s}'(x) = \hat{s}(x) + h(x)$ . The change in the expected loss is given by

$$\Delta = \mathbb{E}[L_{\hat{s}'}] - \mathbb{E}[L_{\hat{s}}] = \int dx f(x|\theta_0)(h^2(x) + 2h(x)\hat{s}(x)) + f(x|\theta_1)(h^2(x) - 2h(x) + 2h(x)\hat{s}(x)) . \quad (19)$$

using  $\hat{s}(x) = f(x|\theta_1)/(f(x|\theta_0) + f(x|\theta_1))$  we obtain

$$\Delta = \int dx (f(x|\theta_0) + f(x|\theta_1)) h^2(x) > 0 . \quad (20)$$

87 Thus any variation on  $\hat{s}(x) = f(x|\theta_1)/(f(x|\theta_0) + f(x|\theta_1))$  has a larger expected loss.

88 The conclusion is that standard classification with a quadratic loss function and training  
 89 data as described above will approximate a discriminative classifier needed to produce an  
 90 equivalent likelihood ratio test.

91 Once the classifier is trained, we can use the generative model and any univariate density  
 92 estimation technique (e.g. histograms or kernel density estimation) to approximate  $\hat{g}(x|\theta)$ .

93 This treatment shows that in the asymptotic limit of large samples from the generative  
 94 model, that we can approximate arbitrarily well the original likelihood ratio test. With  
 95 finite training data for  $\hat{s}(x)$  and samples to approximate  $\hat{g}(x)$  it will be necessary to be more  
 96 specific about the what loss function we are interested in for approximating the likelihood  
 97 ratio test. This will depend in general on the ultimate goal of the test. We know that in  
 98 the case of composite hypothesis tests that there is in general, not uniformly most powerful  
 99 test, thus it is likely that a decision theoretic approach taking into account some weighting  
 100 or utility over the space  $\Theta$  is necessary. This is left as a subject for future work.

## 5.2 Training a parametrized, discriminative classifier

We are left with the practical question of how to train a family of discriminative classifiers parametrized by  $\theta_0$  and  $\theta_1$ , the parameters associated to the null and alternate hypotheses, respectively. While this could be done independently for all  $\theta_0$  and  $\theta_1$ , it is desirable and convenient to have a smooth evolution of the classification score as a function of the parameters. Thus, we anticipate a single learning stage based on training data with input  $(x, \theta_0, \theta_1)_i$  and target  $y_i$ . Somewhat unusually, the unknown values of the parameters are taken as input to the classifier, as latent variables whose values will be specified via the enveloping (generalized) likelihood ratio test. We denote the learned family of classifiers  $\hat{s}(x; \theta_0, \theta_1)$ , and anticipate the training based roughly on the following algorithmic flow.

---

### Algorithm 1 Training the parametrized classifier

---

```

initialize trainingData = {}
for  $\theta_0$  in  $\Theta$  do
  for  $\theta_1$  in  $\Theta$  do
    generate  $x_i^0 \sim f(x|\theta_0)$ 
    append  $\{(x_i^0, \theta_0, \theta_1, y = 0)\}$  to trainingData
    generate  $x_i^1 \sim f(x|\theta_1)$ 
    append  $\{(x_i^1, \theta_0, \theta_1, y = 1)\}$  to trainingData
  end for
end for
use trainingData to learn  $\hat{s}(x; \theta_0, \theta_1)$ 

```

---

While the function  $f(x|\theta_1)/(f(x|\theta_0) + f(x|\theta_1))$  will minimize the expected squared loss based on training data produced according to the algorithm above, it is not clear how training data from  $\theta'_0 \neq \theta_0$  and  $\theta'_1 \neq \theta_1$  will influence a real world classifier with finite capacity. This is left as an area for future work.

## 5.3 Embedding the classifier into the likelihood

In most settings that make use of likelihood ratio tests, the probability densities directly model the observed data via  $f(x|\theta)$ . In the case of a fixed classifier  $\hat{s}(x)$  it is possible to pre-compute  $\hat{s}_e = \hat{s}(x_e)$ . In the parametrized setting this it is not possible to pre-compute  $\hat{s}(x; \theta_0, \theta_1)$  for all values of  $\theta_0$  and  $\theta_1$ , so we must computationally we must embed the classifier into the likelihood function to carry out the composition  $g \circ s$ . A concrete realization of this has been performed for probability models implemented with the `RooFit` probabilistic programming language and discriminative classifiers implemented with `scikit-learn` and `TMVA`.

In both cases, constructing the density  $\hat{g}(\hat{s}|\theta)$  requires running the generative model at  $\theta$ , and in the context of a likelihood fit this would mean that the optimization algorithm that is trying to maximize  $\theta$  needs access to the generative model  $f(x|\theta)$ . This can be impractical

127 when the generative model is computationally expensive or has high-latency (for instance  
 128 some human intervention is required to reconfigure the generative model). In practice,  
 129 one may want to interpolate the distribution between discrete values of  $\theta$  to produce a  
 130 continuous parametrization for  $\hat{g}(s|\theta)$ . In such cases, the properties of the interpolation  
 131 algorithm should be part of the considerations of the over-arching optimization problem.

## 132 6. Typical usage of machine learning in HEP

In high-energy physics (HEP) we are often searching for the properties of some class of events, generically referred to as *signal*, in the presence of a separate class of *background* events. For each event we measure some quantities  $x$  that have corresponding distributions  $f_b(x|\nu)$  for background and  $f_s(x|\nu)$  for signal, where  $\nu$  are nuisance parameters describing uncertainties in the underlying physics prediction or response of the measurement device. In the simple setup, the total model is a mixture of the signal and background components, and  $\mu$  is the mixture coefficient associate dot the signal component. The generative model in this case is

$$f(D|\mu, \nu) = \prod_{e=1}^n [\mu f_s(x_e|\nu) + (1 - \mu) f_b(x_e|\nu)] , \quad (21)$$

133 New particle searches correspond to the hypothesis test  $\mu = 0$ , and are generally formulated  
 134 with the generalized likelihood ratio profiling over  $\nu$ .

135 Often machine learning classification algorithms are trained on large samples of syn-  
 136 thetic data  $\{x_i, y_i\}$  generated with some nominal values of the parameters  $\nu_0$ , where  $y = 0$   
 137 corresponds to background and  $y = 1$  corresponds to signal. Following the result of Sec-  
 138 tion 5, the resulting classifier approximates the likelihood ratio of the mixture components  
 139  $f_s(x|\nu_0)/(f_s(x|\nu_0) + f_b(x|\nu_0))$ , which is one to one with the likelihood ratio of the null to  
 140 the alternate  $f(x|\mu = 0, \nu_0)/f(x|\mu, \nu_0)$  for all  $\mu$ . The resulting classifier is denoted  $\hat{s}(x)$ .  
 141 Based on this classifier and large samples of synthetic data drawn from  $f_s(x|\nu)$  and  $f_b(x|\nu)$   
 142 we construct the distributions  $g_s(\hat{s}|\nu)$  and  $g_b(\hat{s}|\nu)$ . An example of the distributions of the  
 143 distribution of  $\hat{s}$  for the signal and background events with  $\nu = \nu_0$  is shown in Figure 1.

These steps lead to a subsequent statistical analysis where one observes in data  $D = (x_1, \dots, x_n)$ . For each event, the classifier is evaluated and one performs inference on a parameter  $\mu$  related to the presence of the signal contribution. In particular, one forms the statistical model

$$g(D|\mu, \nu) = \prod_{e=1}^n [\mu g_s(\hat{s}(x_e)|\nu) + (1 - \mu) g_b(\hat{s}(x_e)|\nu)] , \quad (22)$$



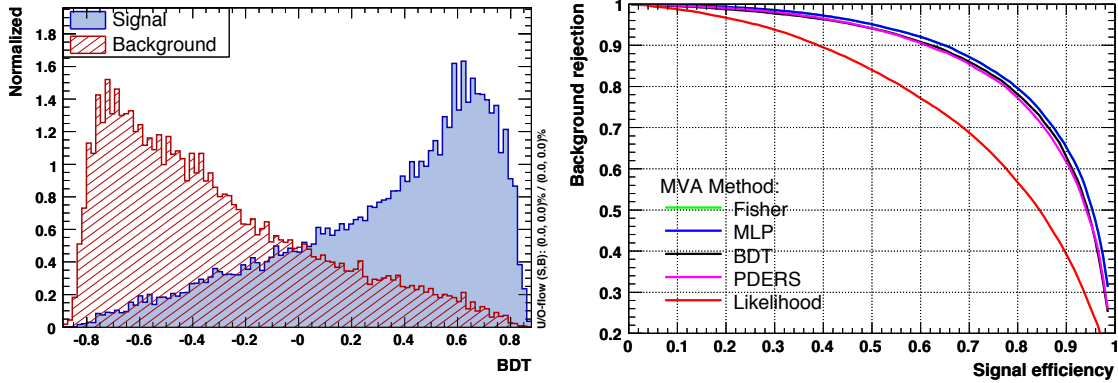


Figure 1: Left: an example of the distributions  $g_b(\hat{s}|\nu)$  and  $g_s(\hat{s}|\nu)$  when the classifier  $s$  is a boosted-decision tree (BDT). Right: the corresponding ROC curve (right) for this and other classifiers. (Figures taken from TMVA manual.)

where  $\mu = 0$  is the null (background-only) hypothesis and  $\mu > 0$  is the alternate (signal-plus-background) hypothesis.<sup>2</sup> Typically, we are interested in inference on  $\mu$  and  $\nu$  are nuisance parameters.

## 6.1 Comments on typical usage of machine learning in HEP

Nuisance parameters are an after thought in the typical usage of machine learning in HEP. In fact, most machine learning discussions would only consider  $f_b(x)$  and  $f_s(x)$ . However, as experimentalists we know that we must account for various forms of systematic uncertainty, parametrized by nuisance parameters  $\nu$ . In practice, we take the classifier as fixed and then propagate uncertainty through the classifier as in Eq. 22. Building the distribution  $g(\hat{s}|\nu)$  for values of  $\nu$  other than the nominal  $\nu_0$  used to train the classifier can be thought of as a calibration necessary for classical statistical inference; however, this classifier is clearly not optimal for  $\nu \neq \nu_0$ .

## 6.2 A more powerful approach

The standard use of machine learning in HEP can be improved by training a parametrized, discriminative classifier corresponding to the generalized likelihood ratio test

$$\lambda(\mu) = \frac{f(D|\mu, \hat{\nu})}{f(D|\hat{\mu}, \hat{\nu})}, \quad (23)$$

2. Sometimes there is an additional Poisson term when expected number of signal and background events is known, which is referred to as an extended likelihood.

157 following the approach outlined in Section 4.

There is an interesting distinction between this approach and the standard use in which the classifier is trained for a fixed  $\nu_0$ . In the standard use one trains a classifier for signal vs. background, which is equivalent (in an ideal setting) to training a classifier for null (background-only) vs. alternate (signal-plus-background) as

$$\frac{f(x|0, \nu_0)}{f(x|\hat{\mu}, \nu_0)} = \frac{f_b(x|\nu_0)}{\mu f_s(x_e|\nu_0) + (1 - \mu) f_b(x_e|\nu_0)} = \left[ c_1 + c_2 \frac{f_s(x|\nu_0)}{f_b(x_e|\nu_0)} \right]^{-1}, \quad (24)$$

and  $c_1$  and  $c_2$  are constants. Specifically, the two approaches likelihood ratios are in one-to-one correspondence, so an ideal algorithm would lead to equivalent tests. In contrast, in the case of the generalized likelihood ratio test

$$\frac{f(x|0, \hat{\nu})}{f(x|\hat{\mu}, \hat{\nu})} = \frac{f_b(x|\hat{\nu})}{\hat{\mu} f_s(x_e|\hat{\nu}) + (1 - \hat{\mu}) f_b(x_e|\hat{\nu})}, \quad (25)$$

158 the background components don't cancel and there is an additional term  $f_b(x|\hat{\nu})/f_b(x|\hat{\nu})$ .  
 159 In practice, with classifiers of finite capacity, there will be some tradeoff between taking  
 160 into account this additional term and the more challenging learning problem when  $\mu$  is  
 161 very small.

### 162 6.3 Related work

163 Refs to include: Clayton Scott; Andrew Y. Ng; Sam T. Roweis; Eric P. Xing; Sujay Sang-  
 164 havi; Bianca Zadrozny; Zadrozny and Elkan (2001); Tommi Jaakkola  
 165 Xin Tong (2013)

## 166 7. Conclusions

167 We have shown that a parametrized family of discriminative classifiers  $s(x; \theta_0, \theta_1)$  trained  
 168 and calibrated with a generative model  $f(x|\theta)$  can be used to approximate statistical in-  
 169 ference likelihoods based on the ratio  $f(x|\theta_0)/f(x|\theta_1)$  when it is not possible to evaluate the  
 170 densities  $f(x|\theta)$  for an arbitrary  $x$ . This approach leverages the power of machine learning  
 171 in a classical statistical setting.

## 172 Acknowledgements

173 KC would like to thank Daniel Whiteson for discussions and encouragement throughout  
 174 the project, Alex Ihler for challenging discussions that led to a reformulation of the initial  
 175 idea, and Shimon Whiteson for patient feedback in that process. KC would also like to  
 176 thank Yann LeCun, Philip Stark, and Pierre Baldi for their feedback on the project early  
 177 in its conception, Balázs Kégl for discussions about the Kaggle challenge and non-standard  
 178 loss functions, and Yuri Shirman for reassuring discussions on elementary vector calculus.

179 KC is supported by the US National Science Foundation grants PHY-0854724 and PHY-  
180 0955626. KC is grateful to UC-Irvine for their hospitality while this research was carried  
181 out and the Moore and Sloan foundations for their generous support of the data science  
182 environment at NYU.

## 183 References

- 184 Michael I. Jordan Andrew Y. Ng. On Discriminative vs. Generative clas-  
185 sifiers: A comparison of logistic regression and naive Bayes. URL  
186 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.9829&rank=1>.
- 187 Charles Elkan Bianca Zadrozny. Obtaining calibrated probability es-  
188 timates from decision trees and naive Bayesian classifiers. URL  
189 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.3039>.
- 190 Robert Nowak Clayton Scott. A Neyman-Pearson approach to statistical learning. URL  
191 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.6850&rank=5>.
- 192 Michael I. Jordan Stuart Russell Eric P. Xing, Andrew Y. Ng. Distance Met-  
193 ric Learning, With Application To Clustering With Side-Information. URL  
194 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.6509&rank=4>.
- 195 Lawrence K. Saul Sam T. Roweis. Nonlinear dimen-  
196 sionality reduction by locally linear embedding. URL  
197 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.3313>.
- 198 Alan Willsky Sujay Sanghavi, Vincent Tan. Learning graph-  
199 ical models for hypothesis testing and classification. URL  
200 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.4311>.
- 201 David Haussler Tommi Jaakkola. Exploiting Generative Models in Discriminative Classi-  
202 fiers. URL <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.7709>.
- 203 Xin Tong. A Plug-in Approach to Neyman-Pearson Classification.  
204 *Journal of Machine Learning Research*, 14:3011–3040, 2013. URL  
205 <http://jmlr.org/papers/v14/tong13a.html>.
- 206 Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from  
207 decision trees and naive Bayesian classifiers. pages 609–616, June 2001. URL  
208 <http://dl.acm.org/citation.cfm?id=645530.655658>.